EXata 5.1
User's Guide

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

Preface

Who Should Read this Guide

The *EXata 5.1 User's Guide* is intended for users familiar with the Windows or Linux operating systems. It assumes that you are familiar with network terminology and concepts.

How this Guide is Organized

This guide contains the following information:

- Chapter 1 introduces EXata and describes its components.
- Chapter 2 describes the EXata Command Line Interface, including the command to run EXata from the command line and the format of the input and output files.
- Chapter 3 describes the Design mode of EXata Architect, including its layout and controls, basic editing, building scenarios, creating and modifying applications, devices, nodes and network components, and hierarchical network design.
- Chapter 4 describes how to model networks in EXata.
- Chapter 5 describes how to use EXata for emulation.
- Chapter 6 describes the Visualize mode of EXata Architect, including an overview of the various options, commands and functions in the Architect Visualize mode, which are used to run and animate experiments.
- Chapter 7 provides an overview of the various options, commands, and functionalities of the Analyzer component of EXata GUI. It also provides examples of analyzing statistics in EXata Analyzer.
- Chapter 8 describes the EXata Packet Tracer, including components, menu and toolbar, tracer table columns, left and bottom panels, and how to use the filter queue editor.
- Chapter 9 describes the EXata File Editor, including component descriptions and editing features.
- Appendix A contains information on license and installed libraries.
- Appendix B describes how to use the online help features of EXata.
- Appendix C has Advanced Run Settings information.
- Appendix D contains information on the EXata Utility Programs.

EXata Document List

The following table shows the EXata Documentation Set and offers a brief description of each document.

Document	Description
EXata Connection Manager User's Guide	This guide provides information on installing and using EXata Connection Manager.
EXata API Reference Guide	This guide is a supplement to *EXata Programmer's Guide* and provides detailed information on the EXata API functions and parameters. This is available in both PDF and HTML formats.
EXata Connection Manager User's Guide	This guide provides information on installing and using EXata Connection Manager.
EXata Distributed Reference Guide	This guide provides instructions for running EXata on a distributed architecture.
EXata Documentation Portfolio	The documentation portfolio combines all EXata documents in a single PDF file.
EXata Installation Guide	This guide provides detailed steps for installing EXata on Windows and Linux platforms.
EXata Model Libraries	This set of documents contains detailed reference information on all EXata models and includes the following protocol libraries. See *EXata Model Library Index* for an alphabetical list of all our models and a reference to which library they can be found in. Advanced Wireless Cellular Cyber Developer Federation Interfaces LTE Multimedia and Enterprise Network Management Sensor Networks UMTS Urban Propagation Wireless
EXata Product Tour	This tour provides an introduction to EXata by means of an example.
EXata Programmer's Guide	This is a guide to the EXata programming interface and functions, allowing users to develop and customize protocol models.
EXata Release Notes	This document lists the changes (added and removed features, bug fixes, etc.) made in the current version of EXata with respect to the previous version.
EXata Statistics Database User's Guide	This is a guide to the statistics database generated by EXata.
EXata User's Guide	This is a detailed guide for using *EXata* and works in combination with the *EXata Model Libraries* set of documents.

Document Conventions

EXata documents use the following conventions:

Convention	Description
Book Title	Title of a document.
`Command Input`	A command name or qualified command phrase, daemon, file, or option name.
`Command Output`	Text displayed by the computer.
Note: or **Notes:**	Information of special interest.
[ ]	In syntax definitions, square brackets indicate items that are optional.
`Code Segment`	Segment of code from EXata source files used for illustration.
`Added Code`	Example of code that the user should add to existing EXata functions and declarations to add a custom model to EXata. A vertical margin in the left column indicates new lines of code that need to be added.
Ellipses (. . .)	Ellipses are used to indicate lines of code from EXata source files that have been omitted from an example for the sake of brevity.

More Information

- For general information about SCALABLE, visit the company website at www.scalable-networks.com.
- For more information on EXata, please contact EXata Sales at info@scalable-networks.com or visit the EXata website at www.exata.com.
- For technical help on EXata or help on EXata documentation, please contact EXata Support at support@scalable-networks.com or visit our Support website at support.scalable-networks.com.

1 Introduction to EXata

EXata is a network emulator that lets you evaluate on-the-move communication networks faster and with more realism than any other emulator. With the optional Cyber Model Library, EXata can be used as a toolkit for research and development, test and evaluation, and training of cyber warfare technologies. It uses a software virtual network (SVN) to digitally represent the entire network, the various protocol layers, antennas, and devices. EXATA_HOME can interoperate, at one or more protocol layers, with real radios and devices to provide hardware-in-the-loop capabilities. EXata can also be connected to systems with real applications, which run on the SVN just as they would run on real networks.

Emulation and Simulation

A network emulator mimics the functions of a real network so that it appears, interacts, and behaves like the real network. The emulator provides an exact, high quality, reproduction of external behavior so that the emulated system is indistinguishable from the real system. An emulator provides a cost-effective method of evaluating new network technologies before actual systems or networks are built.

A network simulator duplicates the behavior of a real network, but cannot interact with real networks. A simulator uses lower quality reproduction or abstraction of the real system and focuses on simply replicating the real network's behavior. A network simulation is a cost-effective method for developing the early stages of network centric systems. Users can evaluate the basic behavior of a network and test combinations of network features that are likely to work.

Network emulation helps in developing a net-centric system by providing an environment in which design decisions can be easily changed and their impact evaluated. Customers of the net-centric system can use the emulated network and see how their applications (such as VoIP, situational awareness, sensor data, and streaming video) will perform when the real system is built. The emulated network can also be integrated with legacy systems to test interoperability and be used to train users on the next generation networks. By evaluating what works best early in the design cycle, the cost of modifying a system can be greatly reduced. This also sets realistic expectations of what the communications network will deliver, i.e., it provides predictability.

The emulation capabilities of EXata in conjunction with the Cyber Model Library can be used to create and modify attacks and counter measures. Attacks can be targeted on specialized networks, such as wireless, wired, mobile ad-hoc (MANET), and tactical networks. Users can then analyze the impact of attacks and counter-measures on the network itself, applications, and end-users.

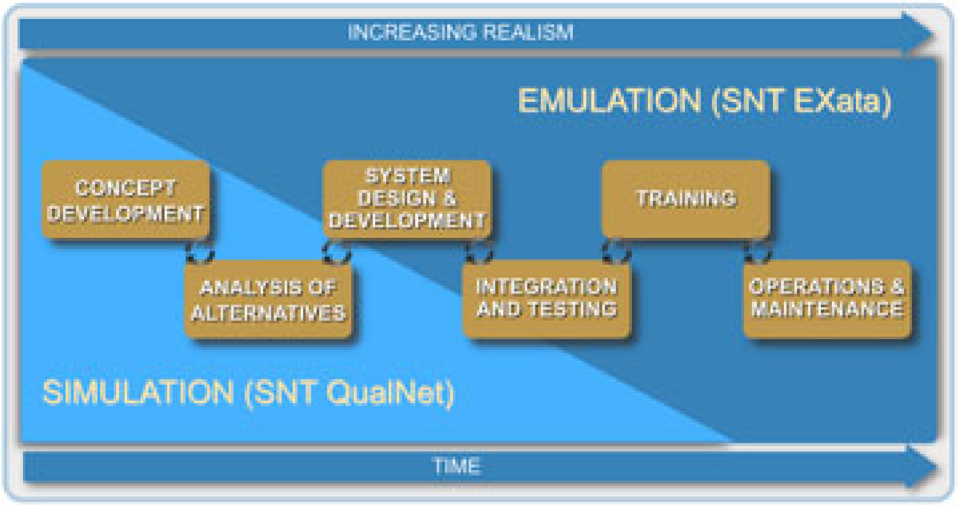


FIGURE 1-1. Life-cycle of Developing a Net-Centric System

EXata is composed of the following tools:

- EXata Architect — A graphical experiment design and visualization tool. Architect has two modes: Design mode, for designing experiments, and Visualize mode, for running and visualizing experiments.
- EXata Analyzer — A graphical statistics analyzing tool.
- EXata Packet Tracer — A graphical tool to display and analyze packet traces.
- EXata File Editor — A text editing tool.
- EXata Command Line Interface — Command line access to the simulator.

Note: Some of the features in this guide are not included with the standard EXata distribution and may be sold separately. To inquire about purchasing additional EXata modules, contact Scalable Network Technologies' Sales Department at sales@scalable-networks.com.

1.1 EXata Features and Benefits

EXata is a comprehensive suite of tools for emulating large wired and wireless networks. It uses simulation and emulation to predict the behavior and performance of networks to improve their design, operation, and management. EXata SVN provides a cost-effective and easy-to-use alternative to physical testbeds that typically have high equipment costs, complex setup requirements and limited scalability.

EXata enables users to:

- Develop emulation or simulation models for new networking technologies.
  - Design new network technologies: Design new communications protocol models using the OSI-style architecture of EXata protocol stack.
  - Design wireless networks of real world size: EXata can be run on multi-core processor computers to evaluate large wireless networks of 100s and 1000s of devices.

  - Perform what-if analyses: Analyze the performance of networks and perform “what-if” analyses to optimize them. You can design a network, and then run batch experiments to test network performance when parameters, such as routing protocols, timers, and transmission power, are varied.
- Connect real networks, applications, and devices with EXata emulated network.
  - See real applications run on emulated networks: EXata can run real applications, such as VoIP, Internet browsers, and streaming video, on emulated networks as if they are running on real networks.
  - Train with the network before it is ready to be deployed: EXata makes it possible to conduct training and operations with next-generation tactical networks and devices that are still being designed.
- Analyze and manage EXata virtual networks with popular, industry-standard, tools.
  - Snoop on packets: EXata has a packet sniffer interface that enables third-party tools like Wireshark and Microsoft Network Monitor to snoop/capture traffic from any device in EXata and analyze it. This lets you debug and troubleshoot network problems.
  - Manage an emulated network: EXata comes with a SNMP Agent, which enables you to use standard SNMP managers to view, monitor and control emulated networks in EXata just like managing real networks.
- Develop, test and evaluate, and train users on cyber warfare and network security technologies.
  - Research and develop new methods of attacks, counter measures, and intrusion detection.
  - Improve capabilities of existing network intrusion detection tools, such as Snort, against new types of attacks.
  - Study the effectiveness of counter-measures against new or existing cyber threats.
  - Train users on network security tools and procedures on isolated and quarantined environment modeled as software virtual networks.
  - Incorporate cyber warfare exercises into conventional wargaming exercises with other SAF/CGF simulations.

The key features of EXata that enable creating a virtual network environment are:

- **Speed**

  EXata can support real-time speed to enable software-in-the-loop, network emulation, and hardware-in-the-loop modeling. Faster speed enables model developers and network designers to run multiple “what-if” analyses by varying model, network, and traffic parameters in a short time.
- **Scalability**

  EXata can model thousands of nodes by taking advantage of the latest hardware and parallel computing techniques. EXata can run on cluster, multi-core, and multi-processor systems to model large networks with high fidelity.
- **Model Fidelity**

  EXata uses highly detailed standards-based implementation of protocol models. It also includes advanced models for the wireless environment to enable more accurate modeling of real-world networks.
- **Portability**

  EXata and its library of models run on a vast array of platforms, including Windows and Linux operating systems, distributed and cluster parallel architectures, and both 32- and 64-bit computing platforms. Users can now develop a protocol model or design a network in EXata on their desktop or laptop Windows computer and then transfer it to a powerful multi-processor Linux server to run capacity, performance, and scalability analyses.

- **Extensibility**
  EXata can connect to other hardware and software applications, such as OTB, real networks, and third party visualization software, to greatly enhancing the value of the network model.

1.2 EXata Architecture

Figure 1-2 illustrates the EXata architecture. A high-level description of the various components is provided below.

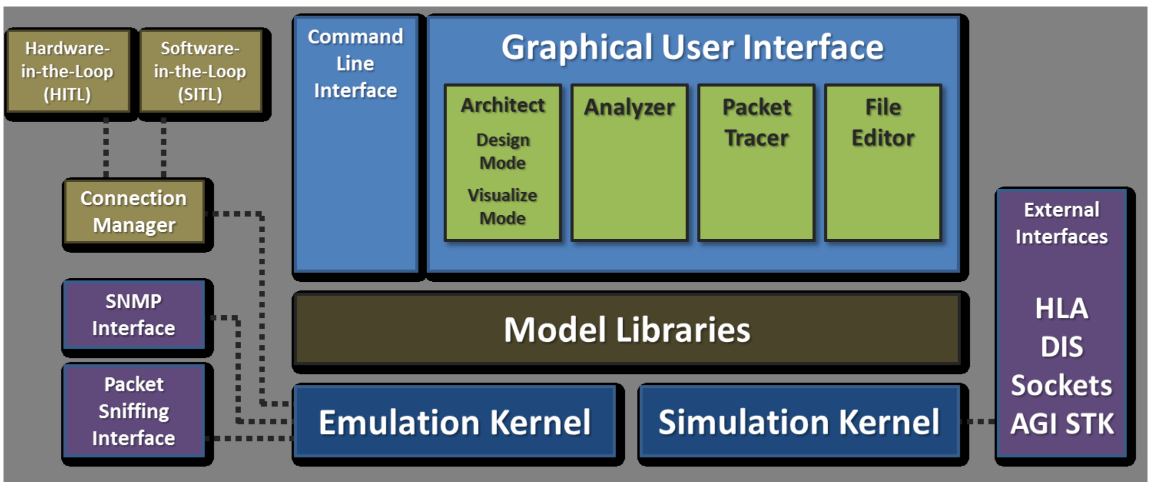


FIGURE 1-2. EXata Architecture

EXata Simulation Kernel

The simulation kernel of EXata is a parallel discrete-event scheduler. It provides the scalability and portability to run hundreds and thousands of nodes with high-fidelity models on a variety of platforms, from laptops and desktops to high performance computing systems. Users do not directly interact with the kernel, but use the EXata API to develop their protocol models.

EXata Emulation Kernel

The emulation kernel of EXata is a high-fidelity, real-time interface to connect the modeled network with external real applications and hardware. The core of the emulation kernel is a real-time event scheduler that processes the internal EXata events, as well as the events from external sources, with real time deadlines. It also provides a transparent interface to the real-world applications and hardware such that the latter can access any emulated network resource as if it exists physically. Users do not directly interact with the kernel, but use the Connection Manager or interfaces such as SNMP, Packet Sniffing interfaces to connect the real applications and hardware with the emulated network.

The EXata emulation kernel is enabled by default and it can be disabled to run EXata in simulation mode. In simulation mode, EXata runs as fast as possible (non-real time mode). Simulation mode is useful for creating and validating scenarios: both for network performance analyses and for potential subsequent use in emulation. In simulation mode, all emulation features (Connection Manager, SNMP and Packet Sniffing Interfaces, Hardware/Software in-the-loop) are disabled.

EXata Model Libraries

EXata supports a number of model libraries that enable you to design networks using various protocol models. EXata includes the Developer, Multimedia and Enterprise, and Wireless Model Libraries. Additional libraries to model network security and cyber attack and defense strategies, cellular networks, satellite networks, UMTS, WiMAX, sensor networks, military radio networks, and advanced propagation model libraries are also available. Refer to the *EXata Model Libraries* data sheet for more information or check the products page on our website.

EXata Graphical User Interface (GUI)

EXata GUI consists of Architect, Analyzer, Packet Tracer, and File Editor.

- **Architect** is a network design and visualization tool. It has two modes: Design mode and Visualize mode.

  In Design mode, you can set up terrain, network connections, subnets, mobility patterns of wireless users, and other functional parameters of network nodes. You can create network models by using intuitive, click and drag operations. You can also customize the protocol stack of any of the nodes. You can also specify the application layer traffic and services that run on the network. Design mode of Architect is described in Chapter 3.

  In Visualize mode, you can perform in-depth visualization and analysis of a network scenario designed in Design mode. As simulations are running, users can watch packets at various layers flow through the network and view dynamic graphs of critical performance metrics. Real-time statistics are also an option, where you can view dynamic graphs while a network scenario simulation is running. Visualize mode of Architect is described in Chapter 6.

  You can also assign jobs to run in batch mode on a faster server and view the animated data later. You can perform “what-if” analysis by setting a range of values for a particular protocol parameter and comparing the network performance results for each of them.
- **Analyzer** is a statistical graphing tool that displays hundreds of metrics collected during simulation of a network scenario. You can choose to see pre-designed reports or customize graphs with their own statistics. Multi-experiment reports are also available. All statistics are exportable to spreadsheets in CSV format. Analyzer is described in Chapter 7.
- **Packet Tracer** provides a visual representation of packet trace files generated during the simulation of a network scenario. Trace files are text files in XML format that contain information about packets as they move up and down the protocol stack. Packet Tracer is described in Chapter 8.
- **File Editor** is a text editing tool that displays the contents of the selected file in text format and allows the user to edit files. File Editor is described in Chapter 9.

EXata Command Line Interface

The EXata command line interface enables a user to run EXata from a DOS prompt (in Windows) or from a command window (in Linux). When EXata is run from the command line, input to EXata is in the form of text files which can be created and modified using any text editor. Building and running scenarios with the command line interface takes less memory and scenarios typically run faster than with the GUI. With the command line interface the users have the flexibility to interface with visualization and analysis tools of their choice. The command line interface is described in Chapter 2.

EXata Connection Manager

EXata Connection Manager is the companion module of the main EXata emulation engine. The EXata emulation engine creates a digital replica of the target network, and EXata Connection Manager is used to run applications on the emulated network. The Connection Manager makes EXata advanced emulation technology easy and simple to use. Applications need no modification or customization to use the realistic emulated network in EXata.

Connection Manager supports a large variety of applications such as:

- Internet browsers
- Tactical communications
- Situational awareness information
- Sensor data
- Instant messaging
- VoIP
- Streaming video
- Multiplayer games

EXata SNMP and Packet Sniffing Interfaces

EXata supports a packet sniffer interface to enable capture and analysis of network traffic using standard packet sniffer/analyzer tools like Wireshark or Microsoft Network Monitor. Additionally, EXata can be managed using standard SNMP network managers like HP OpenView, IBM Tivoli, or SolarWinds Orion.

EXata HLA, DIS, Socket and STK Interfaces

EXata can also interact with a number of external tools in real-time.

- The EXata HLA and DIS modules, which are part of the Federation Interfaces Library, allows EXata to interact with other simulators and computer-generated force (CGF) tools, such as OTB, using High Level Architecture (HLA) or Distributed Interaction Simulation (DIS).
- The Socket Interface, which is a part of the Federation Interfaces Library, provides inter-process communication between EXata and an external program over a TCP socket, with EXata acting as the server and the external program as the client.
- The EXata Satellite Toolkit (STK) Interface, which is a part of the Developer Model Library, provides a way to interface EXata with the Satellite Toolkit developed by Analytical Graphics, Inc. (AGI) and function in a client-server environment.

1.3 Scenario-based Network Emulation

In EXata, a specific network topology is referred to as a scenario. A scenario allows the user to specify all the network components and conditions under which the network will operate. This includes: terrain details, channel propagation effects including path loss, fading, and shadowing, wired and wireless subnets, network devices such as switches, hubs and routers, the entire protocol stack of a variety of standard or user-configured network components, and applications running on the network. Most of these are optional; you can start with a basic network scenario and specify as much detail as necessary to improve the accuracy of your network model.

1.3.1 General Approach

In general, an emulation study comprises the following phases:

- The first phase is to create and prepare the simulation scenario based on the system description and metrics of interest. Section 1.3.2 introduces the steps for creating a scenario which are described in detail in Chapter 4.
- The second phase is to configure the emulation test-bed. Chapter 5 provides a description of the steps involved in creating the emulation test-bed.
- The next step is to execute, visualize, and analyze the created scenario and collect simulation results. Simulation results can include scenario animations, runtime statistics, final statistics, and output traces. Chapter 6 describes how to visualize scenarios at run time.
- Concurrently, external applications and hardware can interact with a scenario at run time. Chapter 5 describes the EXata interface for interacting with a network scenario.
- The last phase is to analyze the simulation results. Typically, users may need to adjust the scenarios based on the collected simulation results. Chapter 7 describes how to analyze simulation results with EXata Analyzer.

This general procedure is illustrated in Figure 1-3.

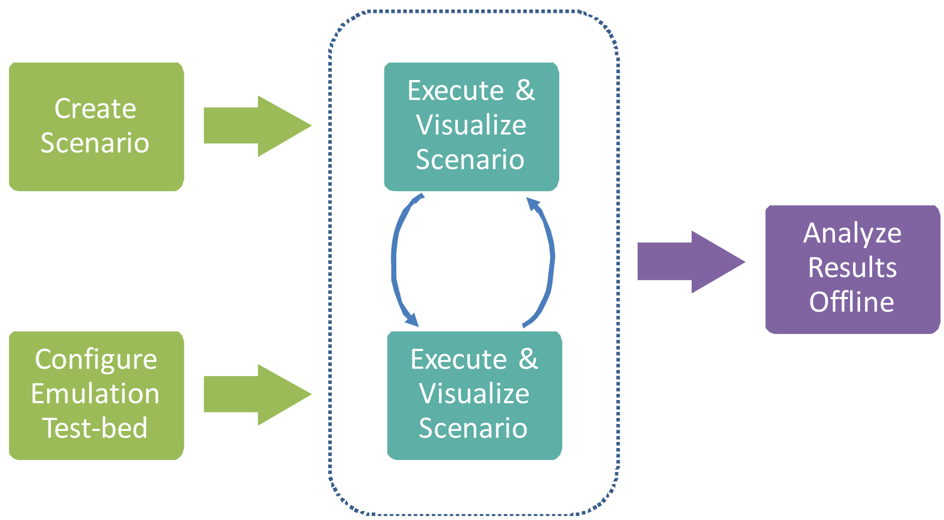


FIGURE 1-3. Emulation Using EXata

1.3.2 Creating Scenarios

Creating a scenario can be divided into several steps focusing on different aspects. The key steps in creating a simulation scenario for EXata are illustrated in Figure 1-4. The general approach is to first configure the general properties which are applicable to the whole scenario. Next, specify the network topology by creating subnets, placing nodes, and defining node mobility. Then one needs to configure the protocol stack for individual nodes or groups of nodes as necessary. The last step is to configure parameters for collecting simulation results and controlling runtime performance.

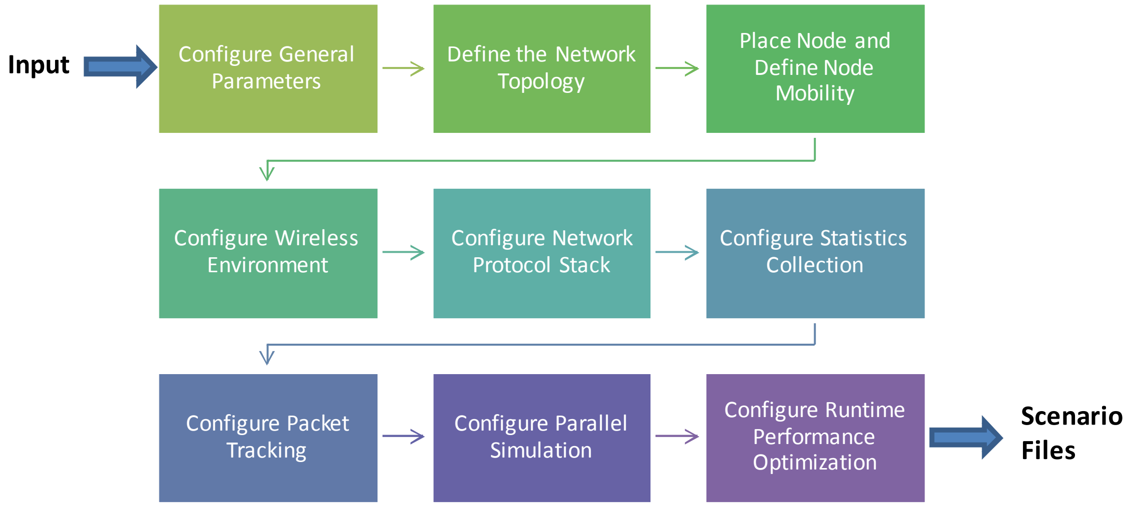


FIGURE 1-4. Creating a Scenario

Refer to Chapter 4 for details on how to create scenarios in both the command line and the GUI interfaces.

1.3.3 Files Associated with a Scenario

Input to the EXata simulator consists of several files. For the command line interface, the input files are text files. The main input files for command line are:

- Scenario configuration file: This is the primary input file for EXata and specifies the network scenario and parameters for the simulation. This file usually has the extension ".config".
- Node placement file: This is referenced by the scenario configuration file and specifies the initial position of nodes in the scenario. (The node placement file may also contain the future positions of nodes.) This file usually has the extension ".nodes".
- Application configuration file: This is referenced by the scenario configuration file and specifies the applications running on the nodes in the scenario. This file usually has the extension ".app".

In addition to the above three files, EXata may use other input files. These additional files depend upon the models specified in the configuration file and are referenced by the configuration file. The input files are described in detail in Chapter 2. These input files are text files which can be created using any text editor. When using the command line interface, the user has to create these files manually.

When the user creates a scenario in Architect, the major input files representing the scenario (scenario configuration, node placement, and application configuration files) are automatically created by Architect.

The primary output file generated by an EXata simulation run is a statistics file, which has the extension ".stat". This file contains the statistics collected during the simulation run. Other output files that may be generated by EXata include the trace file (which has the extension ".trace") which records packet traces, and the animation file (which has the extension ".anim") which records the animation trace of a scenario when the scenario is run in Architect.

Both the statistics and trace files are text files which can be viewed using any text editor. In addition, Analyzer can be used to view the contents of the statistics file in a graphical, easy to analyze manner.

The output files are described in detail in Chapter 2.

1.4 Using EXata

1.4.1 System Requirements and Installation

Refer to *EXata Installation Guide* for the system requirements and instructions to install EXata on Windows and Linux systems. Refer to *EXata Distributed Reference Guide* for additional requirements and instructions for running EXata on distributed platforms.

1.4.2 License

A valid license file is required to run EXata. Your license file will activate the base version of EXata and all additional libraries purchased by you. The base version of EXata includes the following model libraries:

- Developer Model Library
- Multimedia and Enterprise Model Library
- Network Management Model Library
- Wireless Model Library

See Appendix A for more information on license files.

1.4.3 Executable Files

The EXata distribution includes one or more executable files that have been compiled with the following model libraries that are part of the base EXata distribution:

- Developer Model Library
- Multimedia and Enterprise Model Library
- Network Management Model Library
- Wireless Model Library

and the following addon libraries:

- Advanced Wireless Model Library
- Cellular Model Library
- Cyber Model Library
- Federation Interfaces Library
- LTE Model Library
- Sensor Networks Model Library
- UMTS Model Library
- Urban Propagation Model Library

EXata does not need to be recompiled in order to use the models in these libraries. However, EXata will need to be recompiled if the source code is modified or any other libraries or addons are installed. Refer to *EXata Programmer's Guide* for instructions for compiling EXata.

The EXata executable files are located in the bin folder of the EXata installation directory.

Windows Executable Files

For Windows platforms, the EXata distribution includes the following executable files:

- exata-precompiled-32bit.exe: This is a 32-bit executable that can run on both 32-bit and 64-bit platforms.
- exata.exe: This is a copy of exata-precompiled-32bit.exe.

  Note that exata.exe is overwritten every time you recompile EXata. If you recompile EXata but want to use the pre-built executable, then copy the file exata-precompiled-32bit.exe to exata.exe.

Note: If you copy exata-precompiled-32bit to exata.exe, you must also copy libexpat.dll and pthreadVC2.dll from EXATA_HOME/lib/windows to EXATA_HOME/bin. (EXATA_HOME refers to the directory where EXata is installed.)

Linux Executable Files

For Linux platforms, the EXata distribution includes the following executable files:

- exata-precompiled-32bit (included only for 32-bit platforms): This is a 32-bit executable that can run on 32-bit platforms.
- exata-precompiled-64bit (included only for 64-bit platforms): This is a 64-bit executable that can run on 64-bit platforms.
- exata: This is a copy of exata-precompiled-32bit for 32-bit platforms and a copy of exata-precompiled-64bit for 64-bit platforms.

  Note that the file exata is overwritten every time you recompile EXata. If you recompile EXata but want to use the pre-built executable, then copy the file exata-precompiled-32bit or exata-precompiled-64bit to exata.

Note: The executable files will run only on the machine on which EXata is installed. To use EXata on a different machine, it must be installed on that machine.

1.4.4 Using EXata Command Line Interface

See Chapter 2 and Chapter 4 for details of creating network models and running simulations from EXata's command line interface.

1.4.5 Using EXata GUI

The EXata GUI can be used to create network models and run simulations. In addition, simulation statistics and packet traces can be analyzed graphically using the GUI.

Starting EXata GUI on Windows

To start the EXata GUI on a Windows system, do one of the following:

- Double-click on the following icon on the desktop (this option is available only if you chose to install desktop shortcuts during installation):

- Select **Start > All Programs > Scalable > EXata-5.1 > EXata 5.1 GUI** (this option is available only if you chose to create Start menu options during installation).

- Open a command window and type the following commands:

```
cd %EXATA_HOME%\bin
EXataGUI.exe
```

- Navigate to bin folder of the EXata installation directory in an explorer window and double-click on the file EXataGUI.exe.

Notes: **1.** Some firewall programs may prevent EXata GUI from running. To use EXata GUI, you may need to add it to the exception list of your firewall program. Check the documentation of your firewall program for details on adding a program to the exception list or contact your system administrator. If you are using Microsoft Windows firewall, visit the Microsoft website for details of adding a program to the exception list.

2. To run a scenario in emulation mode in EXata GUI, the GUI must be launched from an administrator account.

Starting EXata GUI on Linux

To start the EXata GUI on a Linux system, do one of the following:

- Double-click on the following icon on the desktop (this option is available only if you installed EXata using the installer's GUI and chose to install desktop shortcuts):

- Open a command window and type the following commands:

```
cd $EXATA_HOME/bin
./EXataGUI
```

Note: To run a scenario in emulation mode in EXata GUI, the GUI must be launched from a root user account.

The following screen is displayed when the EXata GUI starts (EXata GUI opens in the Design mode of Architect):

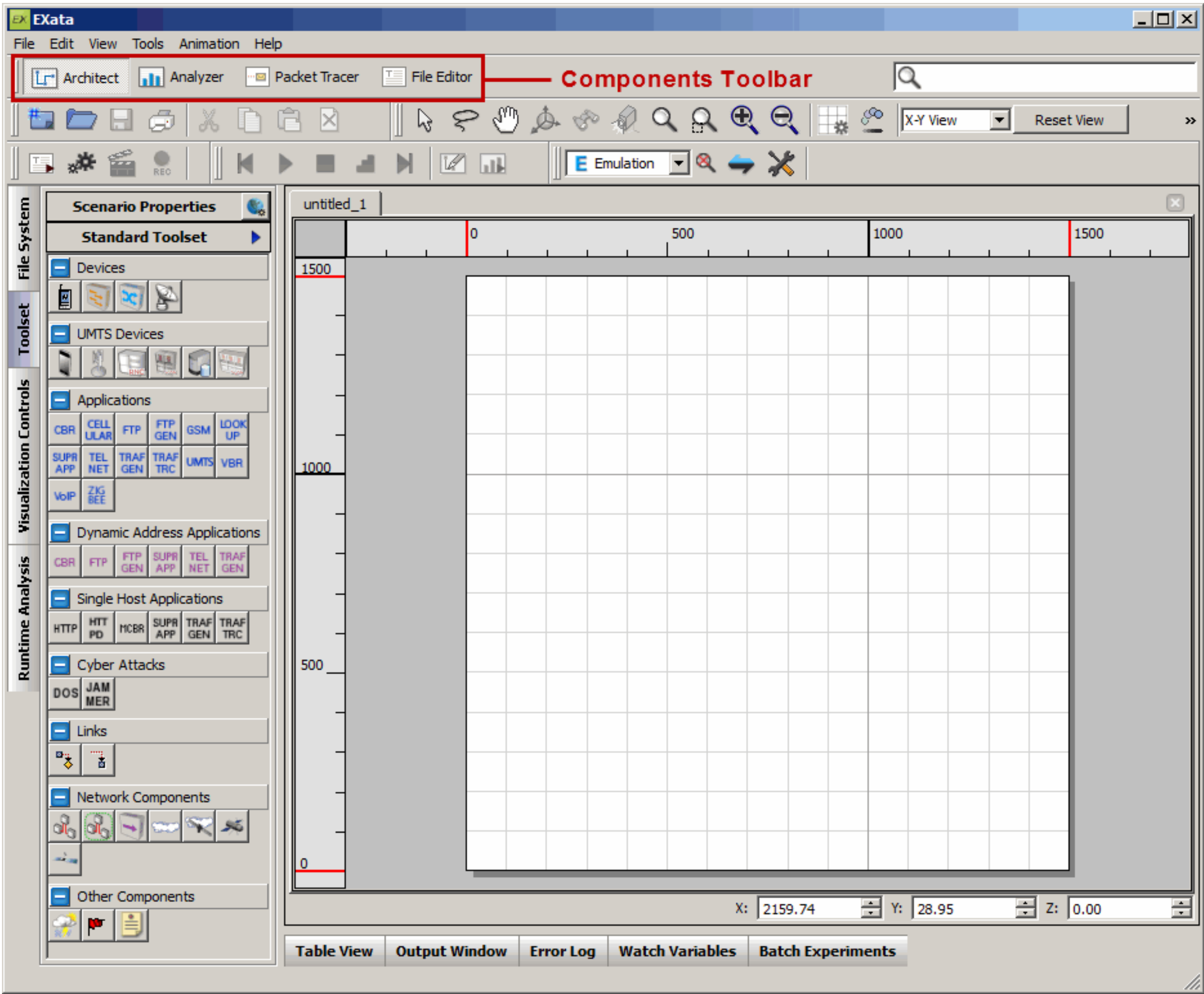


FIGURE 1-5. EXata Startup Screen

Components Toolbar

You can switch between the GUI components (Architect, Analyzer, Packet Tracer, and File Editor) by using the Components Toolbar (see Figure 1-5), which is present in all components.

See Chapter 3 and Chapter 4 for details of creating network models using the Design mode of Architect.

See Chapter 6 for details of running simulations in Visualize mode of Architect.

See Chapter 7 for details of analyzing simulation results in Analyzer.

See Chapter 8 for details of analyzing trace files using Packet Tracer.

See Chapter 9 for details of viewing and editing text files associated with a scenario in File Editor.

1.4.6 Online Help

See Appendix B for a description of the online help features that are available in Architect.

1.4.7 Advanced Run Settings

See Appendix C for details of running EXata on a remote machine or on a multi-processor (shared-memory or distributed architecture) system.

1.4.8 Utility Programs

See Appendix D for details of utility programs that are included in the EXata distribution.

2 Command Line Interface

The command line interface enables a user to run EXata from a DOS prompt (in Windows) or from a command window (in UNIX). When EXata is run from the command line, input to EXata is in the form of text files which can be created and modified using any text editor.

The command-line interface enables a user to use batch files on Windows or shell scripts on UNIX systems. The user can create copies of an EXata scenario file, change parameters of interest, for example, the routing protocol or MAC protocol, and then run the scenarios automatically using a batch file or shell scripts. This allows the user to compare the performance of different protocol models for a given network scenario. This is similar to the batch execution capability of the EXata GUI.

While EXata GUI provides a useful graphical interface to visualize simulation scenarios, it may be more convenient to build scenarios using the command line interface. This is especially true for large network scenarios as building large scenarios using the GUI can be tedious and time consuming. Since the input files used by the command line interface are text files, users can employ various tools to edit and manipulate the input files. Building and running scenarios with the command line interface takes less memory and scenarios typically run faster than with the GUI. Lastly, with the command line interface the users have the flexibility to interface with visualization and analysis tools of their choice.

This chapter describes how to use the EXata command line interface. Section 2.1 describes how to run EXata in the command line interface, the input and output files, and the command line parameters.

Section 2.2 describes the syntax of input files used for specifying scenarios and Section 2.3 describes the syntax of the primary output file produced by EXata.

See Section 1.3 for an overview of scenario-based network simulation. Chapter 4 describes in detail how to configure the different components of a scenario.

2.1 Running EXata from the Command Line Interface

This section describes how to run EXata using the command line interface and the input and output files associated with a scenario. (Refer to *EXata Programmer's Guide* for instructions for debugging EXata.)

To run EXata, perform the following steps:

1. Open a command line, and go to the directory where your scenario is located.

2. Run EXata.

For Windows, type:

```
%EXATA_HOME%\bin\exata myconfig.config
```

For Linux, type:

```
$EXATA_HOME/bin/exata myconfig.config
```

In the example above, myconfig.config is a configuration file that describes a scenario to be simulated. Any file with the extension .config can be specified as the configuration file. For information on additional parameters that can be specified in the command line, including the number of processors, see Section 2.1.2.

Notes: **1.** If a message is displayed stating that the EXATA_HOME variable is not set properly, then set the environment variables, as described in Section 2.1.3.

2. To run EXata in emulation mode (i.e., without using the **`-simulation`** option described in Section 2.1.2), this command must be executed from an administrator or root user account.

3. Wait for the simulation to complete. A file with the extension ".stat" is generated. (The exact name of the file that is generated is discussed in Section 2.1.1.2). This file contains the statistics collected during the simulation. The statistics file is a plain text file that can be opened by using any text editor. It can also be viewed graphically using EXata Analyzer (see Chapter 7).

2.1.1 Input and Output Files

EXata uses several input files and generates one or more output files. The file names are of the form:

`<filename>.<extension>`

where

`<filename>`	Any string
`<extension>`	Indication of type of file.
	Examples: config (for configuration file), app (for applications file), stat (for statistics file), etc.

2.1.1.1 Input Files

EXata uses the following three main input files (Chapter 4 describes the input files in detail):

- `<filename>.config`: This is the primary input file for EXata and specifies the network scenario and parameters for the simulation. A sample scenario configuration file, called default.config, is included with the EXata distribution in the directory EXATA_HOME/scenarios/default.
- `<filename>.nodes`: This file is referenced by the configuration file (as the parameter `NODE-POSITION-FILE`) and specifies the initial position of nodes in the scenario. A sample nodes file, called default.nodes, is included with the EXata distribution in the directory EXATA_HOME/scenarios/default. The file default.nodes is referenced by the scenario configuration file default.config.
- `<filename>.app`: This file is referenced by the configuration file (as the parameter `APP-CONFIG-FILE` and specifies the applications running on the nodes in the scenario. A sample applications file, called default.app, is included with the EXata distribution in the directory EXATA_HOME/scenarios/default. The file default.app is referenced by the scenario configuration file default.config.

In addition to the above three files, EXata may use other input files. These additional files depend upon the models specified in the configuration file and are referenced by the configuration file. For example, if the configuration file specifies the routing protocol to be OSPF, then an input file (with the extension .ospf) is also used by EXata. The .ospf file specifies parameters specific to the OSPF protocol. Description of models in the model libraries list any additional input files needed by specific models.

Chapter 4 describes how to create input files for a scenario.

2.1.1.2 Output Files

The primary output file generated by an EXata simulation run is a statistics file which contains the statistics collected during the simulation run. The statistics file has the extension ".stat". The name of the statistics file depends on the `EXPERIMENT-NAME` parameter in the configuration file (see Section 4.2.1) and the `<experiment-name>` command line parameter (see Section 2.1.2), as described below:

- If the `<experiment-name>` parameter is not specified on the command line, and the configuration file does not include the `EXPERIMENT-NAME` parameter, then the statistics file that is generated is named exata.stat.
- If the `<experiment-name>` parameter is not specified on the command line, and the configuration file specifies a value for the `EXPERIMENT-NAME` parameter, then the name of statistics file that is generated is based on the `EXPERIMENT-NAME` parameter.

  Example:

  If the configuration file contains the following line:

  ```
  EXPERIMENT-NAME wireless-scenario
  ```

  then the statistics file that is generated is named wireless-scenario.stat.
- If the `<experiment-name>` parameter is specified on the command line, then the name of statistics file that is generated is based on the `<experiment-name>` command line parameter, irrespective of the value of the `EXPERIMENT-NAME` parameter in the configuration file.

  Example:

  If EXata is run using the following command:

  ```
  exata myconfig.config myscenario
  ```

  then the statistics file that is generated is named myscenario.stat.

Section 2.3 describes the format of the statistics (.stat) file.

Other output files may also be generated by running EXata, depending on the options specified in the configuration file. For example, if packet tracing is enabled by setting parameters in the configuration file (see Section 4.2.10), then a trace file is generated. The trace file has the extension ".trace" and is named following the same rules as the statistics file. Refer to *EXata Programmer's Guide* for a description of the format of the trace (.trace) file.

2.1.2 Additional Command Line Parameters

In addition to the configuration file name, several optional parameters can be specified when EXata is run from the command line. The general format for running EXata from a command line is:

```
exata <input-filename> [simulation] [-animate] [-seed <N>]
      [-interactive <host> <port>]
      [-np <num-processors>][-lookahead <time>] [-loose <time>]
      [-forceeot] [-friendly] [-greedy]
      [<experiment-name>]
```

Notes: **1.** All parameters must be entered on the same line.

2. To run EXata in emulation mode (i.e., without using the `-simulation` option), this command must be executed from an administrator or root user account.

These parameters are explained in Table 2-1.

To list all options, type:

```
exata -h
```

To obtain information about the libraries installed on your system, type:

```
exata -print_libraries
```

To obtain information on the version of EXata installed on your system, type:

```
exata -version
```

To obtain only the version number of EXata installed on your system, type:

```
exata -version_number_only
```

TABLE 2-1. EXata Command Line Parameters

Parameter	Description
`<input-filename>` *Optional parameter*	Name of the configuration file, e.g., myconfig.config. This parameter is required unless the `-h`, `-print_libraries`, `-version`, or `-version_number_only` option is specified.
`-simulation` *Optional parameter*	Option to run EXata in simulation mode. In simulation mode, EXata runs as fast as possible (non-real time mode). All emulation-related features, such as Connection Manager and SNMP, are disabled in simulation mode. If this option is not specified, EXata runs in emulation mode.
`-animate` *Optional parameter*	Option to print animation commands to standard output.
`-seed <N>` *Optional parameter*	Option to run EXata with the specified seed instead of the seed specified in the scenario configuration (.config) file (see Section 4.2.1). `<N>` is the seed to use.
`-interactive <host> <port>` *Optional parameter*	Option to run EXata on in interactive mode with an external GUI communication over TCP sockets. `<host>` is the host running the GUI. `<port>` is the port number for communication. See Section 4.4 for details of running EXata on multiple processors.
`-np <num-processors>` *Optional parameter*	Option to run EXata on multiple processors. `<num-processors>` is the number of processors to use. See Section 4.4 for details of running EXata on multiple processors.
`-lookahead <time>` *Optional parameter*	Option to adjust the parallel lookahead time in order to trade timing precision for greater program performance. `<time>` is the lookahead time specified as an integer with one of the following time unit indicators: `NS` : Nanoseconds `US` : Microseconds `MS` : Milliseconds The time unit indicator is optional and by default it is `NS`.
`-loose <time>` *Optional parameter*	Same as the `-lookahead <time>` (provided for compatibility with the previous versions of EXata).
`-friendly` *Optional parameter*	Option to run EXata such that when EXata is run on a shared memory system, each thread will not try to use as much CPU as possible. This makes it possible to run with more threads than the number of cores.

TABLE 2-1. EXata Command Line Parameters (Continued)

Parameter	Description
-greedy *Optional parameter*	This option is the opposite of the **-friendly** option. If this option is used and EXata is run on a shared memory system, each thread will try to use as much CPU as possible and run as possible. This is the default synchronization mode.
-forceeot *Optional parameter*	Option to force the simulator to run in Earliest Output Time (EOT) mode even though there are Earliest Common Output Time (ECOT) models in a scenario. This is most useful when a scenario contains a mix of TDMA and CSMA wireless or wired models. **Note:** This option should be used with caution as it will not have repeatable results.
<experiment-name> *Optional parameter*	Experiment name. If this parameter is specified, output files are created with this name. See Section 2.1.1.2 for details.
-print_libraries *Optional parameter*	Option to print information about the libraries installed on your system. See Appendix A for details.
-version *Optional parameter*	Option to print information on the version of EXata installed on your system. This command will print out the version number, build number, build date, and the current value of the environment variable EXATA_HOME.
-version_number_only *Optional parameter*	Option to print only the version number of EXata installed on your system.

2.1.3 Environment Variables

The EXata installer for Windows and Linux systems sets the environment variables required for running EXata.

However, if the environment variables are not properly set (for example, when you copy EXata from another machine), a message will be displayed when you run EXata. In this case, you will need to set the environment variables manually, as described in Section 2.1.3.1 (for Windows) and Section 2.1.3.2 (for Linux).

2.1.3.1 Environment Variables for Windows

To set the environment variables on Windows systems, do the following:

1. Right click on **My Computer** and select **Properties** from the menu. Select the **Advanced** tab and click on **Environment Variables**.
2. Add or update the environment variable `EXATA_HOME`. This variable should be set to the root directory of the EXata installation.
3. Add `EXATA_HOME\bin` and `EXATA_HOME\lib` to the `PATH` variable.

2.1.3.2 Environment Variables for Linux

To set the EXata environment variables on a Linux system, perform the following steps (assuming that EXata is installed at ~/scalable/exata/5.1):

1. Open a command window.
2. Edit the shell startup script. To check which shell you are using, type the following command in a command window: **`echo $SHELL`**.

   Edit the shell startup script as follows.

   - For csh and tcsh

     Open ~/.cshrc and add the following lines:

     ```
     setenv EXATA_HOME ~/scalable/exata/5.1
     set path = ( $path ~/scalable/exata/5.1/bin )
     ```

   - For bash

     Open ~/.bashrc and add the following lines:

     ```
     export EXATA_HOME=~/scalable/exata/5.1
     PATH=$PATH:~/scalable/exata/5.1/bin
     ```

   - For sh

     Open ~/.profile and add the following lines:

     ```
     EXATA_HOME=/home/username/scalable/exata/5.1; export EXATA_HOME
     PATH=$PATH:/home/username/scalable/exata/5.1/bin
     ```

Note: Replace `/home/username` with the absolute path to the home directory if necessary.

2.2 Elements of Input Files

This section describes the syntax used for input files that constitute an EXata scenario, such as the configuration (.config) file. For examples of input files, see the files used by the sample scenario in EXATA_HOME/scenarios/default.

2.2.1 Comments

In an input file anything following a "`#`" character is treated as a comment. The sample configuration file, default.config, contains descriptive information about the parameters that can be configured, using this style of commenting.

2.2.2 EXata Time Format

A scenario typically requires several time values to be specified. EXata uses the following format for specifying time values:

```
<Numeric-value>[<Time-unit>]
```

where

`<Numeric-value>`	A non-negative real or integer value indicating the time value
`<Time-unit>`	Abbreviation for the time unit. The different time units are listed in Table 2-2. The time unit specification is optional. If the time unit is not specified, then it is assumed to be seconds.

Note: There should not be any spaces between the numeric value and the time unit.

TABLE 2-2. EXata Time Units

Time Unit Abbreviation	Description
NS	Nanoseconds
US	Microseconds
MS	Milliseconds
S	Seconds
M	Minutes
H	Hours
D	Days

Examples of specifications of time values in EXata are: `20MS`, `2.5S`, `100`, and `5M`.

2.2.3 Coordinate and Orientation Formats

In an EXata scenario, a node's position is indicated by specifying its coordinates and orientation. Coordinates can be specified in the Cartesian system, the Latitude-Longitude-Altitude (Lat-Lon-Alt) system, or the Military Grid Reference System (MGRS). The coordinate system used in a scenario is selected by setting the `COORDINATE-SYSTEM` parameter in the configuration file (see Section 4.2.2). All coordinate specifications in a scenario should use the same system.

In the Cartesian system, coordinates are specified using the following format:

```
(<x>, <y>[, <z>])
```

where

`<x>`	x-coordinate, in meters. This is specified as a real number.
`<y>`	y-coordinate, in meters. This is specified as a real number.
`<z>`	z-coordinate, in meters. This is specified as a real number. Specifying the z-coordinate is optional, and it is assumed to be zero when not specified explicitly.

In the Lat-Lon-Alt system, coordinates are specified using the following format:

```
(<lat>, <lon>[, <alt>])
```

where

`<lat>`	Latitude, in degrees. This is specified as a real number between -90.0 and 90.0
`<lon>`	Longitude, in degrees. This is specified as a real number between -180.0 and 180.0.
`<alt>`	Altitude, in meters. This is specified as a real number. Specifying the altitude is optional, and it is assumed to be zero when not specified explicitly.

Examples of node coordinates are:

```
(20.2, 0.9, 0.11)
(-22.2110679314668, 132.8618458505577, 0.0)
(150, 200)
```

In MGRS, coordinates are specified using the following format:

```
(<MGRS location string>[, <alt>])
```

Examples of node coordinates are:

```
(04QFJ0020000200, 0.0)
(04QFJ0022000220, 10.0)
```

Node orientation is specified using the following format:

```
<azimuth> <elevation>
```

where

`<azimuth>`	Azimuth, in degrees. This is specified as a real number between 0.0 and 360.0.
`<elevation>`	Elevation, in degrees. This is specified as a real number between -90.0 and 90.0.

Examples of node orientation are:

```
45.0 90.0
0 -25.0
```

The complete specification of a node's position consists of its coordinates followed by its orientation. Specifying node orientation is optional and is assumed to be (0.0, 0.0) when not specified.

Examples of node positions are:

```
(100, 200, 2.5) 45.0 90.0
(25.5, 300.0, 1.0)
(10, 15, 0) 0 0
(75.258934, -127.09378)
(-25.34678, 25.2897654) 0.0 -25.5
```

2.2.4 Node, Network and Interface Identifiers

A node in EXata can represent any of the several devices that connect to a network, such as radio devices, desktop computers, routers, satellites, etc. These nodes can have one or more network interfaces, each of which has its own IP address and subnet mask.

Node Identifiers

Every EXata node has a unique node identifier (or node ID), which is a unique, positive integer. These integers need not be contiguous, so they can be numbered for user readability. For example, if a scenario has three nodes, a user can assign them node IDs 1, 2, and 3. The user can also assign these three nodes node IDs 13, 16, and 159, or, 100, 200, 300.

Node IDs are used in input and output files to refer to specific nodes. The keyword `thru` is used to denote a range of Node IDs. For example, an input file the phrase "`3 thru 7`" refers to the five nodes with node IDs 3, 4, 5, 6, and 7. For simplicity, in the documentation, the node ID is used directly after the word "node" to refer to a node with a certain node ID. For example, "node 1" is used to refer to the node with the node ID 1.

Subnets and Interfaces

An EXata simulation experiment is composed of networks of nodes, which are referred to as subnets. Nodes communicate with each other through connected network interfaces. A node can be a member of multiple subnets and has an interface to each subnet it belongs to. Every node must have at least one network interface. Examples of network interfaces include 802.11b PCMCIA cards, Ethernet adapters, and serial links on routers.

2.2.4.1 EXata Syntax for Subnet Addresses for IPv4 Networks

In IPv4 networks, each network interface is identified by a 32-bit address. The most significant bits of an interface (or host) address are used to identify the network, while the least significant bits identify the hosts within the network. All interfaces on a subnet share a 32-bit subnet mask. If *n* bits are used to identify hosts within a network, the least significant *n* bits of its subnet mask are 0 and all other bits are 1. The network address of a subnet is obtained by applying a bit-wise AND operation to the subnet mask and the IP address of any interface. Thus, the least significant *n* bits of the network address are all 0. If *n* host ID bits are used, then 2^*n* unique addresses can be derived. Out of these, one address is reserved for the network and one is reserved for the broadcast address. The remaining 2^*n*-2 addresses can be used for hosts in the network.

An example of a network IP address is 192.168.0.0, and an example of a corresponding subnet mask is 255.255.255.0. The example subnet mask indicates that 8 bits are used to determine host IP addresses. These hosts would have IP addresses from 192.168.0.1 through 192.168.0.254, for a maximum of 254 hosts on this subnet. 192.168.0.0 is the network address and 192.168.0.255 is the broadcast address for this subnet.

EXata uses a shorthand notation for IPv4 network addresses and subnet masks, which has the following syntax:

```
N<number-of-host-bits>-<network-address>
```

where

`<number-of-host-bits>`	Number of bits used to identify host addresses
`<network-address>`	IP address of the network, where leading zeroes may be omitted

An example of this syntax is N8-1.0. This example specifies that eight bits are reserved for host addresses, which defines the subnet mask to be 255.255.255.0. The network address is 0.0.1.0, because omitted bits are assumed to be zero. N8-1.0, N8-0.1.0, and N8-0.0.1.0 are equivalent. Hosts on this network would have IP addresses from 0.0.1.1 through 0.0.1.254, for a maximum of 254 hosts. 0.0.1.255 is the broadcast address for this subnet.

Table 2-3 lists examples of subnet addresses in the EXata N syntax, the corresponding IP address in dot notation and in slash notation, and the subnet mask for the subnet.

TABLE 2-3. Examples of EXata N Syntax

Subnet Address in N Syntax	IP Address (Dot Notation)	Subnet Mask	IP Address (Slash Notation)
N16-0	0.0.0.0	255.255.0.0	0.0.0.0/16
N2-1.0	0.0.1.0	255.255.255.252	0.0.1.0/30
N8-192.168.0.0	192.168.0.0	255.255.255.0	192.168/24
N24-10.0.0.0	10.0.0.0	255.0.0.0	10/8

Nodes are associated with a subnet by means of the keywords `SUBNET` or `LINK` (see Section 4.2.5 for details). The following declaration indicates that nodes 1 through 10 are in the subnet with address N16-1.0:

```
SUBNET N16-1.0.0 {1 thru 10}
```

The network address for this subnet is 0.1.0.0 and the broadcast address is 0.1.255.255. The subnet mask is 255.255.0.0. The 10 nodes in the subnet are automatically assigned IP addresses as shown in the following table.

Node ID	IP Address
1	0.1.0.1
2	0.1.0.2
3	0.1.0.3
...	....
9	0.1.0.9
10	0.1.0.10

As another example, consider the following declaration that assigns nodes 1, 3, and 5 to the subnet with address N8-192.168.2.0:

```
SUBNET N8-192.168.2.0 {5, 1, 3}
```

The network address for this subnet is 192.168.2.0 and the broadcast address is 192.168.2.255. The subnet mask is 255.255.255.0. The three nodes in the subnet are automatically assigned IP addresses as shown in the following table.

Node ID	IP Address
5	192.168.2.1
1	192.168.2.2
3	192.168.2.3

If a node is in multiple subnets, then each of its interfaces is assigned a unique IP address in the manner described above (see Section 4.2.5 for details).

Notes: **1.** Multiple private networks and secondary IP addresses are not supported in EXata.

2. In a scenario, all network addresses must be distinct, even if different subnet masks are used. Thus addresses N8-1.0 and N16-1.0 can not both be used in a scenario.

2.2.4.2 EXata Syntax for Subnet Addresses for IPv6 Networks

Network, host, and interface addresses in IPv6 networks are 128 bits long. An IPv6 address is usually represented in the format `x:x:x:x:x:x:x:x`, where each x represents a 16-bit hexadecimal value. An example IPv6 address is 1080:0:0:0:8:800:200C:417A. Similar to an IPv4 network address, the most significant bits of an interface (or host) IPv6 address are used to identify the network, while the least significant bits identify the hosts within the network. Usually, the slash notation is used for IPv6 addresses. For example, 1080:0:0:0:8:800:200C:417A/64 indicates that the first (most significant) 64 bits are the network prefix.

In addition, EXata uses a shorthand notation for IPv6 network addresses which is very similar to that of IPv4 addresses. The shorthand notation for IPv6 addresses has the following syntax:

```
N<number-of-host-bits>-<network-address>
```

where

`<number-of-host-bits>`	Number of bits used to identify host addresses
`<network-address>`	IPv6 address of the network, where leading zeroes may be omitted

An example of this syntax is N16-::2.0. This example specifies that 16 bits are reserved for host addresses. Hosts on this network would have IPv6 addresses from ::2:1 to ::2:FFFF, for a maximum of 65535 hosts. (The string “::” is an abbreviation for consecutive 0’s. For example, ::2:1 is equivalent to 0:0:0:0:0:0:2:1, 2000:0:1:: is equivalent to 2000:0:1:0:0:0:0:0, and 100:50::1 is equivalent to 100:50:0:0:0:0:0:1.)

Consider the following declaration which indicates that the subnet with address N16-2000:0:: has nodes 1 through 10:

```
SUBNET N64-2000:0:0:1:: {1 thru 10}
```

The 10 nodes in the subnet are automatically assigned IPv6 addresses as shown in the following table.

Node ID	IPv6 Address
1	2000:0:0:1::1
2	2000:0:0:1::2
3	2000:0:0:1::3
...	....
9	2000:0:0:1::9
10	2000:0:0:1::a

An alternate format for IPv6 addresses is:

<FP><TLA-ID><RES><NLA-ID><SLA-ID><Interface-ID>

where

<FP>	3-bit filed denoting Format Prefix for aggregate global unicast addresses, which is 001
<TLA-ID>	13-bit Top-Level Aggregation (TLA) identifier
<RES>	8-bit filed reserved for future use
<NLA-ID>	24-bit Next-Level Aggregation (NLA) identifier
<SLA-ID>	16-bit Site-Level Aggregation (SLA) identifier
<Interface-ID>	64-bit interface identifier

EXata supports the TLA-NLA-SLA notation for IPv6 addresses. This notation has the following syntax:

TLA-<TLA-ID>.NLA-<NLA-ID>.SLA-<SLA-ID>

where

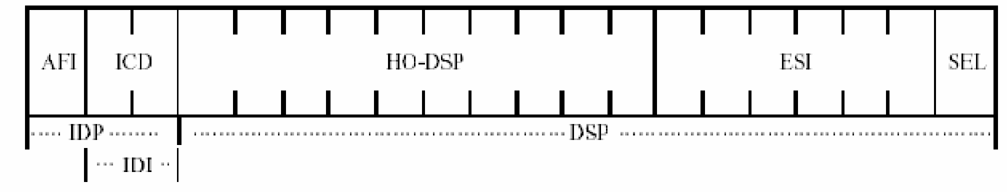
<TLA-ID>	TLA identifier
<NLA-ID>	NLA identifier
<SLA-ID>	SLA identifier

An example of an IPv6 address in this syntax is TLA-1.NLA-2.SLA-1.

Note: The TLA-NLA-SLA format is deprecated and it is recommended that it not be used.

2.2.4.3 EXata Syntax for Subnet Addresses for ATM Networks

Addresses in ATM networks are 20 octets long. An ATM address identifies the location of a single ATM interface and consists of parts shown below.



The Initial Domain Part (IDP) uniquely specifies an administration authority, which has the responsibility for allocating and assigning values of the Domain Specific Part (DSP).

In EXata we consider only the International Code Designator (ICD) Address format. International organizations use this format for address allocation. Similar to the IPv4 and IPv6 address syntax, EXata syntax for ATM addresses also consists of a network prefix and an interface ID.

EXata uses the following syntax for ATM addresses:

```
ICD-<icd-value>.AID-<aid-value>.PTP<ptp-value>
```

where

`<icd-value>`	Integer between 0 and 65535
`<aid-value>`	Integer between 0 and 65535
`<ptp-value>`	Integer between 0 and 65535

An example of an ATM address is ICD-1.AID-1.PTP-1.

2.2.5 Instance Specification

Some parameters in EXata can have multiple instances, such as queues, channels, etc. A specific instance of such a parameter is referred to by an index. If there are *n* instances of a parameter, the index ranges from 0 to *n* - 1. The parameter specification is similar to an array, i.e., the parameter name is followed by an index enclosed in square brackets, "[" and "]".

The following are examples of instances of parameters:

```
IP-QUEUE-PRIORITY-QUEUE-SIZE[1]
PROPAGATION-CHANNEL-FREQUENCY[0]
```

2.2.6 Filename and Path Parameters

Some parameters in input files take file names or paths to directories as their value. When specifying the file name or path, the path relative to the current directory or the absolute path should be included.

The following are examples of values that can be assigned to file name parameters:

```
../../data/terrain/los-angeles-w
./default.fault
C:\scalable\exata\5.1\scenarios\default\default.nodes       (For Windows)
/root/scalable/exata/5.1/scenarios/default/default.nodes    (For UNIX)
```

The following are examples of values that can be assigned to path parameters:

```
../../data/terrain
C:\scalable\exata\5.1\scenarios\default                     (For Windows)
/root/scalable/exata/5.1/scenarios/default                  (For UNIX)
```

2.2.7 Include Command

In some cases, especially if an input file is large, it may be convenient to split the input file into a primary file and one or more secondary files. The secondary files are linked to the primary file by means of the `INCLUDE` command, which has the following format:

```
INCLUDE <filename>
```

where

`<filename>`	Name of the secondary file to be included. The file name should include the path relative to the current directory or the absolute path.

The effect of the command is to replace the command with the contents of the included file.

The `INCLUDE` command can be used in the scenario configuration (.config) and some other supplemental input files.

For example, the configuration file for a wireless scenario may be called “wireless-scenario.config”. The Physical Layer parameters may be specified in a file called “wireless-scenario.phy-config” and the MAC Layer parameters in another file called “wireless-scenario.mac-config”. File “wireless-scenario.config” should include the following lines:

```
INCLUDE ./wireless-scenario.phy-config
INCLUDE ./wireless-scenario.mac-config
```

2.2.8 Random Number Distributions

For some parameters that take numerical values, it is possible to specify a random distribution instead of a specific numerical value. In this case a random value from the specified random distribution is assigned to the parameter. A random distribution is specified by a keyword identifying the distribution followed by distribution parameters.

A random distribution is specified in the following format:

```
<Distribution Identifier> <Parameter List>
```

where

`<Distribution Identifier>`	String identifying the distribution.
`<Parameter List>`	Parameters for the distribution.

The string identifier and parameters for random distributions that can be read from an input file are listed in Table 2-4.

TABLE 2-4. Distribution Identifiers and Parameters

Distribution Name	Distribution Identifier	Parameters	Description
Deterministic	`DET`	• `val`	Always returns the value `val`.
Exponential	`EXP`	• `val`	Returns a value from an exponential distribution whose mean is `val`.
Pareto	`TPD`	• `val1` • `val2` • `alpha`	Returns a value from a truncated Pareto distribution with `val1` as the lower end of the range (= the lower limit of the truncation), `val2` as the upper limit of the truncation, and `alpha` as the shape parameter.
Pareto4	`TPD4`	• `val1` • `val2` • `val3` • `alpha`	Returns a value from a truncated Pareto distribution with `val1` as the lower end of the range, `val2` as the lower limit of the truncation, `val3` as the upper limit of the truncation, and `alpha` as the shape parameter.
Uniform	`UNI`	• `min` • `max`	Returns a value `x`, where `x` is uniformly distributed in the range `min` <= `x` < `max`.

Examples:

`UNI 10 30` : Denotes a uniform distribution in the range 10 to 30

`DET 20MS` : Denotes a deterministic distribution with the value 20 milliseconds.

2.2.9 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.

2.2.9.1 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 2.2.9.2) 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 be at least one space between the qualifier and the parameter name. 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.

2.2.9.2 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.

The following examples illustrate the use of precedence rules.

Example 1: Consider the following declarations in a configuration file:

```
SUBNET N8-1.0 {1, 2, 3}
SUBNET N8-2.0 {3, 4, 5}
ROUTING-PROTOCOL                          AODV
AODV-HELLO-INTERVAL                       10
[2 4 5] AODV-HELLO-INTERVAL               20
[N8-1.0] AODV-HELLO-INTERVAL              30
[0.0.1.3 0.0.2.3] AODV-HELLO-INTERVAL     40
```

The above configuration specifies five nodes in two subnets. Node 3 has interfaces to both subnets, while the other nodes have one interface each. The following table lists the interface addresses of the four nodes (see Section 4.2.5 for details), the value of the parameter `AODV-HELLO-INTERVAL` for each interface and the precedence rules used to arrive at that value.

Node ID	Subnet Address	Interface Address	Value of `AODV-HELLO-INTERVAL`	Precedence Rules
1	N8-1.0	0.0.1.1	30	Subnet > Global
2	N8-1.0	0.0.1.2	30	Subnet > Node > Global
3	N8-1.0	0.0.1.3	40	Interface > Subnet > Global
3	N8-2.0	0.0.2.1	10	Global
4	N8-2.0	0.0.2.2	20	Node > Global
5	N8-2.0	0.0.2.3	40	Interface > Node > Global

Example 2: Consider the following declarations in a configuration file:

```
SUBNET N8-1.0 {1, 2, 3}
SUBNET N8-2.0 {3, 4}
IP-QUEUE-NUM-PRIORITIES   3
IP-QUEUE-PRIORITY-QUEUE-SIZE                    1000
IP-QUEUE-PRIORITY-QUEUE-SIZE[1]                 2000
IP-QUEUE-PRIORITY-QUEUE-SIZE[2]                 3000
[3] IP-QUEUE-PRIORITY-QUEUE-SIZE[2]             4000
[0.0.1.3] IP-QUEUE-PRIORITY-QUEUE-SIZE[2]       5000
[N8-2.0] IP-QUEUE-PRIORITY-QUEUE-SIZE[1]        6000
[4] IP-QUEUE-PRIORITY-QUEUE-SIZE                7000
```

The above configuration specifies four nodes in two subnets. Node 3 has interfaces to both subnets, while the other nodes have one interface each. Each interface has three priority queues, so there are three instances of the parameter `IP-QUEUE-PRIORITY-QUEUE-SIZE` for each interface. The following table lists the interface addresses of the four nodes (see Section 4.2.5 for details) and the values of the parameter IP-QUEUE-PRIORITY-QUEUE-SIZE for each instance at each interface derived from the above declarations and precedence rules.

Node ID	Subnet Address	Interface Address	IP-QUEUE-PRIORITY-QUEUE-SIZE[0]	IP-QUEUE-PRIORITY-QUEUE-SIZE[1]	IP-QUEUE-PRIORITY-QUEUE-SIZE[2]
1	N8-1.0	0.0.1.1	1000	2000	3000
2	N8-1.0	0.0.1.2	1000	2000	3000
3	N8-1.0	0.0.1.3	1000	2000	5000
3	N8-2.0	0.0.2.1	1000	6000	4000
4	N8-2.0	0.0.2.2	7000	6000	3000

2.3 Syntax of Output Files

At the end of a simulation, EXata generates a statistics file containing information for analyzing the behavior of protocols, network performance, etc. The rules that determine the name of the statistics file are explained in Section 2.1.1.2. The statistics file is a plain text file that can be opened by using any text editor. It can also be viewed graphically using EXata Analyzer.

Normally, the simulation runs for the configured simulation time (see parameter SIMULATION-TIME in Section 4.2.1). However, the simulation can be terminated before the configured simulation time (for example, by typing Ctrl+C). A statistics file is generated in either case. The first two lines of the statistics file indicate the configured simulation time and the simulation time when the simulation actually ended. If the simulation is allowed to run for the configured simulation time, then these two entries are identical. The first two lines have the following format:

```
<Node ID>,,,,,Max Configured Simulation Time = <Max-Simulation-Time>
<Node ID>,,,,,Simulation End Time = <Simulation-End-Time>
```

where

<Node ID>	Lowest node ID in the scenario. This is typically 1.
<Max-Simulation-Time>	Maximum configured simulation time, in seconds.The maximum simulation time is configured by setting the parameter SIMULATION-TIME (see Section 4.2.1) in the scenario configuration (.config) file.
<Simulation-End-Time>	Simulation time, in seconds, when the simulation ended. The statistics in the rest of the file are collected from the beginning of simulation to this time.

Each line after the first two lines of the statistics file lists one statistic of a specific protocol. These lines are grouped first by node, then by layers, then by protocol, then by interface, and finally by index. Each line in the statistics file has the following format:

```
<Node ID>, <Interface Address>, <Index>, <Layer>, <Protocol>, <Metric> =
   <Value>
```

where

`<Node ID>`	Node ID of the node where the protocol is running.
`<Interface Address>`	IP address of the interface where the protocol is running. This field is optional and is left blank if it is not applicable. It is usually used only for queues and schedulers.
`<Index>`	Index to distinguish multiple instances of the same protocol running at the node or interface. For example, this can be the port number for Application Layer, queue index for Network Layer, and interface index for MAC and Physical Layer. This field is optional and is left blank if it is not applicable.
`<Layer>`	Layer of the protocol which is printing the statistic.
`<Protocol>`	Name of the protocol which is printing the statistic.
`<Metric>`	Name of the statistic.
`<Value>`	Value of the statistic variable at the end of the simulation.

The parameters that determine which layer and protocol statistics are printed to the statistics file are described in Section 4.2.9. The statistics that are printed for each protocol are described in the model libraries with the protocol's description.

Figure 2-1 shows part of a statistics file.

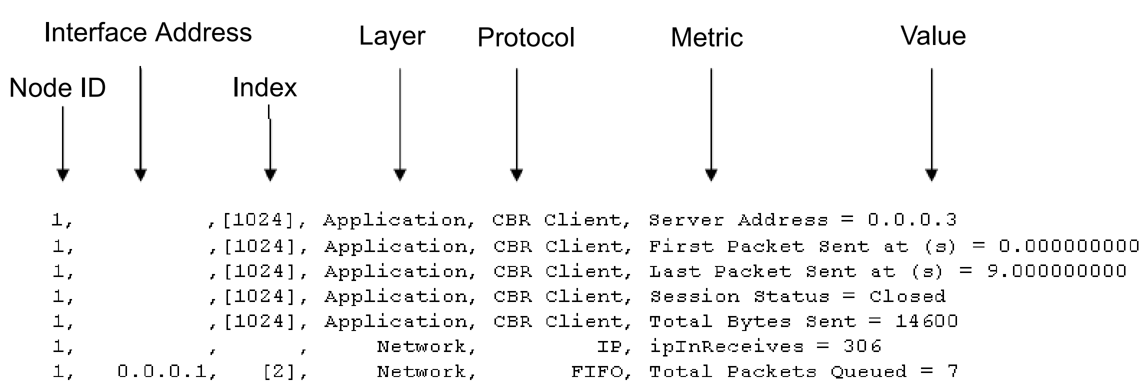


FIGURE 2-1. Example of a Statistics File

3 EXata Architect: Design Mode

In Design mode, Architect provides an easy to use interface for creating network scenarios and setting up simulation parameters. Architect provides simple drag-and-drop functionality to create network topologies and advanced editors to allow fine-grained design of networks. After completing the network design, you can switch to the Visualize mode of Architect to run the simulation and analyze its performance by viewing real-time graphs and animation of the scenario.

When EXata GUI is started, it opens in the Design mode of Architect.

Features of Architect's Design Mode

In Design mode, Architect provides the following features for creating network scenarios:

- Drag-and-drop design of network scenarios.
- Toolsets for devices, links, network components and applications.
- 2D and 3D views of the terrain, including DEM, DTED, and urban terrain features.
- Table view for a quick, comprehensive view of the devices, networks, interfaces, applications and hierarchies in the scenario.
- Specification of mobility models and weather patterns.
- Setup and customization of simulation parameters.
- Properties Editors to customize any protocol layer, device, or application.
- Device Model Editor to build custom devices and network components.
- Hierarchy Editor to build custom platforms and complex network components that contain other network devices.
- Batch experiments to enable simulating the same scenario with different values of network parameters, and comparing simulation results for different combinations of parameter values.
- Settings for running the simulation on multi-core and cluster systems, and for running the simulator on remote, powerful systems while running the GUI on the user's desktop.

This chapter describes the overall layout of Architect, but focuses on the features available in Design mode. Features of the Visualize mode of Architect are described in Chapter 6. Chapter 4 describes in detail how to create scenarios.

Architect's Design mode features are described in the following sections:

- Section 3.1 provides an overview of the components of Architect, with focus on the features accessible in Design mode.
- Section 3.2 describes how to create basic scenarios.
- Section 3.3 describes the various Properties Editors.
- Section 3.4 describes advanced editing features, modifying properties for multiple devices, setting mobility waypoints and weather patterns, and creating hierarchical networks.
- Section 3.5 provides examples for creating custom Network Object Models, Hierarchy Models, and Customizing Toolsets.

3.1 Components of Architect

This section provides an overview of the different components of Architect. By default, EXata GUI opens in Design mode and uses the following active menus, toolbars, panels, and components (see Figure 3-1).

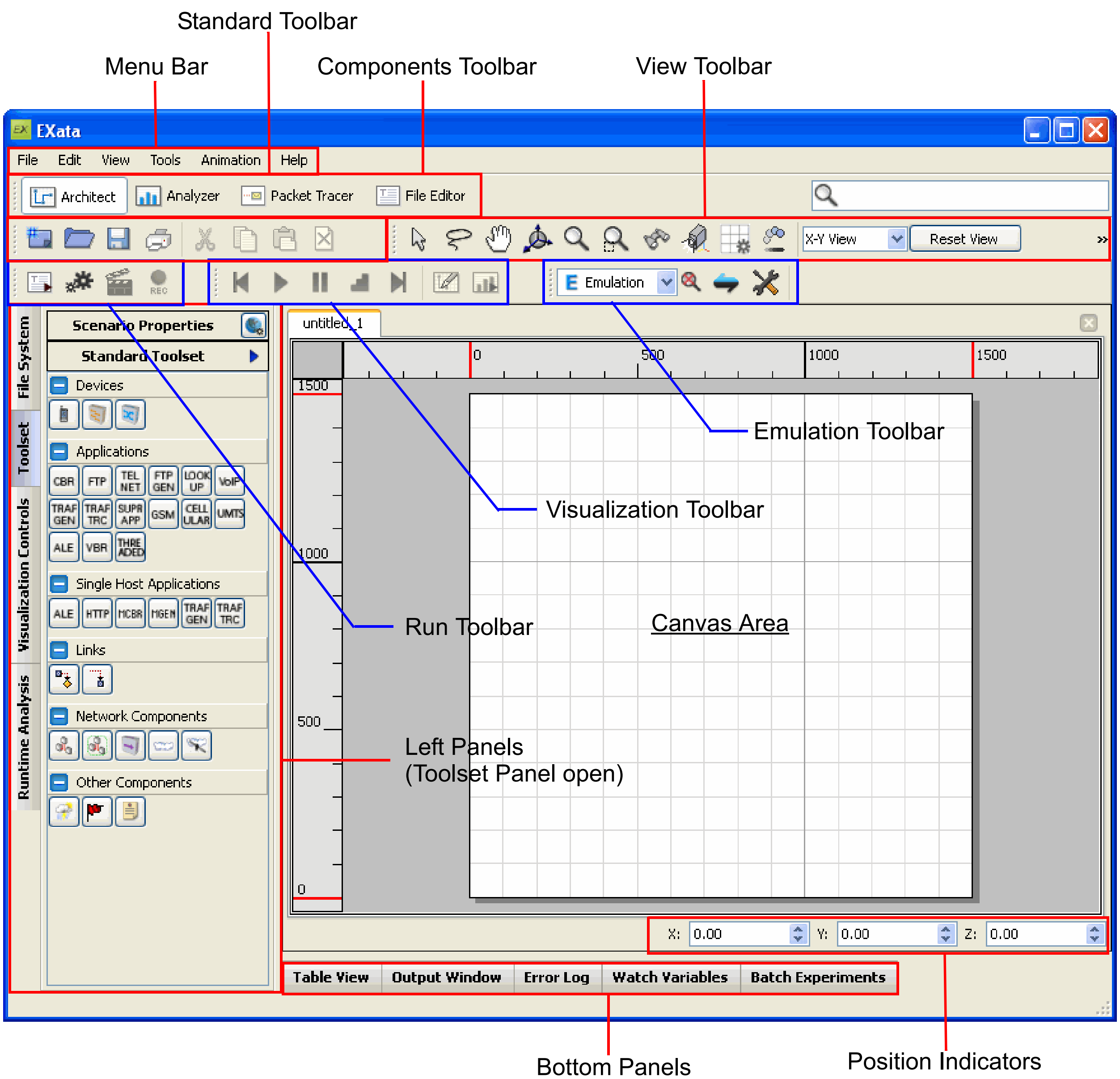


FIGURE 3-1. Architect Layout

3.1.1 Menu Bar

This section describes the menus available from the **Menu** bar.

3.1.1.1 File Menu

The **File** menu provides the following commands for file operations.

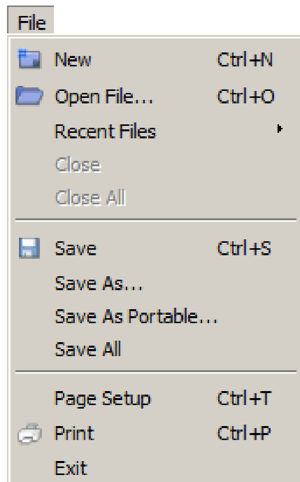


FIGURE 3-2. File Menu

Command	Description
New	Opens a new scenario in a tab on the canvas. New scenarios are called “untitled_1”, “untitled_2”, and so on. This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Open	Opens an existing scenario in a tab on the canvas. A file selection window is launched using which you can select the scenario to open. By default, the file selection window opens with folder the EXATA_HOME\scenarios\user. After the scenario is opened, its name is displayed in the tab. This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Recent Files	Displays a list of recently opened scenarios. Selecting a scenario from this list opens it in a tab on the canvas.
Close	Closes the active scenario. If changes have been made to the scenario since the last time it was saved, the user will be prompted to save the changes. If the last open scenario is closed, a new empty scenario is automatically opened.
Close All	Closes all open scenarios. The user will be prompted to save changes before closing a scenario. After all open scenarios are closed, a new empty scenario is automatically opened.

Command	Description
Save	Saves the active scenario. If the scenario has never been saved before, a file selection window is launched using which the user can specify the name of the scenario and the location where to save it. When saving an existing scenario that opened in architect, the file selected window is not launched and the scenario is changed from where it was being loaded. This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1. **Note:** If there are unsaved changes made in a scenario, an asterisk (“*”) appears after the name of the scenario in the scenario tab. The asterisk(“*”) sign disappears as soon as the **Save** button is clicked, any change made after save commitment will again shows (“*“) sign after scenarios name in the current scenario tab.
Save As	If the scenario is a new scenario, **Save As** works the same way as **Save**; otherwise, it saves a copy of the active scenario. A file selection window is launched using which you can specify the name of the copy and the location where to save it. The original scenario is closed and the copy of the scenario is opened.
Save As Portable	Saves a copy of the active scenario in the same way as **Save As** except all referenced files (except terrain files) used in the scenario are copied into the scenario folder. The file path names in the scenario configuration (.config) file are updated to reference the files saved in the scenario folder. The **Save As Portable** option allows the user to save a scenario that can be run from any location. This option copies all the dependent files used by the scenario locally to the folder where the scenario is saved.
Save All	Performs a **Save** operation on all open scenarios.
Page Setup	Opens a dialog box to set printing options.
Print	Prints a hard copy of the displayed scenario including the blank area surrounding the canvas. This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Exit	Exits from EXata GUI. If there are any unsaved changes in any of the open scenarios, the user will be prompted to save them.

3.1.1.2 Edit Menu

The **Edit** menu provides the following commands for performing editing operations for scenario object on the canvas:

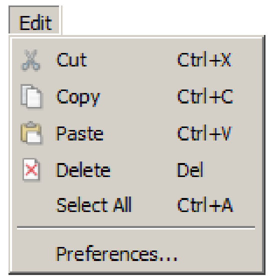


FIGURE 3-3. Edit Menu

Command	Description
Cut	Cuts the selected object(s). If an object is cut, all links attached to it are deleted. If both ends of a link are cut along with a link, the link will be cut and recreated on a paste. This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Copy	Copies the selected object(s). A link can be copied only if the objects that it connects are also copied. This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Paste	Pastes a previously cut or copied object(s). This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Delete	Deletes the selected object(s). This option can also be selected from the **Standard** toolbar. See Section 3.1.2.1.
Select All	Selects all objects in the scenario.
Preferences	Used to set the following preferences: • **Text Editor**: Select the text editing tool for viewing text files associated with a scenario. The default text editing tool is File Editor. • **Port**: Select the TCP port used for communicating with Simulator. • **Terrain**: Enable correction for missing or corrupt data in DTED files. If the elevation for a point in the DTED file is negative and differs from the average elevation in the terrain file by more than the specified amount, then it is replaced by the average of the elevations of the nearest north, south, east, and west neighbor points that have valid elevation data. • **View**: Enable highlighting of nodes when the mouse is hovered over them in Visualize mode.

3.1.1.3 View Menu

The **View** menu provides the following controls for displaying toolbars and commands to configure display and camera settings:

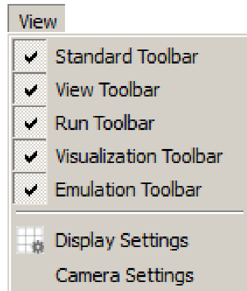


FIGURE 3-4. View Menu

Command	Description
Standard Toolbar	Controls whether the **Standard** toolbar is displayed. See Section 3.1.2.1 for a description of the Standard toolbar.
View Toolbar	Controls whether the **View** toolbar is displayed. See Section 3.1.2.2 for a description of the **View** toolbar.
Run Toolbar	Controls whether the **Run** toolbar is displayed. See Section 3.1.2.3 for a description of the **Run** toolbar.
Visualization Toolbar	Controls whether the **Visualization** toolbar is displayed. See Section 3.1.2.4 for a description of the **Visualization** toolbar.
Emulation Toolbar	Controls whether the **Emulation** toolbar is displayed. See Section 3.1.2.5 for a description of the **Emulation** toolbar.
Display Settings	Used to customize the look and feel of a scenario. See Section 3.1.1.3.1for a description of the **Display Settings** dialog box.
Camera Settings	Used to customize camera options for 3D view. See Section 3.1.1.3.2 for a description of the **Camera Settings** dialog box. This option is enabled only in the 3D view and the 3D window of the Split Screen view.

3.1.1.3.1 Display Settings

The **Display Settings** dialog box is used to control the look and feel of the scenario. The configurable options are categorized under three tabs: **General**, **Light**, and **Terrain**.

General Tab

The **General** tab, shown in Figure 3-5, is used to configure the display of scenario elements on the canvas. When checked, the selected option will be displayed.

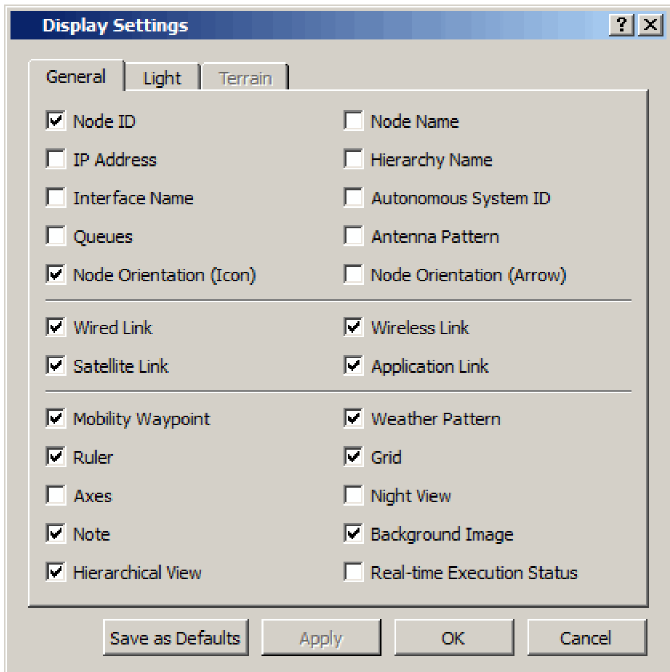


FIGURE 3-5. Display Settings: General Tab

Scene Properties	Default	Description
Node ID	✓	Used to display or hide node IDs in the scenario.
Node Name		Used to display or hide host names.
IP Address		Used to display or hide the IP address of all the interfaces and subnets.
Hierarchy Name		Used to display or hide hierarchy names.
Interface Name		Used to display or hide interface names.
Autonomous System ID		Used to display or hide autonomous system IDs for hierarchies.
Queues		Used to display or hide packet queues of nodes in the scenario. Queues are displayed only in Visualize mode.
Antenna Pattern		Used to display or hide the antenna patterns for nodes.
Node Orientation (Icon)		Used to align each node icon along the node's orientation (specified by the azimuth and elevation angles).
Node Orientation (Arrow)		Used to display a red arrow from each node icon representing the node's orientation (specified by the azimuth and elevation angles).

Scene Properties	Default	Description
Wired Link	✓	Used to display or hide wired links in the scenario.
Wireless Link	✓	Used to display or hide the wireless subnet icons and wireless links in the scenario.
Satellite Link	✓	Used to display or hide the satellite links in the scenario.
Application Link	✓	Used to display or hide application links in the scenario.
Mobility Waypoint	✓	Used to display or hide the mobility waypoints in the scenario.
Weather Pattern	✓	Used to display or hide the weather patterns in the scenario.
Ruler	✓	Used to display or hide the rulers on the canvas.
Grid	✓	Used to display or hide the grid on the canvas.
Axes		Used to display or hide axes in the left bottom corner of the central workspace. The axes show the current direction of X, Y and Z-axis.
Night View		Used to toggle between day and night views. This option is enabled only if the coordinate system is Latitude-Longitude. If night view is selected, stars and moon are displayed.
Note	✓	Used to display or hide notes.
Background Image	✓	Used to display or hide the background image specified for the scenario.
Hierarchical View	✓	Used to display the scenario in a hierarchical or flattened view (if the scenario has hierarchies). When this option is selected, hierarchies are represented by hierarchy icons. When this option is deselected, all nodes and network components within hierarchies are displayed in a flattened view instead of a hierarchical view. **Note:** Changes to the Hierarchical View option will take effect only when you reload the scenario.
Real-time Execution Status		Used to display or hide the Real-time Execution Status indicator in the Visualization Controls panel (see Section 6.5.3).

Note: The display settings are scenario-specific and changes made for one scenario do not apply to other scenarios.

To save the current display settings and use them as default settings in future sessions of Architect, click **Save as Defaults**.

Light Tab

The **Light** tab, shown in Figure 3-6, is used to configure the light settings of the scenario.

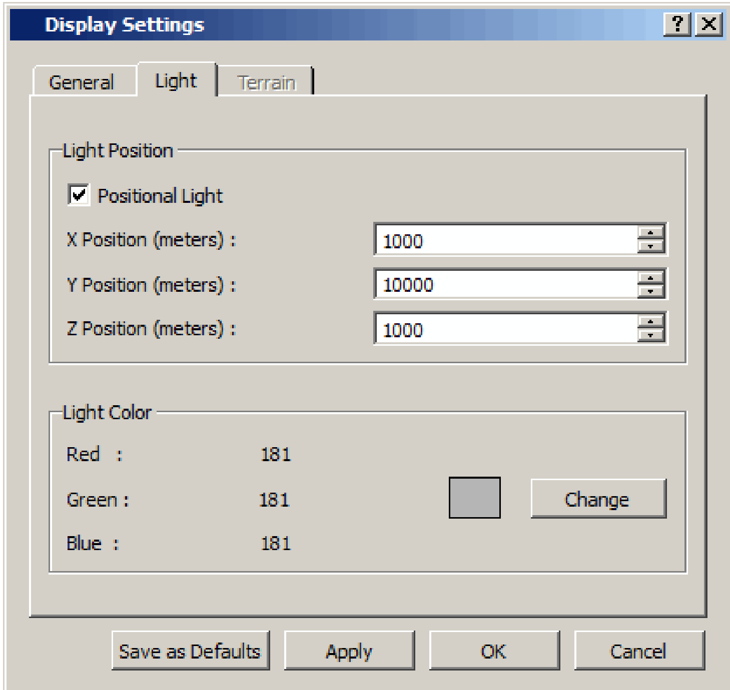


FIGURE 3-6. Display Settings: Light Tab

The light settings can be customized as follows:

- To use positional light, check the box **Positional Light**. By default, light is non-positional (or natural). Non-positional light is similar to sunlight in that all light rays are parallel to each other. Non-positional light produces a more realistic shading effect than positional light, but it is computational intensive.

  If **Positional Light** is checked, fields to enter the light source position are displayed. Enter the coordinates of the light source in the **X Position**, **Y Position**, and **Z Position** fields.

- You can customize the light color by clicking the **Change** button. By default, the light color is gray.

Note: The light settings are effective only in 3D view.

Terrain Tab

The **Terrain** tab, shown in Figure 3-7, is used to configure terrain textures of the scenario. This tab is enabled only if a terrain file is loaded into the scenario (see Section 4.2.2).

Textures are used in a scenario for realism. Architect automatically generates a textured terrain based on elevation points. One texture file is created for each terrain file in the scenario. These files are named genTexture0.bmp, genTexture1.bmp, genTexture2.bmp, etc., and are saved in the directory EXATA_HOME\gui\icons\3Dvisualizer.

You can customize the terrain texture by using the **Terrain** tab.

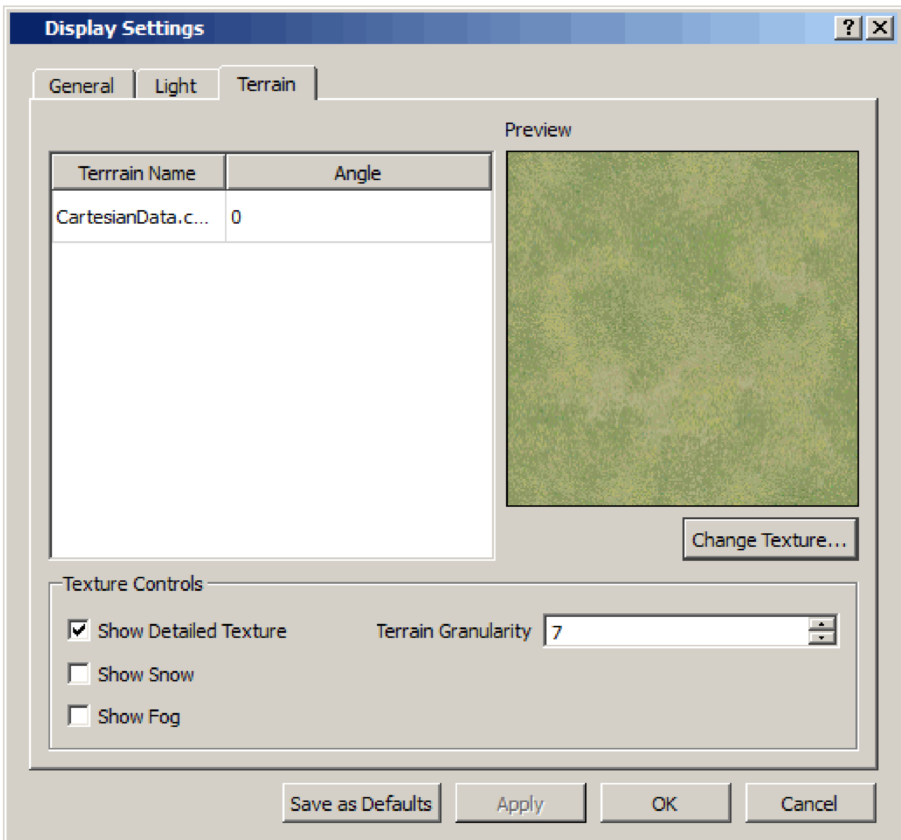


FIGURE 3-7. Display Settings: Terrain Tab

The **Terrain Name** column lists all the terrain files used in the scenario. The angle used for the texture for on the terrain is displayed in the **Angle** column. You can customize the terrain as follows:

Terrain Properties	Description
Angle	The angle specifies the angle of rotation of the texture or image over the terrain surface. You can change the angle used for the texture by entering the desired value in this field. This is particularly useful when the scenario uses multiple terrain files.
Change Texture	Click on this button to select a texture file from the file explorer that opens up. The texture currently assigned to the terrain is displayed in the preview window.
Show Detailed Texture	Used to apply a detailed texture over the terrain. If this option is selected, Architect generates texture for the terrain in real time while incorporating the elevation in different areas for more realistic visualization. The texture is overlaid on the terrain surface. This option can be used to provide some roughness over a terrain by overlaying a texture upon an original texture. **Note**: This option is recommended only for faster processor systems.
Show Snow	Used to hide or display snow on mountain peaks in the terrain.
Show Fog	Used to hide or display fog in valleys in the terrain.
Terrain Granularity	Used to configure the resolution (sharpness) of the terrain display. This field is displayed only if the **Show Detailed Texture** box is checked. The lower the terrain granularity, the sharper the terrain display. **Note**: This option is recommended only for faster processor systems.
Snow Granularity	Used to control the amount of snow. This field is displayed only if the **Show Snow** box is checked. Higher the snow granularity, more the quantity of snow.
Fog Depth	Used to control the density of fog in the scene. This field is displayed only if the **Show Fog** box is checked. Higher the fog depth, denser the fog.

3.1.1.3.2 Camera Settings

The **Camera Settings** dialog box, shown in Figure 3-8, contains options to customize the camera in 3D view. To open the **Camera Settings** dialog box, select **View > Camera Settings** from the menu bar. (This option is enabled only in 3D view.)

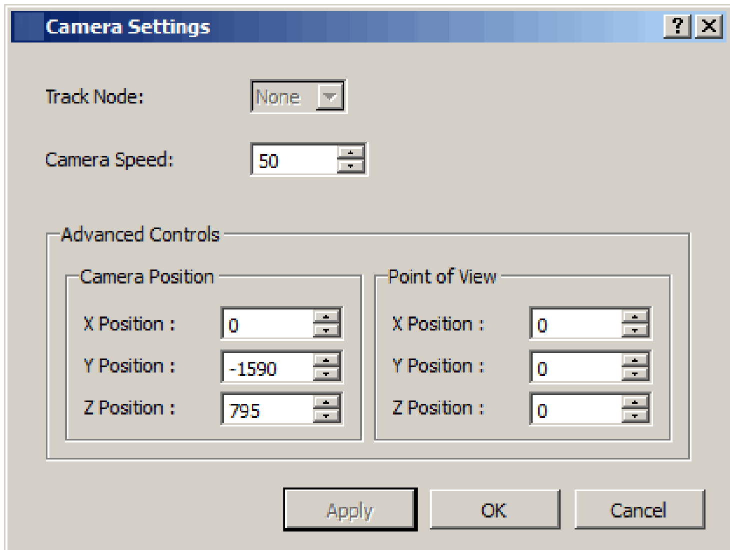


FIGURE 3-8. Camera Settings Dialog Box

The following camera optimization options are available from the **Camera Settings** dialog box:

Camera Settings	Description
Track Node	This option links the camera to a specified node so that the node always remains in view. Select the node to track from the list. You can use the zoom and rotate view controls (see Section 3.1.2.2) with this option. **Note:** In **Track Node** mode, Pan and Free Camera controls can not be used. To exit Track Node mode, set Track Node to None.
Camera Speed	The **Camera Speed** determines the speed of the camera when the arrow keys are used in Free Camera mode (see Section 3.1.2.2). The range of values for this field is 1 to 4000.
Advanced Controls	This option allows you to set a static camera position and direction of view. If the Advanced Controls box is checked, you can specify the X, Y, and Z coordinates of the camera position and the point of view. The direction of view is the straight line from the camera position to the point of view.

3.1.1.4 Tools Menu

The **Tools** menu provides the following commands to launch various editors for configuring a scenario:

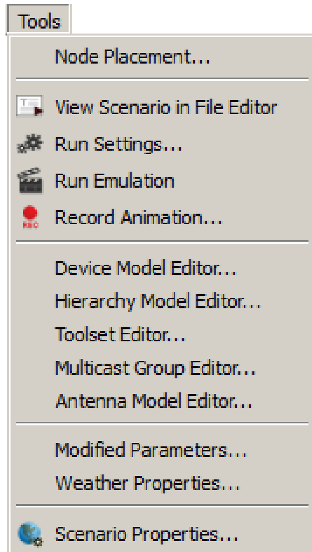


FIGURE 3-9. Tools Menu

Command	Description
Node Placement	Launches the **Node Placement Wizard** which is used for automatic node placement. The **Node Placement Wizard** is explained in Section 4.2.3.
View Scenario in File Editor	Opens the scenario configuration (.config) file of the active scenario in the text editor that has been selected in the **Preferences** dialog (see Section 3.1.1.3). (By default, the file opens in File Editor). This option can also be selected by clicking on the **View Scenario in File Editor** button in the Run toolbar (See Section 3.1.2.3).
Run Settings	Used to open the **Run Settings Editor** which is used to specify the number of processors used for running the simulation and to select the run mode (local or remote). See Section C.2.3 for details. This option can also be selected from the Run toolbar. See Section 3.1.2.3.
Run Emulation	Used to initialize a live simulation of the active scenario. This option can also be selected from the **Run** toolbar. See Section 3.1.2.3. **Note:** This command only initializes the simulation. The simulation and animation do not start until you press the **Play** button in the **Visualization** toolbar (see Section 3.1.2.4).
Record Animation	Used to run the active scenario and record the animation trace in a file, either interactively or non-interactively. All the animation events are printed to an animation trace file. If the interactive mode is selected, the animation is also displayed while recording. If the non-interactive mode is selected, animation is not displayed. This option can also be selected from the **Run** toolbar. See Section 3.1.2.3.
Device Model Editor	Launches the **Device Model Editor** using which you can create custom network-type and device-type models. See Section 3.5.1 for details.
Hierarchy Model Editor	Launches the **Hierarchy Model Editor** using which you can create custom hierarchy models. See Section 3.5.2 for details.

Command	Description
Toolset Editor	Launches the **Toolset Editor** using which you can modify the standard toolset and create new toolsets. See Section 3.5.3 for details.
Multicast Group Editor	Used to create and import multicast groups. See Section 4.2.8.3.3.2.1 for details.
Antenna Model Editor	Launches the **Antenna Model Editor** which is used to import, create, and modify antenna models. See *Wireless Model Library* for details
Weather Properties	Launches the **Weather Properties Editor** which is used to configure weather pattern properties.
Modified Parameters	Launches the **Modified Parameters** window, which lists all scenario parameters that have been set to values that are different from their default values. See Section 3.4.8 for details.
Scenario Properties	Launches the **Scenario Properties Editor** which is used to set scenario level properties. See Section 3.3.2 for details.

3.1.1.5 Animation Menu

The **Animation** menu contains access to six configuration choices: choose animation colors, configure step setting, set communication interval, set event filters, set layer filters, and view dynamic statistics.

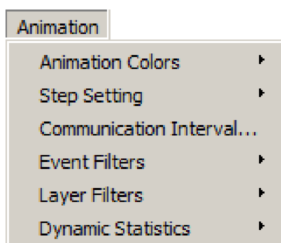


FIGURE 3-10. Animation Menu

Note: The **Animation** menu is enabled only in Visualize mode. See Section 6.3 for details.

3.1.1.6 Help Menu

The **Help** menu provides the following commands to obtain help on EXata GUI:

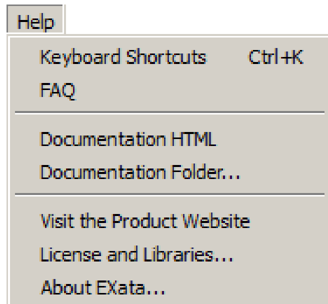


FIGURE 3-11. Help Menu

Command	Description
Keyboard Shortcuts	Displays the keyboard shortcuts for various commands.
FAQ	Opens the FAQ page at the Scalable Network Technologies website.
Documentation HTML	Opens the product's on-line help in a browser window.
Documentation Folder	Opens the folder where all documentation files in PDF format are stored.
Visit the Product Website	Opens the EXata product page at the Scalable Network Technologies website.
License and Libraries	Set, view, and troubleshoot the application license through the following tabs: • **Status**: Provides an overview of the EXata license status and lists the EXata model libraries information, including source file availability and license information. • **Troubleshooting**: Provides an error message relating to a license and steps to correct the error, and procedures on how to generate a license information log that can be used by license support to troubleshoot any issues. See Appendix A for details.
About EXata	Opens the product information page.

3.1.2 Toolbars

This section describes the toolbars available in Architect.

3.1.2.1 Standard Toolbar

The **Standard** toolbar contains buttons (from left to right) to create, open, save, print, cut, copy, paste and delete scenario components.

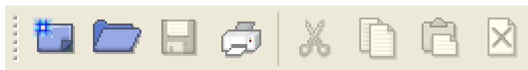
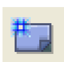
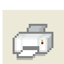
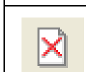


FIGURE 3-12. Standard Toolbar

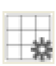
Button	Command	Description
	**New**	Performs the same function as the **File > New** command (see Section 3.1.1.1). It creates a new scenario in a tab on the canvas.
	**Open File**	Performs the same function as the **File > Open** command (see Section 3.1.1.1). It opens an existing scenario in a tab on the canvas. A file selection window is launched using which you can select the scenario to open.
	**Save Scenario**	Performs the same function as the **File > Save** command (see Section 3.1.1.1). If the scenario has never been saved before, a file selection window is launched using which you can specify the name of the scenario and the location where to save it.
	**Print**	Performs the same function as the **File > Print** command (see Section 3.1.1.1). It prints a hard copy of the displayed scenario.
	**Cut**	Performs the same function as the **Edit > Cut** command (see Section 3.1.1.2). It cuts the selected object(s).
	**Copy**	Performs the same function as the **Edit > Copy** command (see Section 3.1.1.2). It copies the selected object(s).
	**Paste**	Performs the same function as the **Edit > Paste** command (see Section 3.1.1.2). It pastes the last cut or copied object(s).
	**Delete**	Performs the same function as the **Edit > Delete** command (see Section 3.1.1.2). It deletes the selected object(s).

3.1.2.2 View Toolbar

The **View** toolbar provides controls for manipulating the scenario view on the canvas.

FIGURE 3-13. View Toolbar

Button	Function	Description
	**Select**	Used to select an object (device, network component, link, etc.) in the scenario. Select this button and click on the object. The selected object will be highlighted. If a node is selected, the coordinates of the selected node are displayed in the Position Indicators (see Section 3.1.4.1 for details of Position Indicators). You can also create a rectangular selection area by clicking and dragging. All objects in the area are selected. The keyboard shortcut for this button is the “S” or “s” key.
	**Lasso**	Used to select multiple objects in a scenario. Select this button, click on the canvas (while holding down the mouse button) drag around the desired objects. The selected objects are also highlighted in the Table View. This tool is useful for multiple selection in an irregular fashion (free-hand selection).
	**Pan**	Used to move the scenario display in the central workspace. To pan a scenario display, move the mouse in the desired direction while keeping the left mouse button pressed. The keyboard shortcut for this button is the “P” or “p” key. If this mode is enabled, the four arrow keys can also be used to move the display.
	**Rotate**	Used to rotate the scenario display around the center of the central workspace. This mode is enabled only in the 3D view and the 3D window of the Split Screen view. To rotate a scenario display, move the mouse in the desired direction while keeping the left mouse button pressed. The keyboard shortcut for this button is the “R” or “r” key. If this mode is enabled, the four arrow keys can also be used to rotate the display.
	**Zoom**	Used to zoom-in and zoom-out in the scenario display. To zoom in or out in a scenario display, move the mouse while keeping the left mouse button pressed. Moving the mouse down zooms out and moving the mouse up zooms in. You can also use the mouse scroll wheel to perform the same function. The keyboard shortcut for this button is the “Z” or “z” key. If this mode is enabled, the up and down arrow keys can also be used to zoom in or out.

Button	Function	Description
	**Region Zoom**	Used to select a region for zooming. This mode is enabled only in the 2D X-Y, Y-Z, and X-Z views. To select a region to zoom in, click on this button and select a rectangular region in the scenario display using the mouse, while keeping the left mouse button pressed. A dotted rectangle outlines the selected region while you drag the mouse. When you release the mouse, the selected region is scaled to the whole view area. The keyboard shortcut for this button is the “A” or “a” key.
	**Binoculars**	Used to magnify a part of the canvas. This mode is enabled only in the 3D View and the 3D window of the Split Screen view. When this mode is enabled, a circle representing a lens of a pair of binoculars is displayed. Initially, the binoculars are focused such that the central region of the canvas is displayed in the circle. Use the mouse (with the left button pressed) to change the direction of view. Use the arrow keys to change the region on which the binoculars are focused. The keyboard shortcut for this button is the “B” or “b” key.
	**Free Camera**	Used to select the **Free Camera** mode. This mode is enabled only in the 3D View and the 3D window of the Split Screen view. In this mode, you can control the direction of view and position of the camera. Use the mouse (with the left button pressed) to change the direction of view. Use the arrow keys to change the position of the camera. The keyboard shortcut for this button is the “F” or “f” key.
	**Open Display Settings**	Used to open the **Display Settings** window. Performs the same function as the **View > Display Settings** (see Section 3.1.1.3).
	**Turn On/Off Motion in Design Mode**	Used to continuously refresh the screen to show 3D motion effects (such as water flows and rotations of a helicopter rotor) in Design mode. In Visualize mode, the 3D motion effects are always enabled and cannot be disabled.
	**Change View**	This menu allows you to switch between X -Y, Y - Z, X - Z, 3D, and Split Screen views. In Split Screen view, the workspace is split into four quadrants which display the X-Y, Y-Z, X-Z, and 3D views.
	**Reset View**	The **Reset View** button is used to reset the current camera position to the position when the scenario was loaded. Clicking the **Reset View** button once changes the view to the initial view.

Button	Function	Description
	**Saved Views**	The **Saved Views** button is used to switch between different views of the scenario. Pressing this button displays a list of saved views. To change the scenario view displayed on the canvas, select a view from this list. See Section 3.4.2 for details. **Note:** Switching between saved views is enabled only in the 3D View and the 3D window of the split Screen view.
	**Overview**	The **Overview** button provides a convenient way to quickly change the camera position from one position in the scenario to another. Press the **Overview** button and a window (called the overview window) opens in the bottom right corner of the central workspace (see Figure 3-14). The overview window displays the entire scenario in X-Y view. A red dot inside a small yellow rectangle in the overview window represents the current position of the camera. A red line starting from the red dot represents the current direction of view. You can change the camera position by dragging the yellow rectangle using the mouse to a different point in the overview window. This will update the view in the main display. You can use all the **View Control** buttons in the main display as usual. The position of the rectangle in the overview window is updated to reflect the new camera position in the main display. **Note:** The Overview feature is enabled only in the 3D View and the 3D window of the split Screen view.

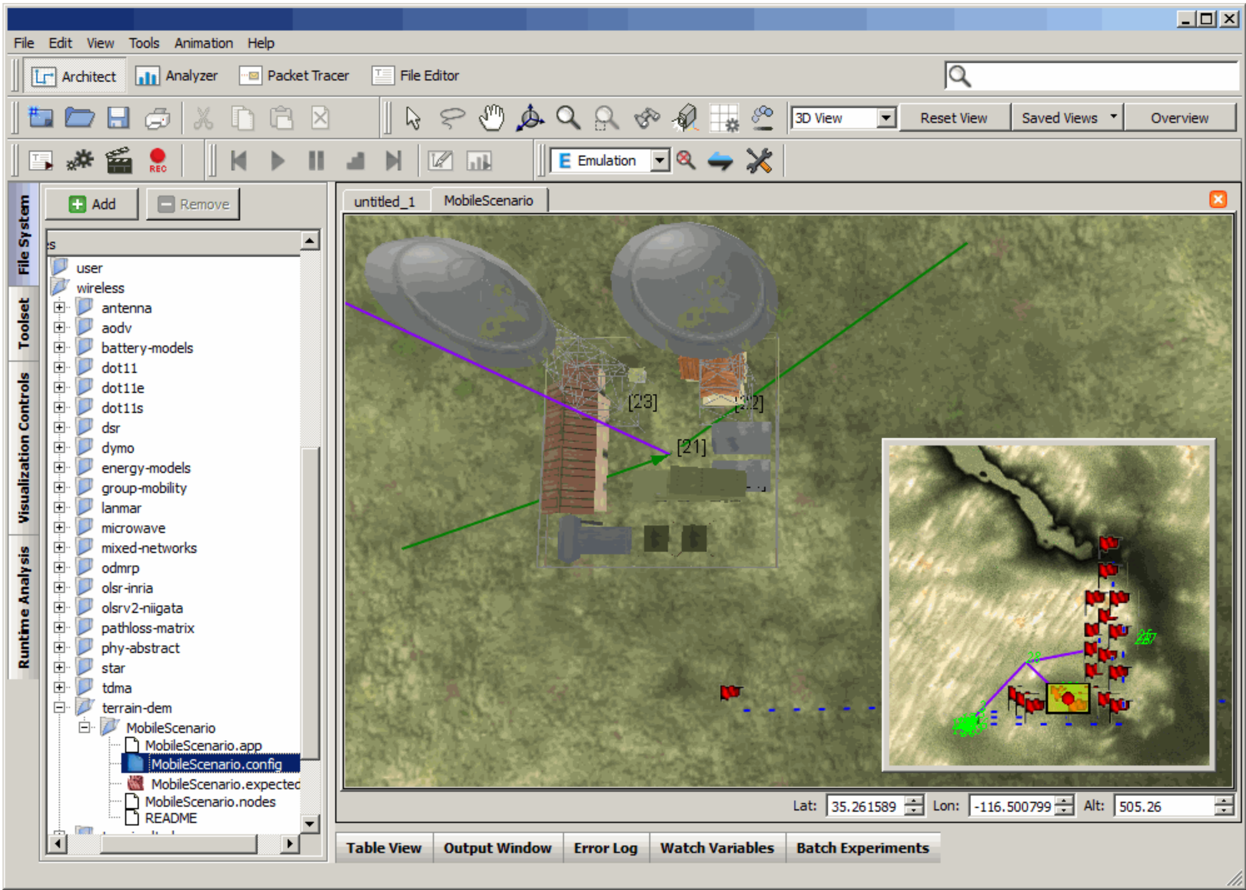
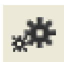


FIGURE 3-14. Overview Feature

3.1.2.3 Run Toolbar

The **Run** toolbar is used to access the run settings editor, initialize a live simulation, and to record the animation trace in a file. The following table describes the buttons of the Run toolbar.

FIGURE 3-15. Run Toolbar

Button	Function	Description
	**View Scenario in Text Editor**	Used to open the scenario configuration (.config) file for the scenario in the text editor that has been selected in the **Preferences** dialog (see Section 3.1.1.3). (By default, the file opens in File Editor). This button performs the same function as the **Tools > View Scenario in File Editor** command. See Section 3.1.1.4.
	**Run Settings**	Used to open the **Run Settings Editor** which is used to specify the number of processors used for running the simulation and to select the run mode (local or remote). See Section C.2.3 for details. This button performs the same function as the **Tools > Run Settings** command. See Section 3.1.1.4.
	**Run Simulation**	Used to initialize a live simulation of the active scenario. This button performs the same function as the **Tools > Run Simulation** command. See Section 3.1.1.4. **Note:** This button only initializes the simulation. The simulation and animation do not start until you press the **Play** button in the **Visualization** toolbar.
	**Record Animation Trace**	Used to run the active scenario and record the animation trace in a file, either interactively or non-interactively. All the animation events are printed to an animation trace file. If the interactive mode is selected, the animation is also displayed while recording. If the non-interactive mode is selected, animation is not displayed. This button performs the same function as the **Tools > Record Animation** command. See Section 3.1.1.4.

3.1.2.4 Visualization Toolbar

The **Visualization** toolbar is used to control the runtime features during animation and is only active in the Visualize mode.

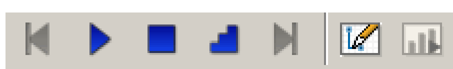


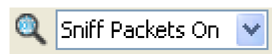
FIGURE 3-16. Visualization Toolbar

Note: The **Visualization** toolbar is enabled only in Visualize mode. See Section 6.4.4 for details.

3.1.2.5 Emulation Toolbar

The **Emulation** toolbar is used for configuring emulation features.

FIGURE 3-17. Emulation Toolbar

Button	Function	Description
	**Select Mode**	This pull-down menu allows you to select the EXata mode (emulation or simulation). **Note:** This feature is enabled only in Design mode.
	**Select Packet Sniffer Node(s)**	Clicking this button enables a pull-down menu from which you can select the node(s) to sniff on.  You can select any one node or all nodes to sniff on. **Note:** This feature is enabled in both Design and Visualize modes. See Section 4.3.3 for details.
	**Manage External Connections**	Launches an editor to manage connections between EXata nodes and operational hosts. **Note:** This feature is enabled in both Design and Visualize modes. See Section 4.3.1 for details.
	**Advanced Emulation Configuration**	Launches the **Advanced Emulation Configuration** editor. **Note:** Some features available from this editor are enabled only in Design mode. The remaining features are enabled in both Design and Visualize modes. See Section 4.3.5 for details.

3.1.3 Left Panels

The following panels are available to the left of the canvas:

- File System
- Toolset
- Visualization Controls
- Runtime Analysis

Note: These four panels occupy the same space and at most one of them can be open at any time. By default, the **Toolset** panel is open.

3.1.3.1 File System Panel

The File System is a tree-based view of directories mounted in the EXata GUI. The **File System** panel can be opened or closed by clicking on the tab at the left of the Architect. By default, the **File System** panel is closed. The following directories are mounted by default (assuming that EXata is installed at the default location):

On Windows:

- C:/scalable/exata/5.1 (Directory where EXata is installed)
- C:/scalable/exata/5.1/scenarios (Directory containing pre-configured scenarios)
- C:/scalable/exata/5.1/scenarios/user (Directory for user's scenarios)

On Linux:

- ~/scalable/exata/5.1 (Directory where EXata is installed)
- ~/scalable/exata/5.1/scenarios (Directory containing pre-configured scenarios)
- ~/scalable/exata/5.1/scenarios/user (Directory for user's scenarios)

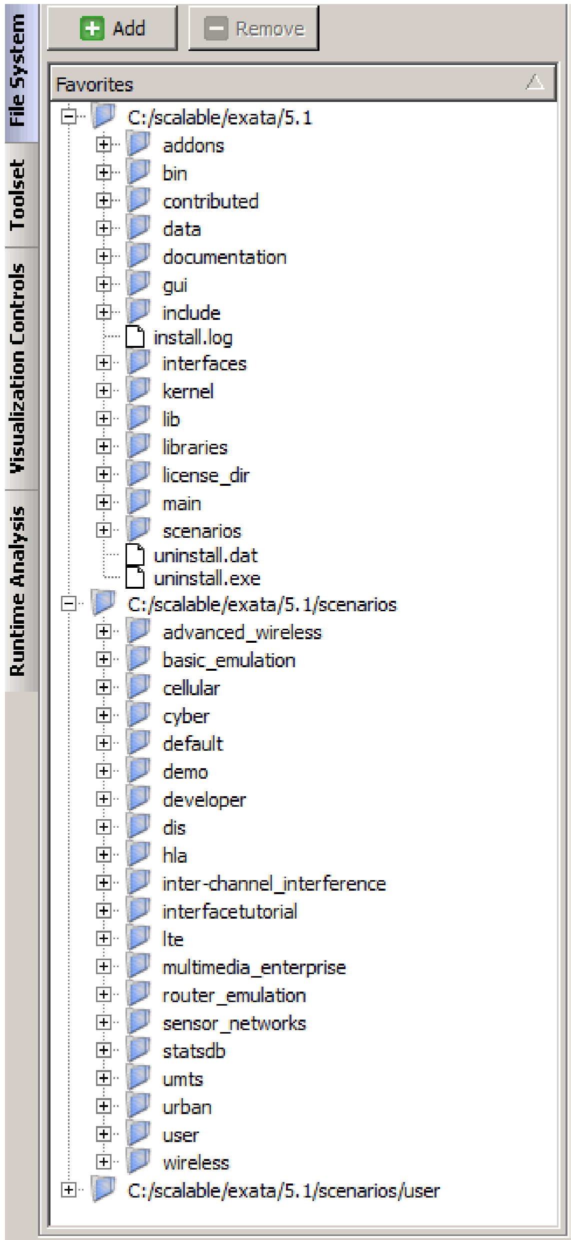


FIGURE 3-18. File System Panel

Directory Operations

1. To mount another directory to the File System, click the **Add** button. This opens a directory selection window from which you can select the directory to mount.
2. To remove a mounted directory, highlight it and click the **Remove** button.

> **Notes:** 1. Removing a directory deletes it from the disk.
>
> 2. Directories mounted by default (C:/scalable/exata/5.1, C:/scalable/exata/5.1/scenarios, and C:/scalable/exata/5.1/scenarios/user on Windows and ~/scalable/exata/5.1, ~/scalable/exata/5.1/scenarios, and ~/scalable/exata/5.1/scenarios/user on Linux) can not be removed.

3. To change the order in which the directories are displayed, click on the **Favorites** header.
4. To refresh the directory listings under a mounted directory, right-click on it and select **Refresh**. (Only mounted directories can be refreshed; subdirectories of a mounted directory can not be refreshed.)

Context-sensitive File Menus

Double-clicking a file opens it in the appropriate application, depending on the file extension:

.config	These are scenario configuration files and open in Architect's Design mode.
.ani	These are animation trace files and open in Architect's Visualize mode.
.stat	These are statistics files and open in Analyzer.
All other extensions	These files open in text editor that has been selected in the **Preferences** dialog (see Section 3.1.1.3). (By default, the files open in File Editor).

In addition, right-clicking on a file name opens a menu which depends on the type of the file (see Figure 3-19).

The following commands are available for all file types:

- **Edit as Text**: Opens the file in the text editing tool selected in the **Preferences** dialog (see Section 3.1.1.3).
- **Delete**: Deletes the selected file or folder from the hard drive.
- **Open Containing Folder**: Opens the folder where the selected file is located.

The following additional command is available for ".config" files:

- **Open**: Opens the scenario in Design mode of Architect.

The following additional command is available for ".ani" files:

- **Run**: Opens the animation trace file in Visualize mode of Architect.

The following additional command is available for ".stat" files:

- **Analyze**: Opens the statistics file in Analyzer.

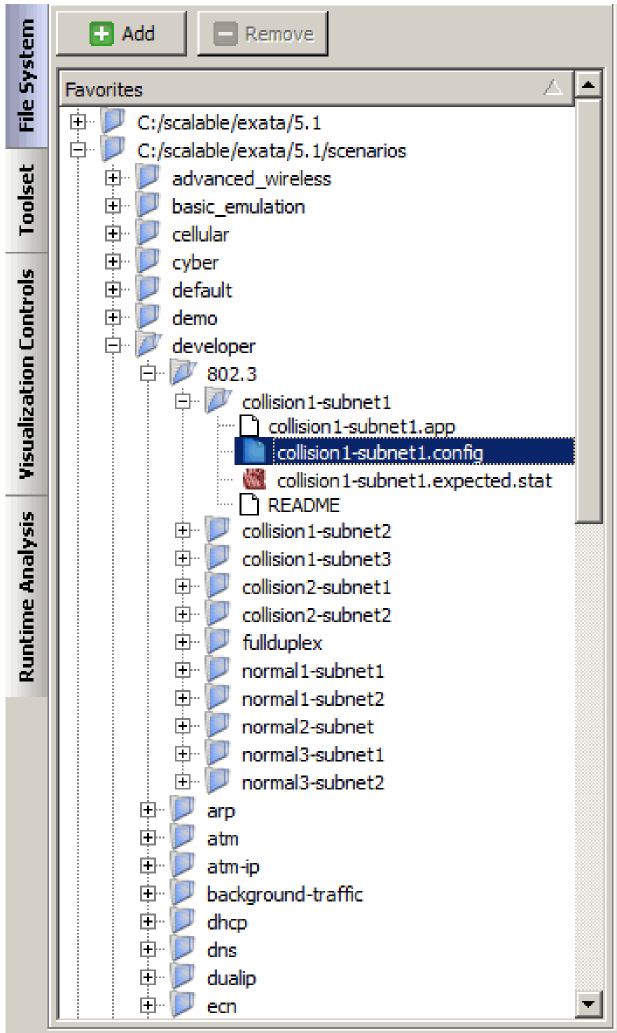


FIGURE 3-19. Context-sensitive Menu in File System Panel

3.1.3.2 Toolset Panel

The Toolset panel provides buttons for the most commonly used scenario components. The Toolset panel can be opened or closed by clicking on the tab at the left.

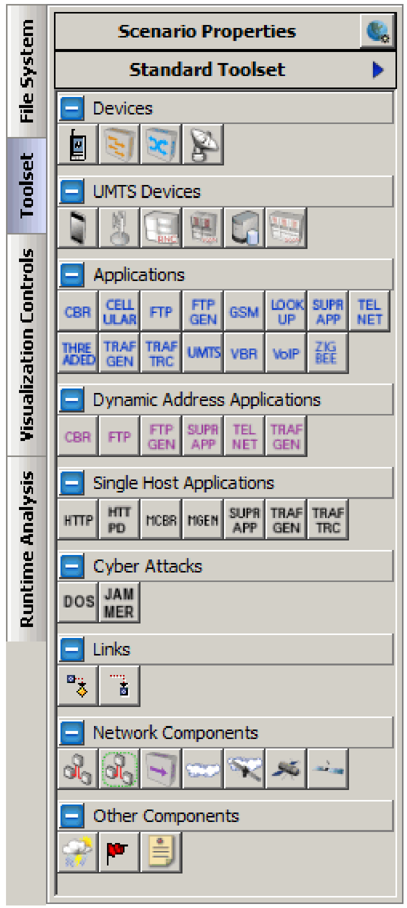


FIGURE 3-20. Toolset Panel

At the top of the Toolset panel is the **Scenario Properties** bar. Clicking on the button in this bar launches the Scenario Properties Editor which is used to configure scenario level properties. See Section 3.3.2 for details. Below the **Scenario Properties** bar is the toolset header which displays the name of the selected toolset. Clicking on the right arrow button displays a list of available toolsets (see Figure 3-21) Note that Enterprise Toolset is a sample of a user-created toolset and is not displayed by default.

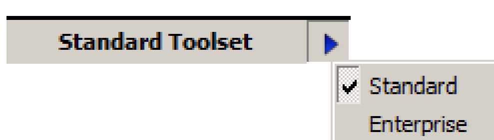


FIGURE 3-21. Selecting a Toolset

EXata GUI comes with the Standard Toolset which consists of the following toolbars: Devices, UMTS Devices, Applications, Dynamic Address Application, Single Host Applications, Cyber Attacks, Links, Network Components, and Other Components. Users can customize the Standard toolset and create new toolsets. See Section 3.5.3 for details of toolset customization.

Note: The list of toolsets, the toolbars in each toolset, and the components that appear in each toolbar can vary depending on your installed options.

The Toolset panel, like all other panels in Architect, can be resized. If the height of the panel is not sufficient to display the entire toolset, a vertical scroll bar is displayed.

Note: The toolbars can be expanded or collapsed by clicking the + and − buttons.

The toolbars in the Standard Toolset are described below.

Devices

The **Devices** toolbar contains buttons for basic devices, which include a standard node (Default device), Switch, ATM device, and Satellite Ground Station.

UMTS Devices

The **UMTS Devices** toolbar contains buttons for UMTS devices, which include a UMTS-UE, UMTS-NodeB, UMTS-RNC, UMTS-SGSN, UMTS-HLR, and a UMTS-GGSN.

Note: The **UMTS Devices** toolbar is visible only if UMTS Model Library is enabled by your license.

Applications

The **Applications** toolbar contains buttons for server-client applications that are configured using the node ID or IP address of the server or destination node.

Dynamic Address Applications

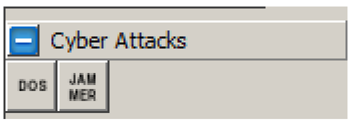
The **Dynamic Address Applications** toolbar contains buttons for server-client applications that are configured using the Fully Qualified Domain Name (FQDN) of the server or destination node.

Single Host Applications

The **Single Host Applications** toolbar contains single host applications. These include multicast applications, where traffic is sent to a multicast or broadcast address, and multi-server applications, such as HTTP.

Cyber Attacks

The **Cyber Attacks** toolbar contains attack models that are available in Architect. These include Denial-of-Service (DoS) and Jammer attack models.



Note: The **Cyber Attacks** toolbar is visible only if Cyber Model Library is enabled by your license.

Links

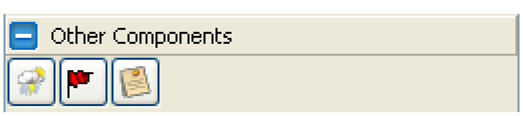
The **Links** toolbar contains buttons for two types of links: Link and BGP Link. Link can be used to create wired or wireless links between nodes or between a node and a hub, wireless network, or satellite. The BGP Link is a conceptual link that defines the speakers only in a BGP protocol.

Network Components

The **Network Components** toolbar contains buttons for Hierarchy, Constrained Hierarchy, Hub, Wireless Network, and Satellite components.

Other Components

The **Other Components** toolbar contains buttons for the Weather Effects component, Mobility Waypoint marker, and Note components.



3.1.3.3 Visualization Controls

The **Visualization Controls** panel provides the simulation status (time, speed, and progress) and displays the event and layer filters that can be applied to the animation.

Note: The Visualization Controls panel is enabled only in Visualize mode. See Section 6.5.3 for details.

3.1.3.4 Runtime Analysis

The **Runtime Analysis** panel displays properties of nodes, queues, and subnets. From this panel, animation filters can be applied independently at the node level. Dynamic parameters can also be set from here.

Note: The **Runtime Analysis** panel is enabled only in Visualize mode. See Section 6.5.4 for details.

3.1.4 Canvas

The canvas is the main work area of the Architect which allows you to build scenarios using components such as devices, applications, links, and subnets from the toolset.

The canvas has one or more tabs, one for each open scenario. If a scenario has unsaved changes, an asterisk (“*”) appears after the scenario’s name in the tab. If the scenario has been saved, positioning the mouse over the tab displays the location of the scenario’s configuration file.

The grid on the canvas and the horizontal and vertical rulers are displayed to help position scenario components. Red markers in the rulers indicate the boundaries of the work area.

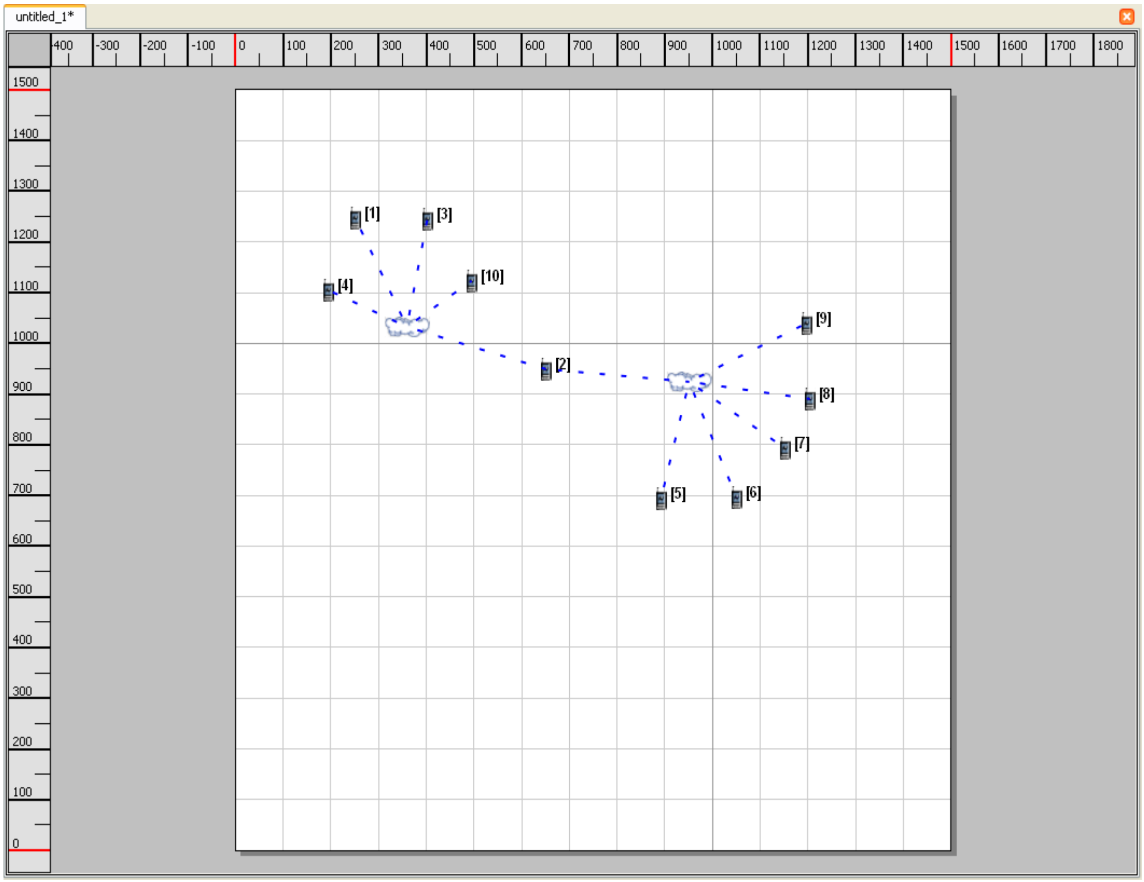


FIGURE 3-22. Architect Canvas

Placing Objects

To place one or more instances of a unit (device, switch, subnet, satellite, hierarchy, etc.) on the canvas, first select the object by clicking on the corresponding unit's button in the **Devices** or **Network Components** toolbar in the toolset. Then click at the desired positions on the canvas.

To exit the insert mode and switch to select mode, select a different component in the toolset or click the **Select**, **Lasso**, or **Region Zoom** buttons on the toolbar or press the Escape key.

Links

To draw a link between two objects, select the **Link**  button from the toolset, left-click on one object on the canvas, drag the mouse to the other object and release it.

A point-to-point link is created by connecting two units. A point-to-point link between two terrestrial units appears as a solid blue line. A point-to-point link between a terrestrial unit and a satellite appears as a solid purple line.

A device is made part of a wireless subnet by drawing a link between the device and the cloud icon representing the subnet. The link between a device and a wireless subnet is displayed as a dashed line.

A device is made part of a wired subnet by drawing a link between the device and the hub icon representing the subnet. The link between a device and a wired subnet is displayed as a solid blue line.

A point-to-point link can also be created between a non-ATM device and an ATM end system. By default, the type of an ATM device is a switch. To make the ATM device an end system, open the ATM Device properties editor (see Section 3.3.10) and change **Node Type** to *ATM End System*.

To create an ATM link between two ATM devices (switch or end system), draw a link between the two ATM devices. Links between two ATM end systems are not allowed.

To create BGP links, refer to *Multimedia and Enterprise Model Library* for details.

Applications

Client-server applications are created by drawing application links between two nodes. Application links appear as solid green lines.

A single host application is created by selecting the application from the **Single Host Applications** toolbar and clicking on the node.

Selection

To select an object, left-click on it. To select a group of objects, hold the control key down while clicking on the objects. A group of objects can also be selected by using the **Select** or **Lasso** buttons, as described in Section 3.4.1. A selected object is highlighted, whereas a selected link turns red. If a hierarchy is open, pressing Ctrl+A selects all components in the active hierarchy.

To deselect a selected object or group of objects, select another object or left-click on the canvas. You can also deselect a single component by clicking on it while holding down the control key.

Display Settings

The display of rulers, grid, labels (node IDs, host-names, etc.), wireless subnet (cloud) icon, etc. can be controlled by setting the options in the Display Settings window (see Section 3.1.1.3).

The type of view (X-Y, X-Z, Y-Z, 3D, or Split Screen) can be selected from the select view pull-down list in the **View** toolbar.

Context-sensitive Right-click Menu

Selecting an object or group of objects on the canvas and right-clicking displays a menu with commands that depend on the selected object(s):

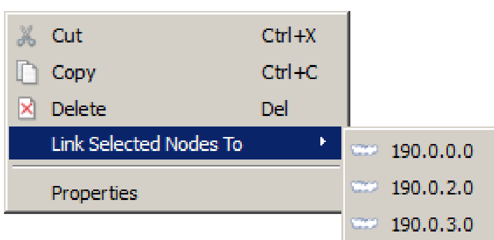


FIGURE 3-23. Context-sensitive Right Click Menu

- **Cut, Copy, Delete**: These editing commands are available for all objects, when applicable, and work in the same way as the editing commands from the **Edit** menu.
- **Link Selected Nodes To**: This command is enabled if one or more connectable devices are selected and one or more wired or wireless subnets have been placed on the canvas. Selecting this command displays the IP addresses of all wired and wireless subnets in the scenario (except the default wireless subnet). Selecting the IP address of a subnet from the list will draw links from the selected nodes to the selected subnet.

  This command is disabled if the selected objects are not all of the same type, or if a non-connectable device are also selected.
- **Properties**: This command opens the properties editor of the type of the selected object(s) (see Section 3.3. If different types of objects are selected, then this command is disabled.

Notes

Descriptive comments can be added to a scenario by means of notes on the canvas. To add a note, click the **Note** button on the **Other Components** tool bar in the toolset, and place a **Note** icon at the desired location on the canvas. Click the **Select** button on the **View** toolbar and then double-click on the **Note** icon. This opens a Note Editor using which any text can be entered. The leading part of the text is displayed as the label of the note.

3.1.4.1 Position Indicators

The position indicators are located below the canvas. If a single node is selected, its position (X-, Y-, and Z- coordinates or latitude, longitude, and elevation) is displayed in the indicators. If no node is selected or multiple nodes are selected, then the position of the cursor is displayed.

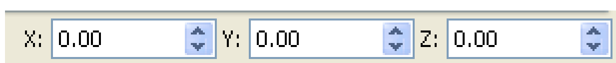


FIGURE 3-24. Position Indicators

The position of a node can be changed by selecting the node and editing its coordinates in the position indicators. A coordinate can be edited by pressing the up- or down-arrow or by entering the new coordinate in the field.

> **Note:** The value entered here is checked against canvas dimensions. Entering a value outside of these dimensions will not move the device.

3.1.5 Bottom Panels

The following panels are available below the canvas, and are used to display scenario properties, capture and log output and error information, watch the state of dynamic variables, and configure batch experiments:

- Table View
- Output Window
- Error Log
- Watch Variables
- Batch Experiments

> **Note:** These panels occupy the same space and only one of them can be opened at any time. A panel can be opened or closed by clicking on its tab. By default all panels are closed.

3.1.5.1 Table View Panel

The **Table View** panel displays the properties of scenario components in a tabular form. The Table View has six tabs: **Nodes**, **Groups**, **Interfaces**, **Networks**, **Applications**, and **Hierarchies**.

Each tab displays a table with one row for each component of the type of the table. Some of the properties of the component are displayed in the table. Properties can not be edited in the **Table View** panel directly and can only be changed in the component's properties editor (see Section 3.3). To open the properties editor of a component, double-click on the row for the component or right-click on the row and select Properties.

Multiple rows in a table can be selected by pressing the control key and clicking on the desired rows. To select a range of adjacent rows, click on the first row in the range, press the shift key, and click on the last row in the range. You can also select a range of adjacent rows by clicking on the first row in the range and dragging the mouse over the rows to be selected. To select all rows, right-click on the table and choose **Select All** from the menu, or use the Ctrl+ A or Ctrl+L shortcut. The group properties editor for multiple components of the same type can be launched by selecting the rows for the components, right-clicking and selecting **Properties**.

To sort a table by a column, click the column heading. To sort the entries in reverse order, click on the column heading again. To delete one or more component(s), select the row(s) and press **Delete** or right-click and select **Delete**.

3.1.5.1.1 Nodes Tab

The **Nodes** tab displays the properties of devices, such as default devices, switches, ATM devices, and satellites.

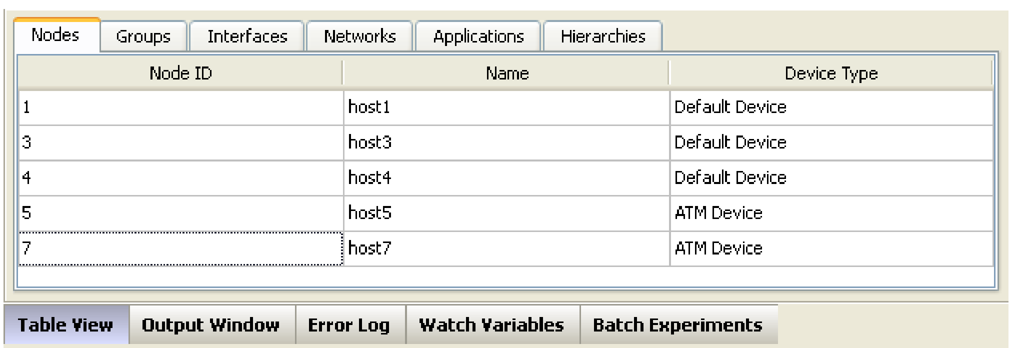
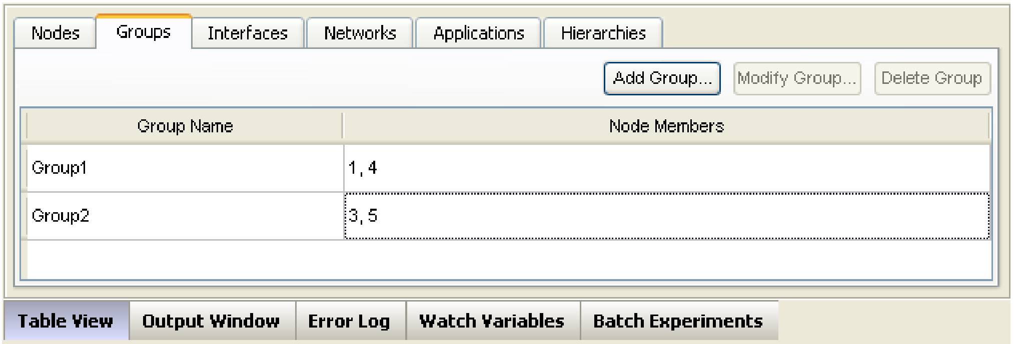


FIGURE 3-25. Nodes Tab in Table View Panel

3.1.5.1.2 Groups Tab

The **Groups** tab displays all logical groups of nodes in the scenario. See Section 3.4.4 for details of creating node groups.

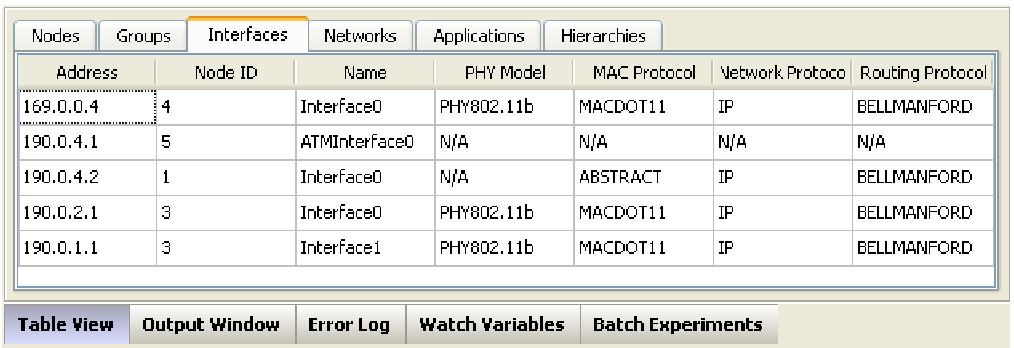


Group Name	Node Members
Group1	1, 4
Group2	3, 5

FIGURE 3-26. Groups Tab in Table View Panel

3.1.5.1.3 Interfaces Tab

The **Interfaces** tab displays all wired, wireless, and ATM interfaces in the scenario.



Address	Node ID	Name	PHY Model	MAC Protocol	Network Protoco	Routing Protocol
169.0.0.4	4	Interface0	PHY802.11b	MACDOT11	IP	BELLMANFORD
190.0.4.1	5	ATMInterface0	N/A	N/A	N/A	N/A
190.0.4.2	1	Interface0	N/A	ABSTRACT	IP	BELLMANFORD
190.0.2.1	3	Interface0	PHY802.11b	MACDOT11	IP	BELLMANFORD
190.0.1.1	3	Interface1	PHY802.11b	MACDOT11	IP	BELLMANFORD

FIGURE 3-27. Interfaces Tab in Table View Panel

3.1.5.1.4 Networks Tab

The **Networks** tab displays the wired and wireless subnets and point-to-point links in the scenario.

Note that each scenario has a default wireless subnet. Every default device that is not connected to any other subnet or link is part of the default wireless subnet.

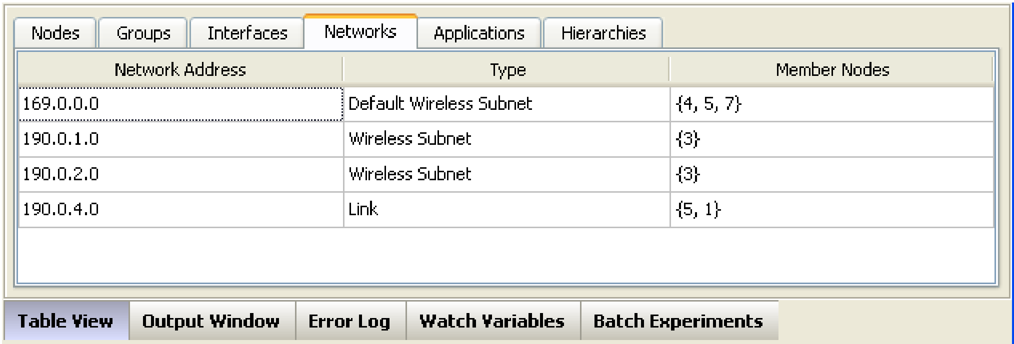


FIGURE 3-28. Networks Tab in Table View Panel

3.1.5.1.5 Applications Tab

The **Applications** tab displays all client-server, single-host, and loopback applications in the scenario.

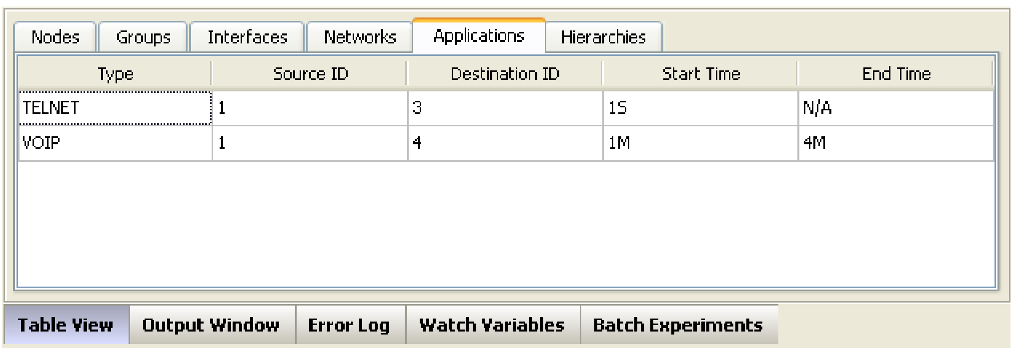


FIGURE 3-29. Applications Tab in Table View Panel

3.1.5.1.6 Hierarchies Tab

The **Hierarchies** tab displays all hierarchies in the scenario.

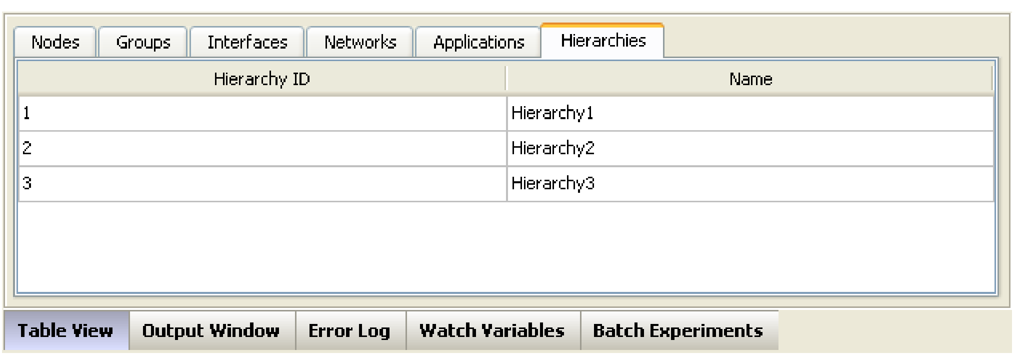


FIGURE 3-30. Hierarchies Tab in Table View Panel

3.1.5.2 Output Window Panel

Any message that the simulator prints to standard output is redirected to the output window.

Note: The **Output Window** panel is enabled only in Visualize mode. See Section 6.6.2 for details.

3.1.5.3 Error Log Panel

The **Error Log** panel displays any error messages that are generated when an application is launched, scenario is opened, or the user tries to load an unsupported file format.

Figure 3-31 shows an example error message in the **Error Log** panel.

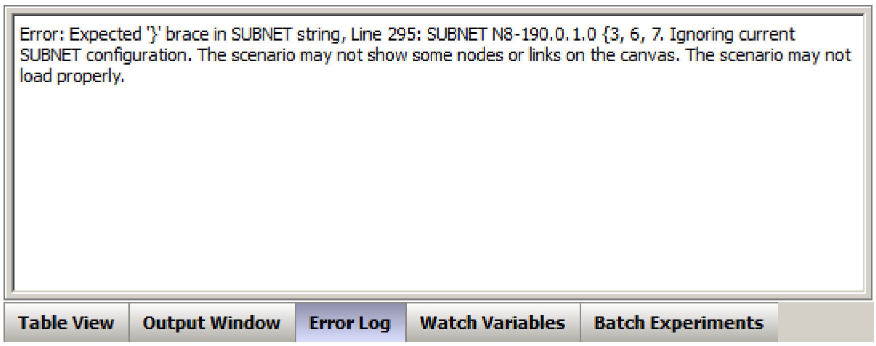


FIGURE 3-31. Error Log Panel

3.1.5.4 Watch Variables Panel

From the **Watch Variables** panel you can watch the values of dynamic variables during a simulation. The watch variables table displays the path to the parameter in the dynamic hierarchy, the name of the parameter, and its current value.

Note: The **Watch Variables** panel is enabled only in Visualize mode. See Section 6.6.4 for details.

3.1.5.5 Batch Experiments Panel

The **Batch Experiments** panel is used to configure batch experiments and allows users to run the same scenario with the same configuration parameters under different set of values. See Section 3.4.9 for details.

3.2 Creating Scenarios

In EXata, a scenario refers to a combination of network topology, properties of network components, characteristics of network traffic, and conditions under which the network operates. Creating a scenario entails specifying details such as:

- Network topology
- Application traffic
- Network properties and simulation parameters
- Run-time settings

This section gives a brief overview of how to create a scenario in Architect.

3.2.1 Scenario Topology

The topology of a network is defined by the number and location of network devices and the physical and logical connections between them.

In Architect, the **Devices** toolbar provides models for commonly used devices (Default device, Switch, and ATM Device). The Default device is highly configurable and can be used to model a variety of communication and network devices.

A wireless subnet is modeled by placing a wireless subnet (cloud) icon on the canvas and connecting it to the nodes that belong to the subnet. The link between a wireless subnet icon and a node is a logical connection, not a physical link.

A wired subnet is modeled by placing a wired subnet (hub) icon on the canvas and connecting it to the nodes that belong to the subnet.

Point-to-point connections between devices are modeled by direct links between them.

Note: Every scenario has a default wireless subnet. Any node placed on the canvas that is not connected to any other device or subnet belongs to the default wireless subnet. The default wireless subnet is not represented by any icon on the canvas.

3.2.1.1 Placing Objects

To place objects (devices or subnets) on the canvas do the following:

1. From the **Toolset** panel, select a device from the **Devices** toolbar or a subnet (wired subnet, wireless subnet, or subnet) from the **Network Components** toolbar. Click on the canvas to place the object.
2. Continue clicking on the canvas to place multiple objects.
3. Once you are finished placing objects, you can exit from insert mode by clicking the **Select**  button, pressing the **Escape** key, or pressing the "**S**" key.

You can also place devices by using the **Node Creation Wizard**, as described in Section 4.2.3.2.2.

3.2.1.2 Creating Links

To create links between objects do the following:

1. Place objects on the canvas as described above in Section 3.2.1.1.
2. Click the **Link** button, then click on the desired object (holding down the left mouse button) and drag the link to the other object.
   - Point-to-point links between default devices, ATM devices, and switches appear as solid blue lines.
   - The connection between a device and a wired subnet appears as a solid blue line.
   - The connection between a device and a wireless subnet appears as a dashed blue line.
   - The connection between a device and a satellite appears as a solid purple line.

Figure 3-32 shows different types of objects and links in a scenario.

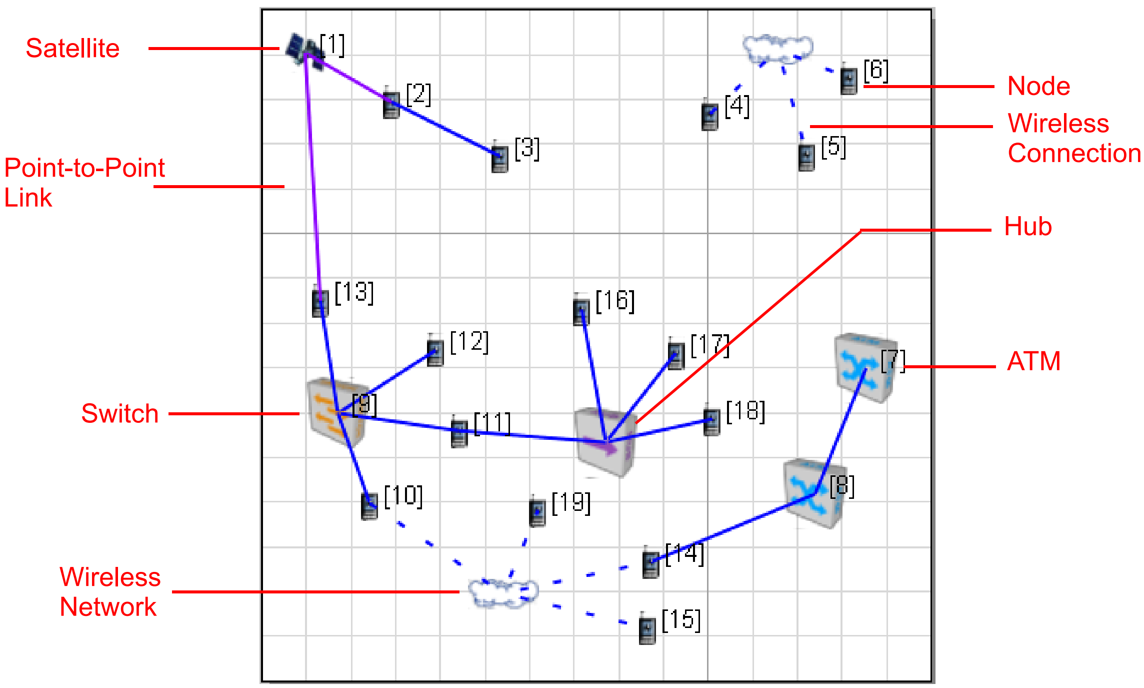


FIGURE 3-32. Objects and Links in a Scenario

To connect several nodes to a subnet at the same time, you can do one of the following:

- To connect multiple nodes to a subnet already on the canvas, select the nodes, right click on one of the selected nodes, select **Link Selected Nodes To**, and select the IP address of the subnet to connect the nodes to.
- To connect multiple nodes to a new subnet, select the nodes. Then select the subnet from the **Network Components** toolbar and place it on the canvas. All selected nodes will be connected to the newly placed subnet.

3.2.2 Configuring Applications

There are four types of applications in EXata. This section describes how to configure each type of application.

Client-Server Applications

To set up a client-server application using the node ID or IP address of the server or destination node, do the following:

1. Click on the desired application button in the **Applications** toolbar.
2. Click on the source node and drag the mouse to the destination node. A solid green line is drawn from the source to the destination.

Figure 3-33 shows a FTP session set up from node 1 to node 3.

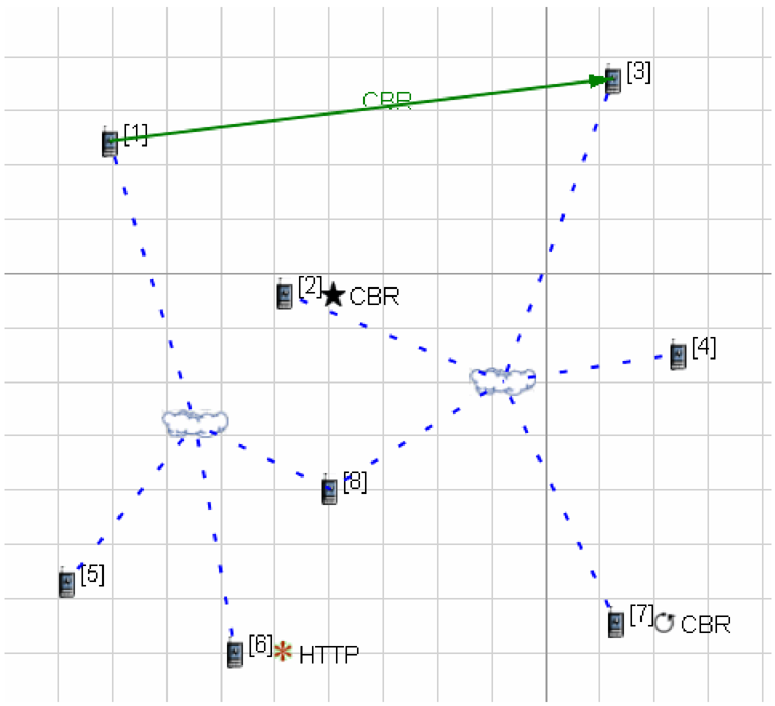


FIGURE 3-33. Types of Applications in a Scenario

Dynamic Address Applications

Dynamic address applications can be used to model client-server applications in which the destination or server node is identified by its Fully Qualified Domain Name (FQDN). To configure a dynamic address application, do the following:

1. Click on the desired application button in the **Dynamic Address Applications** toolbar.
2. Click on the source node. A ★ symbol appears next to the host node.

Figure 3-33 shows dynamic address application (CBR) configured at node 2.

Single Host Applications

Single host applications can be used to model multicast applications in which traffic is sent from a source node to a group of destination nodes. To configure a single host application, do the following:

1. Click on the desired application button in the **Single Host Applications** toolbar.
2. Click on the host node. A  symbol appears next to the host node.

Figure 3-33 shows a single host application (HTTP) configured at node 6.

Loopback Applications

A loopback application is a special configuration of client-server application in which the source and destination are the same. To set up a loopback application, do the following:

1. Click on the desired application button in the **Applications** toolbar.
2. Double-click on the node where you want to configure the application. A  symbol appears next to the node.

Figure 3-33 shows a loopback application (CBR) configured at node 7.

The parameters of an application can be set in the Properties Editor of the application. See Section 3.3.13 for details.

3.2.3 Configuring Parameters

In a scenario, several parameters have to be configured. These include:

- Simulation parameters, such as length of simulation, terrain dimensions, coordinate system, seed for random number generation, etc.
- Network environment parameters, such as terrain properties, channel frequencies and propagation effects including path loss, fading, and shadowing
- Properties of network devices such as nodes, switches, hubs, and routers
- Protocols running at each node and protocol-specific parameters
- Parameters to collect statistics and packet traces

Most of these are optional; you can start with a basic network scenario and specify as much detail as necessary to improve the accuracy of your network model.

Parameters are typically specified in various Properties Editors. A Properties Editor is associated with each type of network component. Properties Editors are described in Section 3.3. Chapter 4 describes in detail how to configure these parameters. Protocol-specific parameters for EXata's pre-configured models are described in the Model Library documents.

3.2.4 Running the Scenario

After configuring a scenario, you need to save it and run it. Saving scenarios is described in Section 3.2.5.

You can start a live simulation of the scenario by clicking the **Run Simulation** button on the **Run** toolbar (see Section 3.1.2.3) or by selecting the **Run Simulation** command from the **Tools** menu (see Section 3.1.1.4). This switches Architect from Design mode to Visualize mode. In Visualize mode, you can watch the animation of the scenario, observe run time statistics, and interact with the simulator in a number of ways. Visualization features of Architect are described in Chapter 6. After running the scenario in

Visualize mode, you can analyze the simulation results in Analyzer. Analyzer features are described in Chapter 7.

After visualizing a scenario, you can return to Design mode to make changes in the scenario by clicking the  button on the Visualization toolbar (see Section 3.1.2.4).

You can also choose to record an animation trace of the scenario. To do so, click on the **Record Animation** button on the **Run** toolbar or by selecting the **Record Animation** command from the **Tools** menu. This opens a dialog box in which you can select to record the trace interactively or non-interactively. In either case, Architect switches to Visualize mode.

You can also specify advanced options for running the scenario. To do so, click on the **Run Settings** button on the **Run** toolbar or by selecting the **Run Settings** command from the **Tools** menu. This opens the **Run Settings** dialog box. The options available in the **Run Settings** dialog box are described in Appendix C.

3.2.5 Saving and Opening Scenarios

After configuring a scenario, you can save it by clicking the **Save** button on the **Standard** toolbar (see Section 3.1.2.1) or by selecting the **Save**, **Save As**, or **Save As Portable** command from the **File** menu (see Section 3.1.1.1). In the window that opens, you can select a name for the scenario and the location to save it.

- A folder is created with the name you specify.
- The folder contains all files created for the scenario, including the scenario configuration file. If the **Save As Portable** command is used to save the scenario, then any external files referenced by the scenario are also saved in this folder. These files are named after the name of the scenario.

For example, if you specify the name of the new scenario to be “wireless-scenario” and specify the location to be C:/scalable/exata/5.1/scenarios/user, then a folder C:/scalable/exata/5.1/scenarios/user/wireless-scenario is created. This folder contains the scenario configuration file, “wireless-scenario.config”, the node placement file, “wireless-scenario.nodes”, the application configuration file, “wireless-scenario.app”, and any other files for that scenario created by Architect.

> **Note:** It is advised to always use **Save As Portable** whenever you are planning to archive a scenario for later use or when you are sending the scenario to another user, since the file paths of a future user may be different from the creator of any given scenario.

You can open a saved scenario by clicking the **Open** button on the **Standard** toolbar or by selecting the **Open** command from the **File** menu. Browse to the folder that contains the files for the scenario you want to open and select the scenario configuration file (which normally has the extension “.config”).

You can also open a scenario from the **File System** panel. To do so, browse to the scenario configuration file and double-click on it or right-click on it and select **Open** from the menu.

3.3 Properties Editors

A properties editor is a dialog box for setting properties of a component, such as a device, link, application, interface, etc. The identity of the component (ID, address, etc.) is displayed in the properties editor title bar. This section gives a general description of all properties editors and describes the more commonly used properties editors. The properties editors for the applications are described in the model libraries.

3.3.1 General Description of Properties Editors

A properties editor has one or more tabs. Each tab has one or two panels. If a tab has a single panel, then a parameter table is displayed in the panel. If the tab has two panels, then a list of parameter groups appears in the left panel. Elements in the list may have subgroups.

3.3.1.1 Parameter Table

The left column (Property) of the parameter table displays the parameter names and the right column (Value) is for entering the parameter values (see Figure 3-34). A parameter value can be entered in one of the following ways:

- *Typing* a value in the field.
- *Selecting* a value from a drop-down list.
- *Checking* or clearing a check-box.
- *Clicking* the select file button, which opens a file selection window using which a file can be selected.
- *Launching* an Array Editor to configure parameters with multiple instances (see Section 3.3.1.4).
- *Launching* an editor using which different components of the parameter value can be specified. The editor is launched by clicking a button in the parameter value field.

For example, in Figure 3-34 **Routing Protocol** is the name of the parameter which is set to value *Bellman Ford*. In this case, the parameter value is selected from a drop-down list.

> **Note:** Specifying the value of some parameters may require a combination of the methods listed above. For example, entering a time value requires you to type in a numeric value in a field and selecting the time unit from a drop-down list.

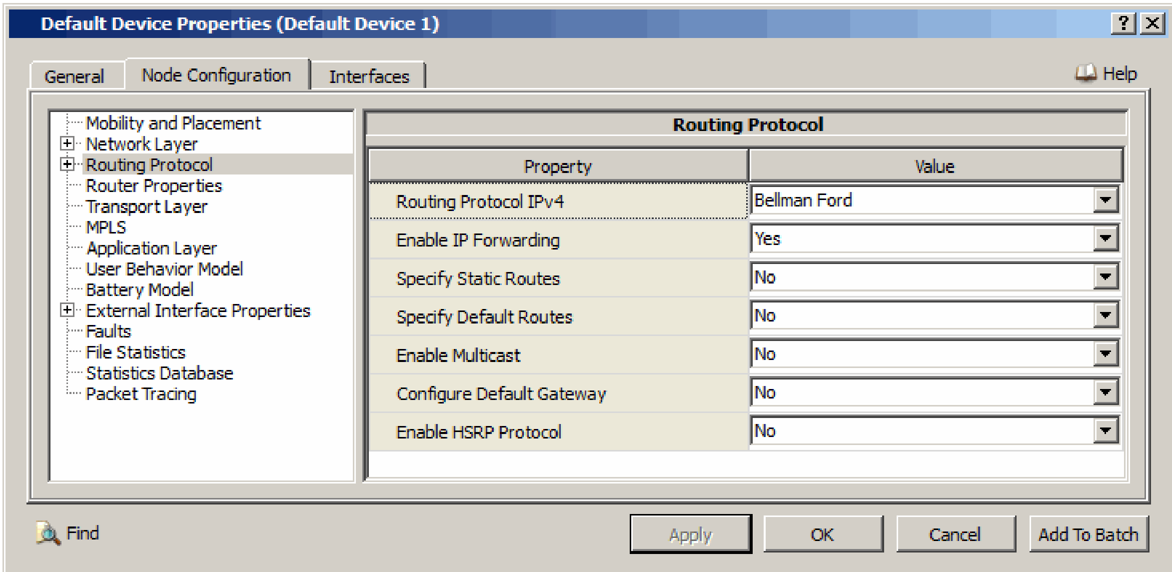


FIGURE 3-34. Parameter Property Names and Values

For most parameters a default value is displayed in the table. If the default value is changed, a symbol appears after the value field indicating that the value is set to something other than the default value. Clicking on the symbol restores the default value of the parameter (see Figure 3-35).

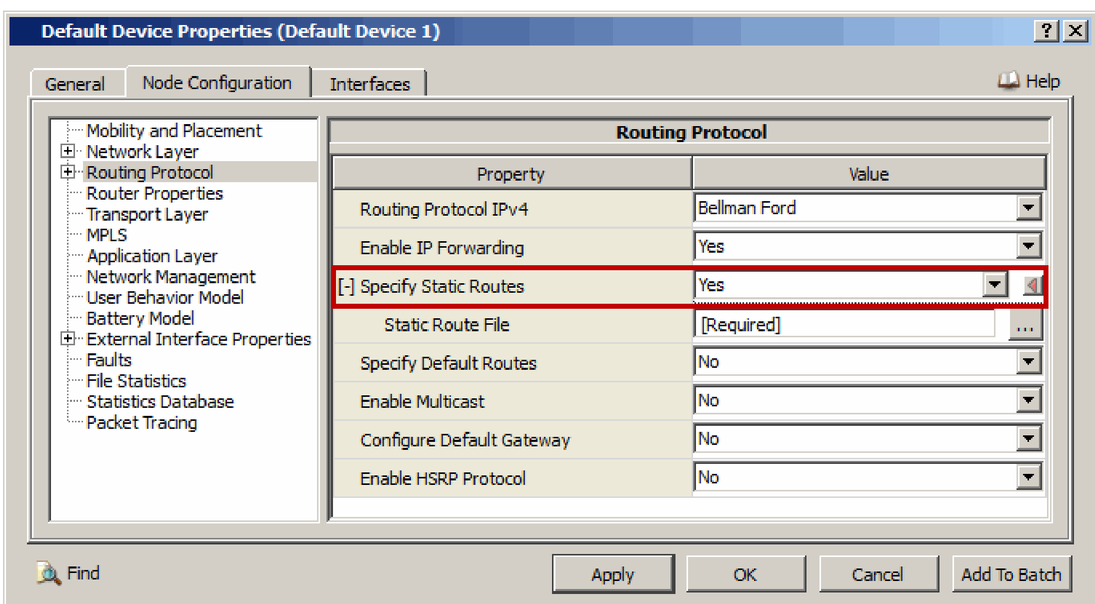


FIGURE 3-35. Parameter Set to Non-default Value

Some parameters (*dependent parameters*) depend on values of other parameters (*controlling parameters*). If you a select a value for the controlling parameter from a drop-down list, the dependent parameters are displayed with an indentation.

Note: After selecting a value of a controlling parameter from the pull-down list, click on the **Apply** button. This will display the dependent parameters for the selected value of the controlling parameter. The dependent parameters may appear in the same tab as the controlling parameter or in other tabs.

Note that in some cases, the dependent parameters are not displayed until you click on the **Apply** button after changing the value of the controlling parameter.

Some properties are not configurable independently. The value of a parameter may be set automatically when another parameter is set. In this case, the parameter value is displayed but is not editable. This is indicated by graying out the value field.

3.3.1.2 Find Function

Users can search for any string using the collapsible find panel that appears at the bottom of the properties editor. The find function can be used to search for strings anywhere in the properties editor: property group names in the left panel, parameter names, and selected or entered values in the table. The find function searches for the specified string in all tabs of the properties editor, starting with the first tab.

3.3.1.3 Applying Changes

1. Click the **Apply** button to apply any changes made to parameter values.
2. Click the **OK** button to apply the changes and close the properties editor.
3. Click the **Cancel** button to close the properties editor without accepting the changes. (Changes that have already been applied are *not* undone if the **Cancel** button is clicked.)

3.3.1.4 Configuring Parameters with Multiple Instances

Array parameters, i.e., parameters with multiple instances, are configured in a separate dialog box called the *Array Editor*. The Array Editor provides convenient means for duplicating array instances and for setting multiple instances of a parameter to the same value.

The parameter specifying the number of instances is configured in the properties editor, Clicking on **Open Array Editor** ... button in the Value column of the number of instances parameter opens the Array Editor. All parameters that are dependent on the number of instances parameter are listed in the Array Editor. These dependent parameters are configured in the same way as the parameters in a properties editor.

Opening the Array Editor

The Array Editor is launched by clicking on the on the **Open Array Editor** ... button in the Value column of the number of instances parameter.

For example, open the Scenario Properties Editor, as described in Section 3.3.2 and go to the Channel Properties tab. Set **Number of Channels** to *2.*

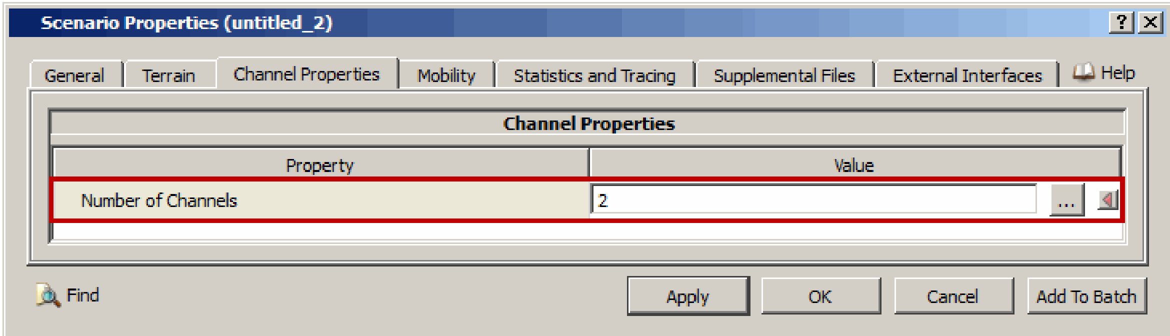


FIGURE 3-36. Opening the Array Editor

Click on the on **Open Array Editor** ... button in the Value column. This opens the Array Editor.

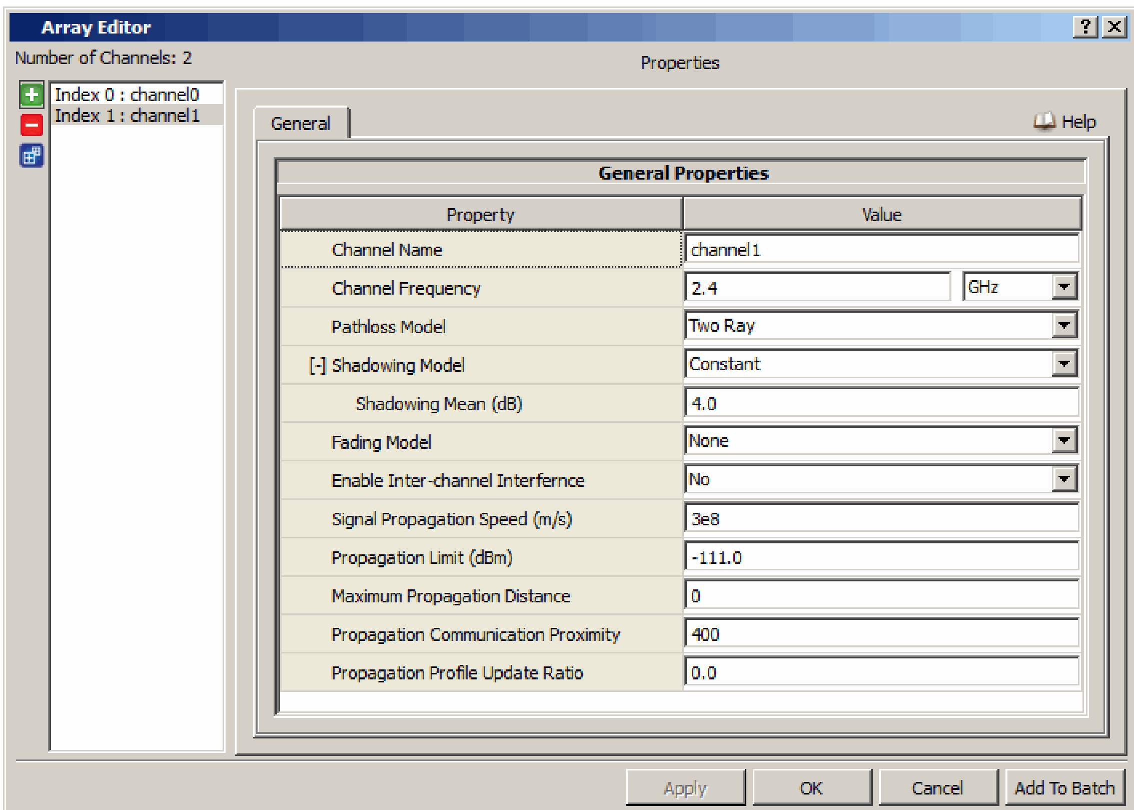


FIGURE 3-37. Array Editor Displaying Channel Properties

The left panel of the Array Editor lists the index (instance) numbers and the right panel displays the channel properties and values.

Setting Properties for a Single Instance

To set properties for a single instance, select the index in the left column and set the properties in the right panel. Properties in the right panel can be set as in any properties editor.

Setting Properties for Multiple Instances

To set properties of multiple instances to the same value, select the instances in the left column by clicking on the index numbers while pressing the control key. Parameter values set in the right panel will apply to all the selected instances.

If two or more of the selected instances have different values set for a parameter, that parameter name appears in red and the Value field is blank, indicating a conflict in the values. If a value is specified for this parameter, it will apply to all selected instances; otherwise, the previously set values are not modified.

Adding and Deleting Instances

To create another instance with default values of the parameters, click on the **Add Instance** button on the left. The newly created instance has default values for all parameters in the right column.

To delete an instance, select the last instance and click on the **Delete Instance** button on the left.

> **Note:** Only the last instance can be deleted. If no instance is sliced or an instance other than the last instance is selected, the **Delete Instance** button is disabled.

Copying an Instance

To create another instance which has the same values of the parameters as another instance which has already been configured, select the instance that has been configured and click on the **Copy Instance** button on the left.

3.3.1.5 Adding Parameters to To Batch Experiments

To add a parameter to a batch experiment, select the parameter from the parameter table and click the **Add to Batch** button. Set the parameter value(s) for the batch experiment in the **Batch Experiment** panel. See Section 3.4.9 for details.

3.3.1.6 Editing Properties for Groups of Objects

To assign the same value for a parameter for multiple objects of the same type, select the objects (on the canvas or in the **Table View** panel), right click and select **Properties**. This launches the group properties editor of the type of selected objects. The group properties editor can also be launched from the **Groups** tab of the **Table View** panel (see Section 3.4.4). Properties that are unique to each object (such as Node Name in the General tab of the Default Device properties editor) appear in red and the corresponding Value field is grayed out indicating that a value can not be entered in the field. For other properties, if the values for different objects are different, then the property name appears in red. However, a new value can be entered for the property and that value will apply to all objects in the group.

3.3.2 Scenario Properties Editor

The **Scenario Properties** editor is used for setting scenario-level properties. It can be opened by clicking on the **Scenario Properties** button in the toolset or by selecting **Tools > Scenario Properties** from the menu bar.

The **Scenario Properties** editor is shown in Figure 3-38 and its tabs are described below.

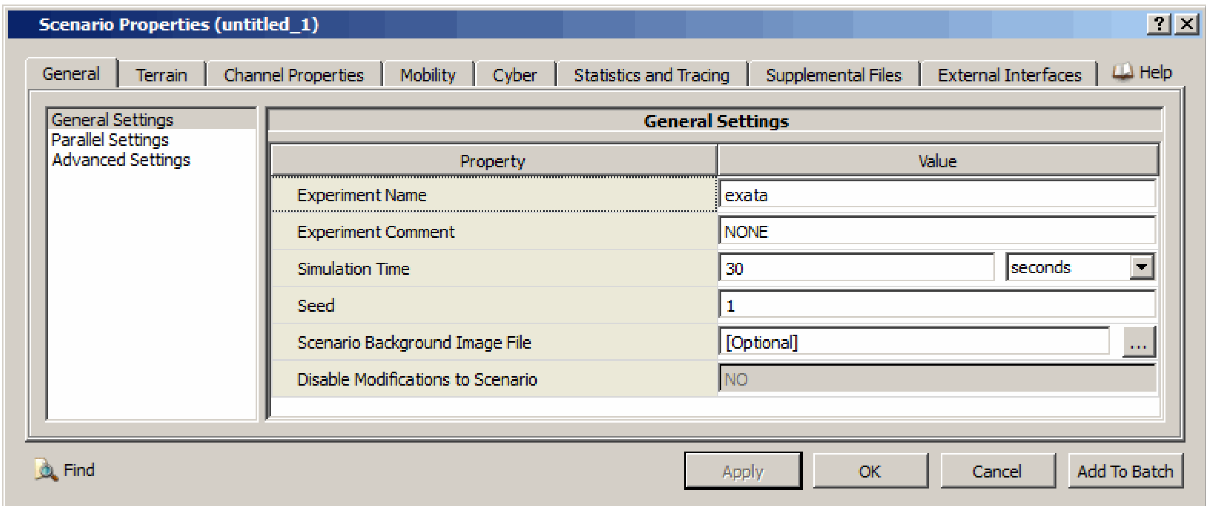


FIGURE 3-38. Scenario Properties Editor

Tab	Description
General	This tab is for setting scenario wide parameters: • General Settings: General parameters such as simulation time, seed, background image, etc. • Parallel Settings: Parameters for parallel simulation. • Advanced Settings: Configuration parameters for enabling Dynamic Commands and configuring the event queue type.
Terrain	This tab is for specifying the coordinate system, terrain dimensions, and terrain data.
Channel Properties	This tab is for configuring channel properties such as frequency and pathloss, fading, and shadowing models.
Mobility	This tab is used to define the mobility strategy for nodes in the scenario.
Cyber	This tab is used to configure some of the Cyber attack models. **Note:** This tab is visible only if the Cyber Model Library is enabled by your license.
Statistics and Tracing	This tab is for configuring parameters to collect statistics and packet traces at the scenario level: • File Statistics: Parameters for enabling statistics for different layers and models at the scenario level. • Packet Tracing: Scenario-level parameters for packet traces. • Statistics Database: Parameters for configuring the statistics database.

Tab	Description
Supplemental Files	This tab is used to define the supplemental files used by models in the scenario.
External Interfaces	This tab is for configuring external interface parameters: • AGI Interface: AGI Satellite Tool Kit (STK) interface parameters. • Socket Interface: Socket interface parameters. • VR-Link Interface: VR-Link interface parameters (for configuring HLA and DIS). • DIS Interface: Legacy Distributed Interactive Simulation (DIS) interface parameters. • HLA Interface: Legacy High-Level Architecture (HLA) interface parameters. • Warm-up Phase: Parameters to configure the warm-up phase.

3.3.3 Default Device Properties Editor

The properties editor for a default device can be opened by one of the following ways:

1. From the canvas, select the device on the canvas, right-click and choose **Properties**, or double-click the device.
2. From the **Nodes** tab in the **Table View** panel, do one of the following:
   a. Double-click the row for the device.
   b. Right-click the row for the device and choose **Properties**.

The default device properties editor is shown in Figure 3-39 and its tabs are described below.

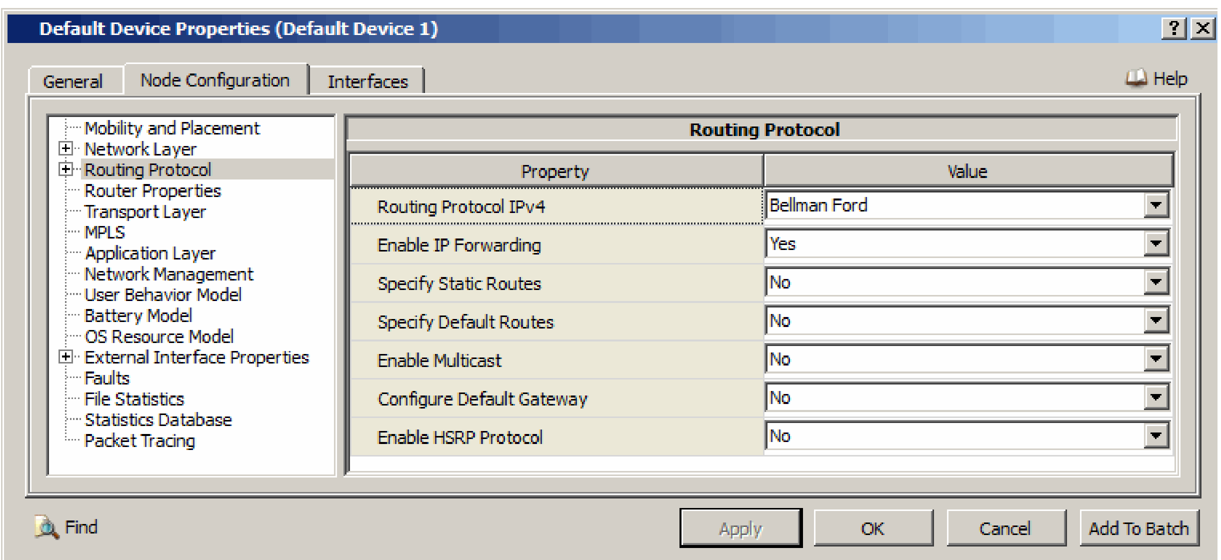


FIGURE 3-39. Default Device Properties Editor

General Tab

This tab is for setting the node name, icons, and partition number.

Node Configuration Tab

The property groups in this tab are described below.

Property Group	Description
Mobility and Placement	Mobility model and related parameters.
Network Layer	Network protocol, queues, schedulers, and other Network Layer parameters at the node level.
Routing Protocol	Routing-related parameters, including unicast and multicast routing protocols at the node level.
Router Properties	Router model properties at the node level.
Transport Layer	Transport protocol (UDP, TCP, or RSVP) configuration parameters at the node level.
MPLS	MPLS configuration parameters at the node level.
Application Layer	Application layer parameters.
Network Management	Parameters for configuring the SNMP model
User Behavior Model	User profile and traffic pattern parameters.
Battery Model	Battery model parameters.
OS Resource Model	OS Resource model parameters. **Note:** This parameter group is visible only if the Cyber Model Library is enabled by your license.
External Interface Properties	This tab is for configuring external interface parameters at the node level: • AGI Interface: AGI Satellite Tool Kit (STK) interface parameters.
Faults	Parameters for configuring faults at the node level.
File Statistics	Parameters to collect statistics at the node level.
Statistics Database	Parameters for specifying the number of meta data columns of statistics database tables and for configuring model-specific tables.
Packet Tracing	Parameters to enable packet traces at the node level.

Interfaces Tab

The property groups in this tab are described below.

Property Group	Description
Physical Layer	Radio and antenna models and other Physical Layer parameters at the interface level.
MAC Layer	MAC protocol and other MAC layer parameters.
Network Layer	Network protocol, queues, schedulers, and other Network Layer parameters at the interface level.
Routing Protocol	Routing-related parameters, including unicast and multicast routing protocols at the interface level.
Faults	Interface faults.
File Statistics	Parameters to collect statistics at the interface level.

3.3.4 Interface Properties Editor

The properties editor for an interface can be opened from the **Interfaces** tab in the Table View panel by doing one of the following:

1. Double-click on the row for the interface.

2. Right-click on the row for the interface and click **Properties**.

The interface properties editor is shown in Figure 3-40 and its tabs are described below.

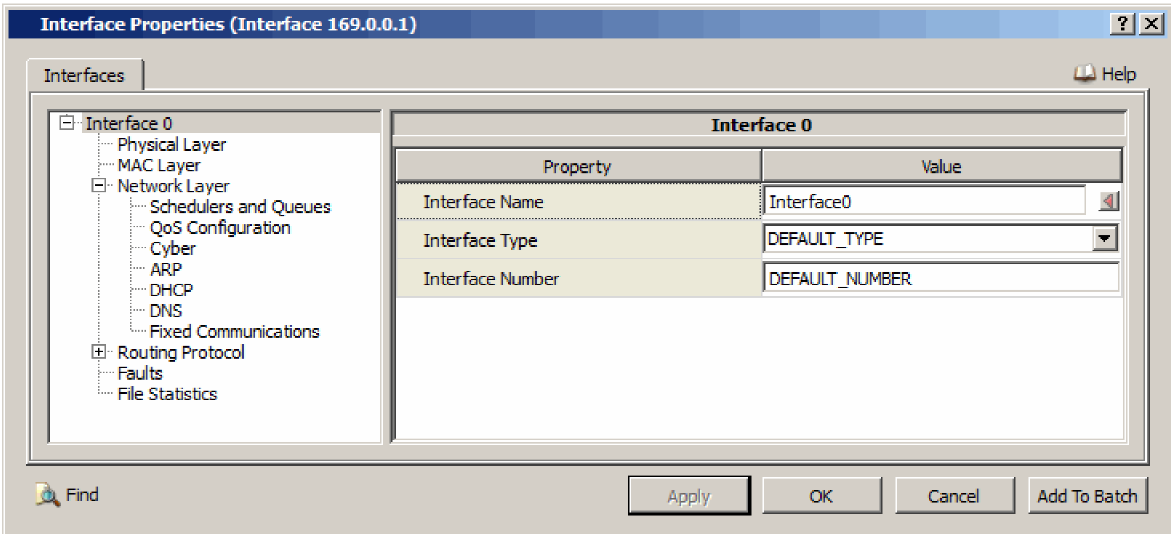


FIGURE 3-40. Interface Properties Editor

Interfaces Tab
This tab is the same as the Interfaces tab of the Default Device Properties Editor (see Section 3.3.3).

3.3.5 Wireless Subnet Properties Editor

The properties editor for a wireless subnet can be opened by one of the following ways:

1. From the canvas, select the wireless subnet on the canvas, right-click and choose **Properties**, or double-click on the wireless subnet.
2. From the **Networks** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the wireless subnet.
   b. Right-click on the row for the wireless subnet and choose **Properties**.

The wireless subnet properties editor is shown in Figure 3-41 and its tabs are described below.

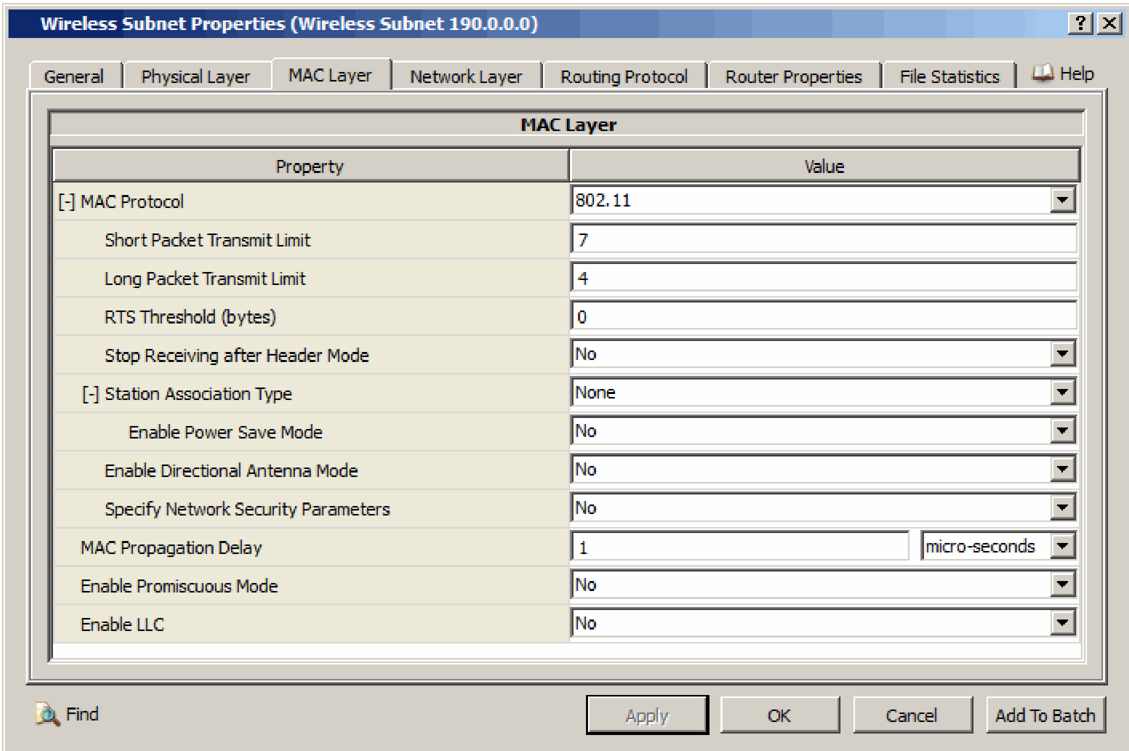


FIGURE 3-41. Wireless Subnet Properties Editor

Tab	Description
General	Used to set the 2D and 3D icons.
Physical Layer	This tab is for radio and antenna models and other Physical Layer parameters at the subnet level.
MAC Layer	This tab is for MAC protocol and other MAC layer parameters at the subnet level.
Network Layer	This tab is for network protocol, queues, schedulers, and other Network Layer parameters at the subnet level.
Routing Protocol	This tab is for routing-related parameters, including unicast and multicast routing protocols at the subnet level.
Router Properties	This tab is for configuring Router model properties at the subnet level.
File Statistics	This tab is for enabling file statistics for the subnet.

3.3.6 Wired Subnet Properties Editor

The properties editor for a wired subnet can be opened by one of the following ways:

1. From the canvas, select a wired subnet (Hub) on the canvas, right-click and choose **Properties**, or double-click on the wired subnet.
2. From the **Networks** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the wired subnet network.
   b. Right-click on the row for the wired subnet and choose **Properties**.

The wired subnet properties editor is shown in Figure 3-42 and its tabs are described below.

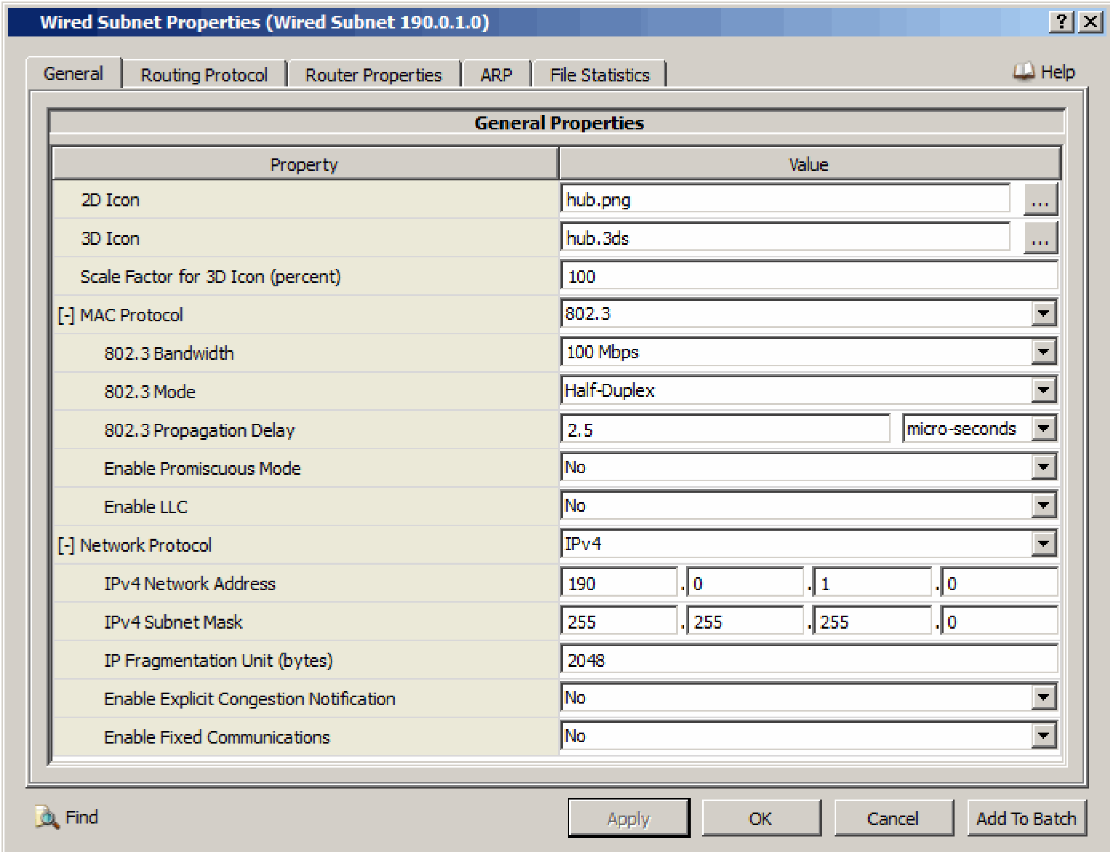


FIGURE 3-42. Wired Subnet Properties Editor

Tab	Description
General	This tab is for configuring properties of a wired subnet, such as the IP address and subnet mask, MAC protocol, and Network protocol parameters.
Routing Protocol	This tab is for configuring routing-related parameters, including unicast and multicast routing protocols at the subnet level.
Router Properties	This tab is for configuring Router model properties at the subnet level.
ARP	This tab is used to configure ARP properties.
File Statistics	This tab is used to enable file statistics for the subnet.

3.3.7 Point-to-Point Link Properties Editor

The properties editor for a point-to-point link can be opened by one of the following ways:

1. From the canvas, select the point-to-point link on the canvas, right-click and choose **Properties**, or double-click on the point-to-point link.
2. From the **Networks** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the point-to-point link.
   b. Right-click on the row for the point-to-point link and choose **Properties**.

The point-to-point link properties editor is shown in Figure 3-43 and its tabs are described below.

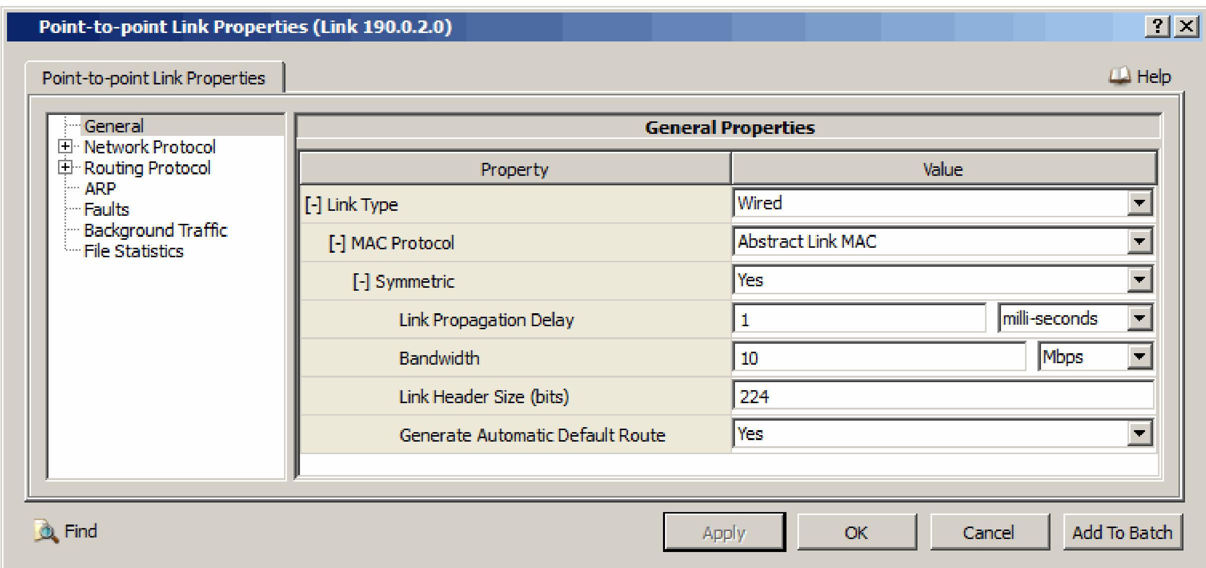


FIGURE 3-43. Point-to-Point Link Properties Editor

Tab	Description
General	This group is for configuring link type (wired, wireless or microwave), MAC protocol, and Virtual LAN (VLAN) parameters.
Network Protocol	This group is for specifying the network protocol and corresponding network address and subnet mask.
Routing Protocol	Routing-related parameters, including multicast routing protocols at the link level.
ARP	This group is for ARP configuration for the point-to-point link.
Faults	Specifies faults for interfaces at both ends of the link.
Background Traffic	Specifies the background traffic on the link.
File Statistics	Enables file statistics for the link.

3.3.8 Satellite Properties Editor

The properties editor for a satellite can be opened by one of the following ways:

1. From the canvas, select the satellite on the canvas, right-click and choose **Properties**, or double-click on the satellite.
2. From the **Nodes** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the satellite.
   b. Right-click on the row for the satellite and choose **Properties**.

The satellite properties editor is shown in Figure 3-44 and its tabs are described below.

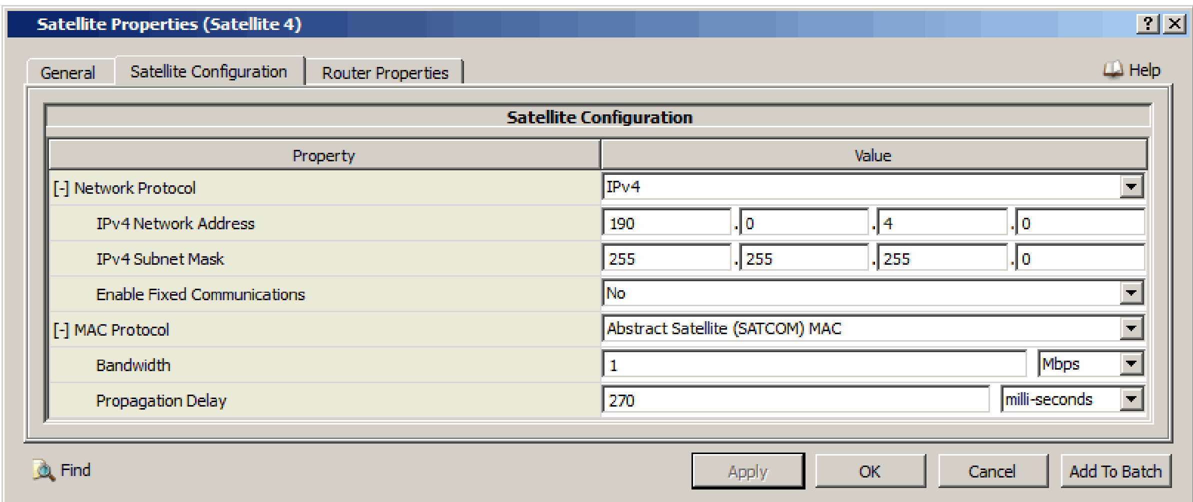


FIGURE 3-44. Satellite Properties Editor

Tab	Description
General	This tab is for name, icons, and partition parameters.
Satellite Configuration	This tab is for MAC and network protocol parameters for the satellite.
Router Properties	This tab is used to set the router type for the satellite.

3.3.9 Switch Properties Editor

The properties editor for a switch can be opened by one of the following ways:

1. From the canvas, select the switch on the canvas, and choose **Properties**, or double-click on the switch.
2. From the **Nodes** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the switch.
   b. Right-click on the row for the switch and choose **Properties**.

The switch properties editor is shown in Figure 3-45 and its tabs are described below.

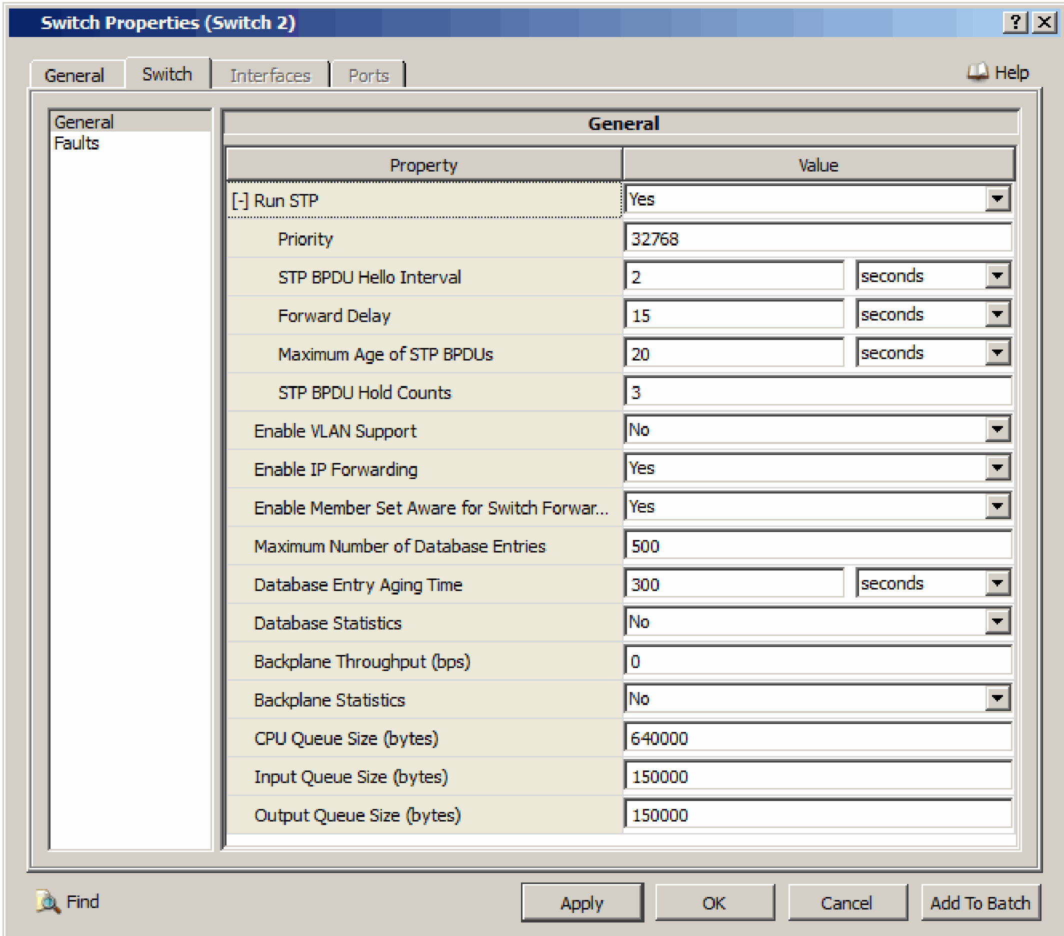


FIGURE 3-45. Switch Properties Editor

Tab	Description
General	This tab is for configuring basic switch properties, such as name and icons.
Switch	This tab has the following property groups: • General: Spanning tree, VLAN, and queue parameters • Faults: Switch faults
Interfaces	This tab is for configuring interface properties such as the interface name and type and IP address and subnet mask and for configuring interface faults.
Ports	This tab is available only when a switch is connected to devices and is used for configuring port level properties. This tab has the following property groups for each port: • STP: Spanning tree parameters • VLAN: VLAN parameters

3.3.10 ATM Device Properties Editor

The properties editor for an ATM device can be opened by one of the following ways:

1. From the canvas, select the ATM node on the canvas, right-click and choose **Properties**, or double-click on the ATM node.
2. From the **Nodes** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the ATM node.
   b. Right-click on the row for the ATM node and choose **Properties**.

The **ATM Device Properties** editor is shown in Figure 3-46 and its tabs are described below.

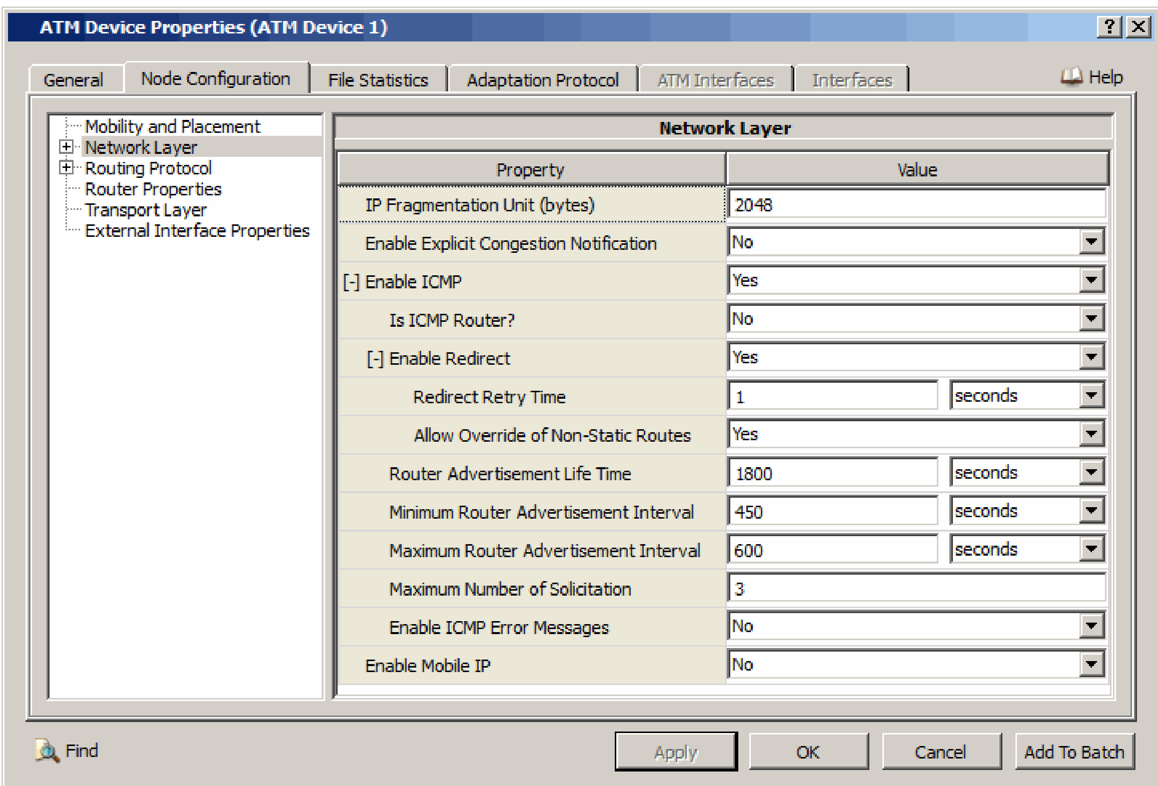


FIGURE 3-46. ATM Device Properties Editor

Tab	Description
General	This tab is for configuring ATM device properties such as node name, icons, partition, and device type (ATM end system or ATM switch).
Node Configuration	This tab has the following property groups: • Mobility and Placement: Mobility model and related parameters • Network Layer: Network protocol, queues, schedulers, and other Network Layer parameters • Routing Protocol: Routing-related parameters, including unicast and multicast routing protocols • Router Properties**:** Router configuration parameters • Transport Layer: Transport protocol (UDP, TCP, or RSVP) configuration parameters
File Statistics	Parameters to collect different statistics at the node level
Adaptation Protocol	This tab provides options to configure ATM Adaptation Layer properties
ATM Interfaces	This tab is available only when the ATM device is connected to another ATM device. This tab has the following property groups for each ATM interface: • ATM Layer 2: ATM layer 2, queue, and scheduler parameters • Adaptation Protocol: ATM Adaptation Protocol parameters • ARP Specific: ARP parameter
Interfaces	This tab is available only when an ATM end system is connected to a device of a different (non-ATM) type. This tab is the same as the Interfaces tab of the Default Device Properties Editor (see Section 3.3.3).

3.3.11 ATM Link Properties Editor

The properties editor for an ATM link can be opened by one of the following ways:

1. From the canvas, select the ATM link on the canvas, right-click and choose **Properties**, or double-click on the ATM link.

2. From the **Networks** tab in the **Table View** panel, do one of the following:

   a. Double-click on the row for the ATM link.

   b. Right-click on the row for the ATM link and choose **Properties**.

The **ATM Link Properties** editor is shown in Figure 3-47 and its tabs are described below.

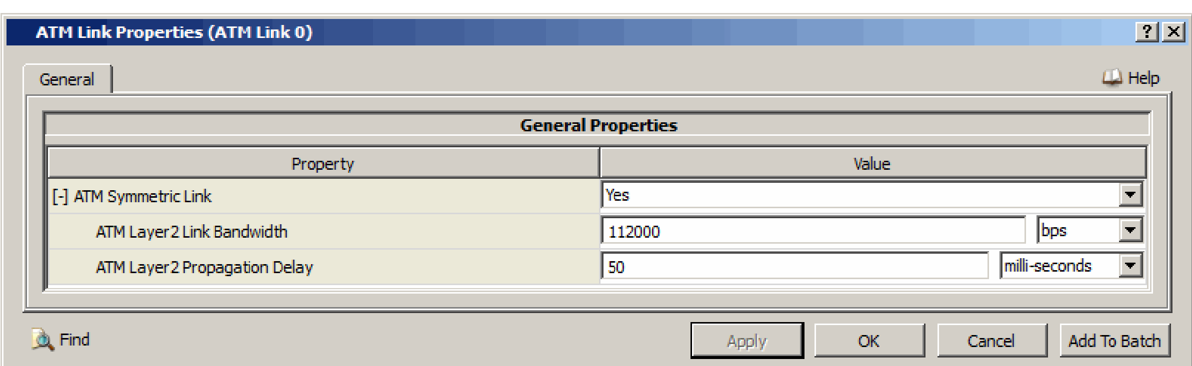


FIGURE 3-47. ATM Link Properties Editor

Tab	Description
General	This tab is for configuring the bandwidth and propagation delay of the ATM link.

3.3.12 Hierarchy Properties Editor

The properties editor for a hierarchy can be opened by one of the following ways:

1. From the canvas, select the hierarchy on the canvas, right click, and choose **Properties**.
2. From the **Hierarchies** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the hierarchy link.
   b. Right-click on the row for the hierarchy and choose **Properties**.

The **Hierarchy Properties** editor is shown in Figure 3-48 and its tabs are described below.

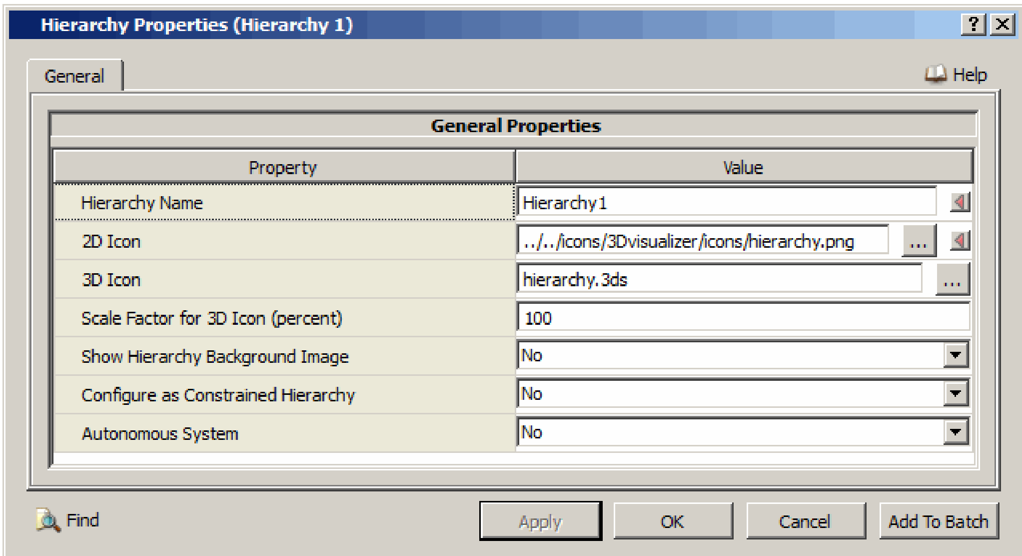


FIGURE 3-48. Hierarchy Properties Editor

Tab	Description
General	This tab is for configuring hierarchy parameters such as name, icons, background images, and autonomous system specifications. Refer to the description of Border Gateway Protocol in *Multimedia and Enterprise Model Library* for configuring autonomous systems.

3.3.13 Application Properties Editors

The scenario properties of applications are used to configure parameters such as source, destination, start and end times, and traffic characteristics. The properties editors of different applications are described in the model libraries.

Figure 3-49 shows the properties editor for the Voice over IP application. The other application properties editors have a similar layout.

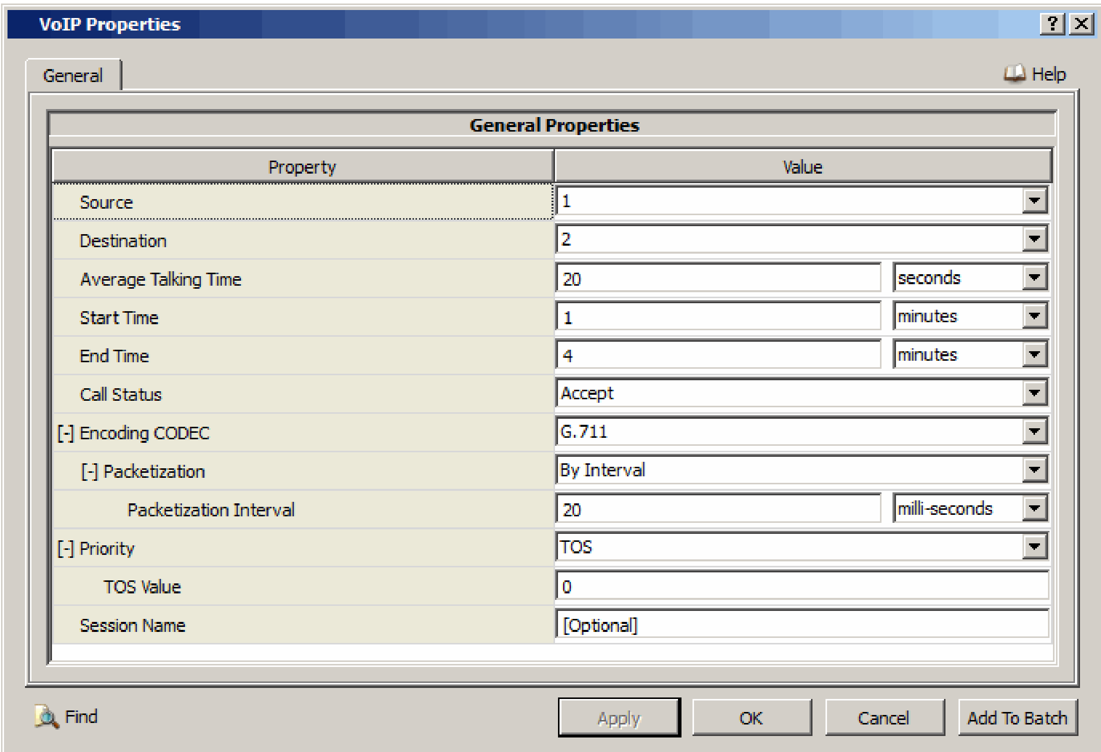


FIGURE 3-49. Voice over IP Application Properties Editor

3.4 Advanced Features

This section describes some advanced features of Architect in Design mode.

3.4.1 Multiple Select and Move

Multiple objects can be selected and moved using the **Select** or **Lasso** tools. Once these objects are selected, they can be moved as a group or their properties can be modified as a group.

Using the Select Arrow

The **Select** arrow is used for selecting objects individually, such as units (devices, subnets, switches, hierarchies, etc.) and links (communication links and application links). When a link is selected, it appears in red. The **Select** arrow can also be used to select all objects within a rectangular region.

1. To select a rectangular region, click on the **Select** arrow, position the cursor at one corner of the region, left-click the mouse, drag the cursor to the diagonally opposite corner and release the mouse. All objects within the rectangle are selected (see Figure 3-50).
2. Select multiple objects by keeping the control key pressed while clicking on the objects.
3. Deselect objects by left-clicking the mouse anywhere on the window.

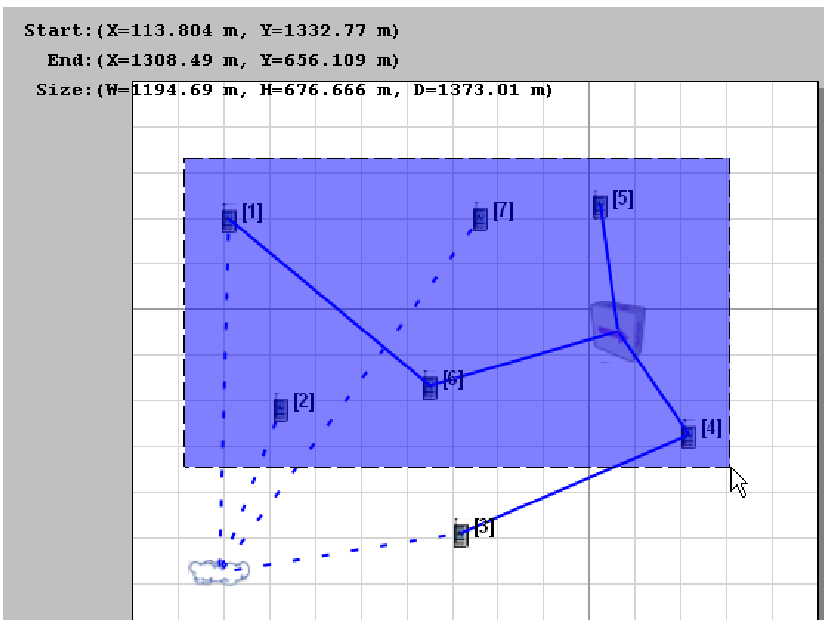


FIGURE 3-50. Selecting a Rectangular Region Using Select Button

When a region is selected, coordinate information displays showing the following information:

- **Start**: The starting point of the selected area
- **End**: The end point of the selected area
- **Size**: Width, height, and length of the diagonal of the selected rectangular region, in meters

Notes: Groups of objects can also be selected by pressing the control key while clicking on individual objects, as well as typing **Ctrl+A** to select all.

Using the Lasso

The **Lasso** button can also be used to select objects individually or to select all objects within a region of an arbitrary shape.

1. To select a region of an arbitrary shape, click the **Lasso** button. Left-click the mouse and select the desired region by tracing the region with the cursor, then release the mouse. All objects within the traced region are selected (see Figure 3-51).
2. Select multiple objects by keeping the control key pressed while clicking on the objects.
3. Deselect objects by left-clicking the mouse anywhere on the window.

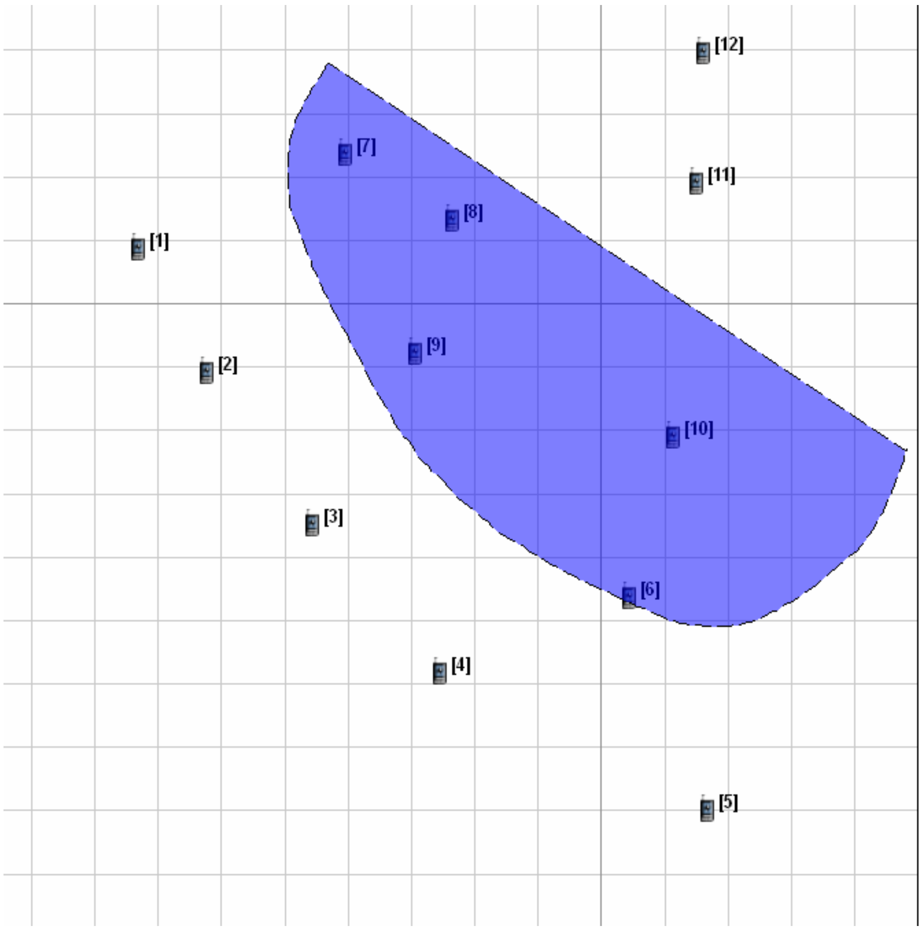


FIGURE 3-51. Selecting an Irregular Region Using Lasso Button

Moving Objects

To move a selected object, left-click on it and drag it to a new location. To move a group of selected objects, left-click on any one of them and move it to the new location. All selected objects will move maintaining their relative positions.

3.4.2 Recording and Using Multiple Scenario Views

In 3D view, it is possible to record scenario views from multiple camera positions and switch between them. This provides a convenient way to navigate to different parts of the scenario terrain.

Recording Scenario Views

To record the current scenario view, do one of the following:

- Press the “C” key
- Right click on the canvas and select **Save Current View**

Pressing the **Saved Views** button on the **View** toolbar displays the list of saved views. The most recent nine recorded views are saved.

Note: The list of saved views contains scenario views saved from all scenarios, not just the current scenario.

Refreshing Saved Views

When a scenario view is saved, the current scenario view is added to the list of saved views. The current camera position is also recorded with the saved view. The list of saved views may contain views (and camera positions) saved from other scenarios. Hence, when the **Saved Views** button is pressed, the scenario views that are displayed in the list may be from multiple scenarios.

To apply the camera positions recorded with the saved views to the current scenario and regenerate the views displayed in the list, do one of the following:

- Press the “V” key
- Right click on the canvas and select **Reload All Saved Views**

Switching Between Views

To switch from the current scenario view to one generated by a recorded camera position, do one of the following:

- Click the **Saved Views** button in the **View** Toolbar and select a view from the displayed list. The camera position recorded with the selected view will be applied to the current scenario.
- Press one of the keys “1” to “9” to select a recorded camera position.

Deleting Recorded Camera Positions

To delete all recorded camera positions, do one of the following:

- Press the “0” key (in Design and Visualize modes)
- Right click on the canvas and select **Delete All Saved Views** (only in Design mode)

3.4.3 Modifying Properties of Multiple Objects

The properties of multiple objects of the same type can be modified as a group. This is a convenient way of applying the same properties to a group of objects.

To assign the same value for a parameter for multiple objects of the same type:

1. Select the objects on the canvas or in the **Table View** panel.
2. Right-click on one of the selected objects and select **Properties**. This launches the group properties editor of the type of selected objects. Figure 3-52 shows the **Group Default Device Properties** editor.

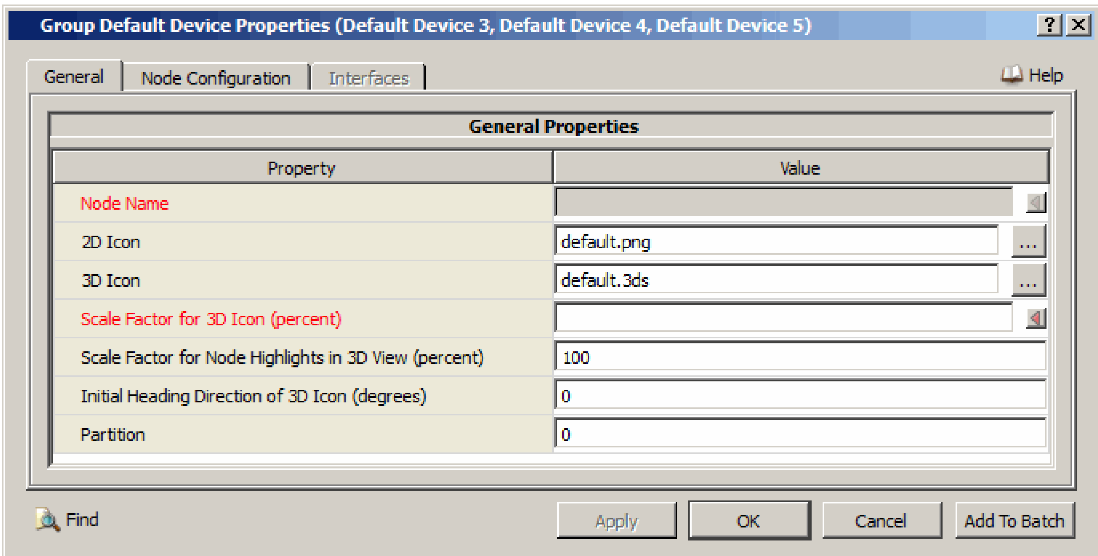


FIGURE 3-52. Group Default Properties Window

3. Edit the properties as needed and click **OK** to apply the changes to all of the selected objects.

Note:
1. If the selection includes objects of different types, the **Properties** menu entry will be disabled.
2. Properties that are unique to each object (such as **Node Name** in the **General** tab of the **Group Default Device Properties** editor) appear in red and the corresponding Value field is grayed out indicating that a value can not be entered in the field.
3. For properties that need not be unique, if the values for different objects are different, then the property name appears in red. However, a new value can be entered for the property and that value will apply to all objects in the group.

3.4.4 Defining Node Groups

Node groups provide an alternative way of modifying properties of multiple nodes. Nodes that are logically related can be placed in the same node group and the same set of properties can be conveniently applied to all nodes in the group. Node groups are also used for defining mobility parameters for the Group Mobility model (see Section 4.2.6).

Creating Node Groups

To create a node group, do the following:

1. Place nodes on the canvas by using the **Node Placement Wizard** (under the Tools menu) or by placing nodes individually (see Section 3.2.1.1).
2. Open the **Table View** panel and select the **Groups** tab (see Figure 3-54).
3. Click the **Add Group** button. This opens a dialog box (see Figure 3-53) in which you can set a name for the group and specify the nodes that belong to the group.

Note: You can also select the nodes on the canvas to associate together as a group, and then click the **Add Group** button. The selected nodes are automatically added to the new group.

4. The members of the group can be defined by listing them one after another, each separated by white-space character(s) or commas. A range of nodes can be defined by using the keyword "`thru`". For example: A node member definition "`1, 2, 5 thru 10`" includes nodes 1, 2, 5, 6, 7, 8, 9, and 10.

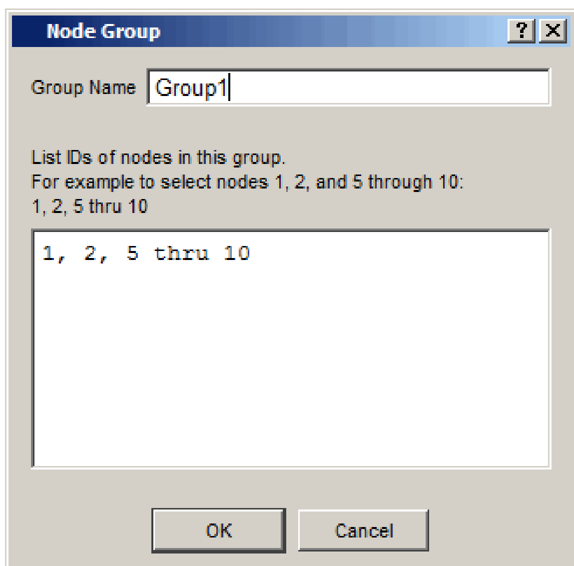


FIGURE 3-53. Creating a Node Group

5. Click the **OK** button to save the group.

Figure 3-54 shows the membership of two node groups in the **Groups** tab of the **Table View** panel.

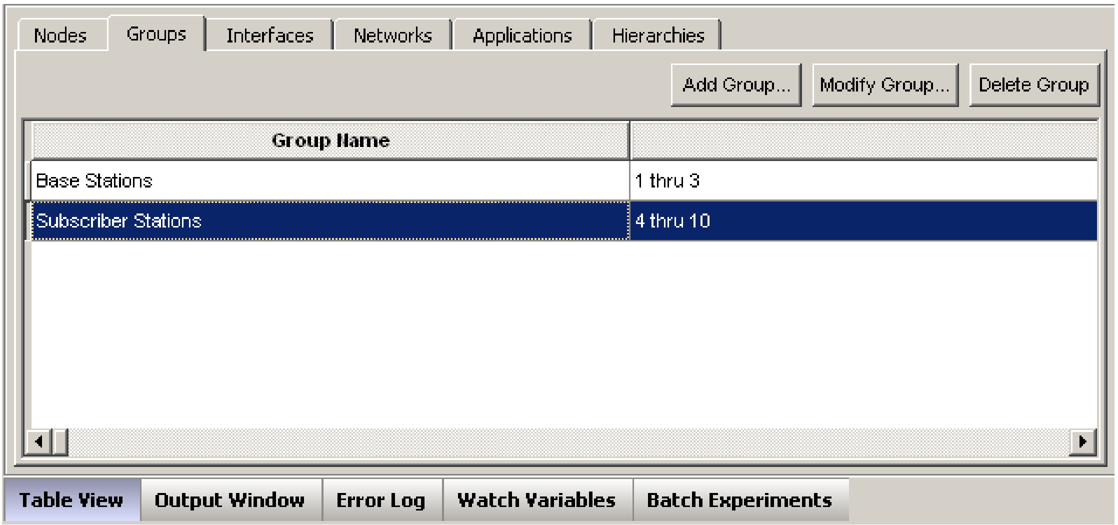


FIGURE 3-54. Node Groups

Modifying and Deleting Node Groups

To modify the members of a group, select the group in the **Groups** tab and click the **Modify Group** button.

To delete a group, select the group and click the **Delete Group** button.

Note: Deleting a node group does not delete its member nodes.

Modifying Properties for Nodes in a Group

To modify properties for all nodes in a group, do the following:

1. Select the node group in the table. All member nodes of the group become selected and are highlighted on the canvas.
2. Right-click on the row and select **Properties** or double-click on the row. This opens the **Group Default Device Properties** editor.
3. Edit the properties, as described in Section 3.4.3.

3.4.5 Setting Mobility Waypoints

The mobility pattern of a mobile object (such as a device, satellite, or weather pattern) can be specified by setting waypoints. To set a waypoint, a location and a time need to be specified: the object will be at the specified location at the specified time. From one waypoint to the next, the object moves in a straight line at a constant speed that is determined by the two waypoint locations and times.

To set waypoints for an object, perform the following steps:

1. Select the **Waypoint**  button from the **Other Components** toolbar of the **Standard Toolset**.
2. Select the object for which you want to set waypoints by left-clicking on it.

3. Next, left-click on the canvas at the desired location for the first waypoint. A waypoint marker is placed at the waypoint location and a line is drawn between the object and the waypoint marker.
4. Click on the canvas at the location of the next waypoint. A waypoint marker is placed at that location and it is connected by a line to the previous waypoint. Similarly, place subsequent waypoints on the canvas. See Figure 3-55.

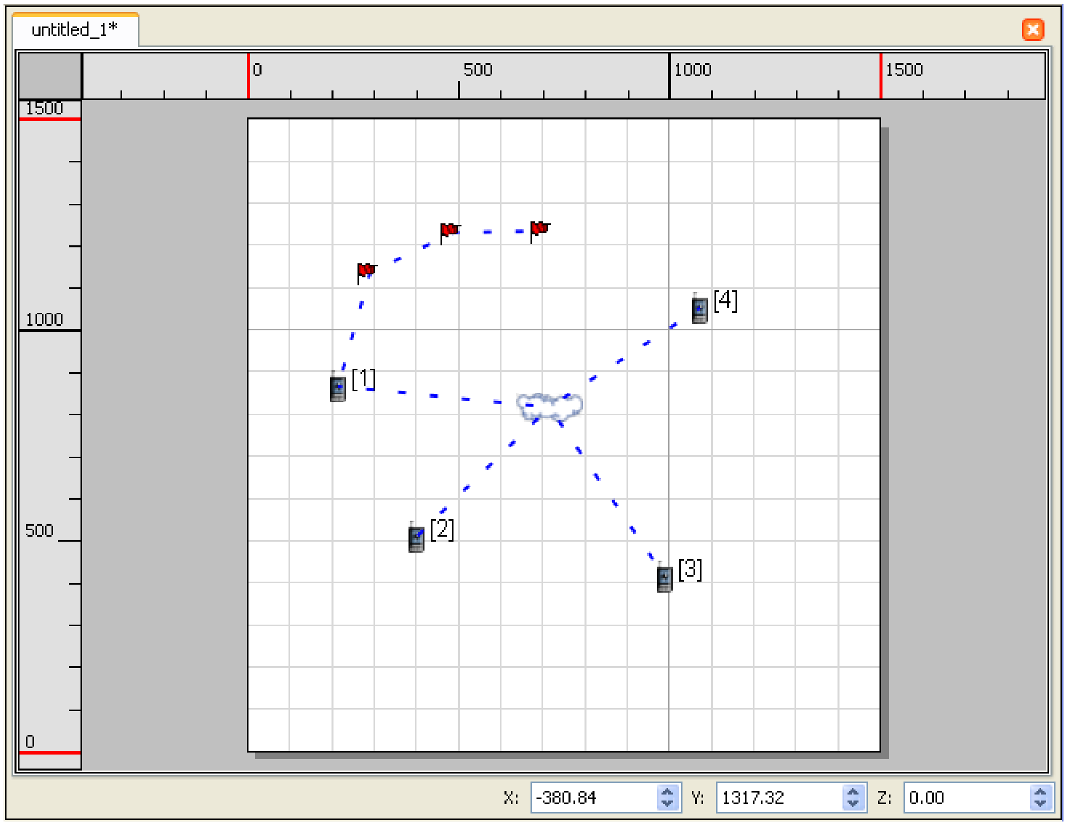


FIGURE 3-55. Setting Mobility Waypoints

5. After adding the last waypoint, click the right mouse button.
6. To specify the waypoint times, open the **Mobility Waypoint Editor** by right-clicking on any waypoint marker and selecting properties. Enter the waypoint times in increasing order, as described in Section 3.4.5.1.

Adding Additional Waypoints

If waypoints have already been added for a node and you want to add additional waypoints, perform the following steps:

1. Select the **Waypoint** button from the **Other Components** toolbar of the **Standard Toolset**.
2. Select the object for which you want to add waypoints by clicking on it.
3. Click on the canvas at the desired location for the new waypoint. A waypoint marker is placed at the waypoint location and a line is drawn between the waypoint marker and the previous last waypoint.
4. Continue adding waypoints in a similar way. Right-click to exit from insert waypoint mode.
5. Open the **Mobility Waypoint Editor** and enter the time for the new waypoints.

Note: Waypoints can only be added at the end. You can not add a new waypoint between two existing waypoints.

You can also add waypoints by using the **Mobility Waypoint Editor** (see Section 3.4.5.1).

Moving and Deleting Waypoints

A waypoint can be deleted or moved by selecting the waypoint marker and deleting or moving it, just like any other object on the canvas. Any waypoint can be moved or deleted.

Group operations (select, move, delete) can be performed on waypoint markers, just like other objects.

A waypoint can be moved by using the **Mobility Waypoint Editor** (see Section 3.4.5.1).

3.4.5.1 Mobility Waypoint Editor

The **Mobility Waypoint Editor** is used to specify waypoint times. It can also be used to change waypoint coordinates and orientation, to add new waypoints, and to delete waypoints. It can also be used to configure waypoints for nodes as well as weather effects.

To open the **Mobility Waypoint Editor**, right-click on a waypoint marker and select **Properties**. All waypoints for all nodes are displayed in the editor. Figure 3-56 shows the Mobility Waypoint Editor for a scenario with four nodes.

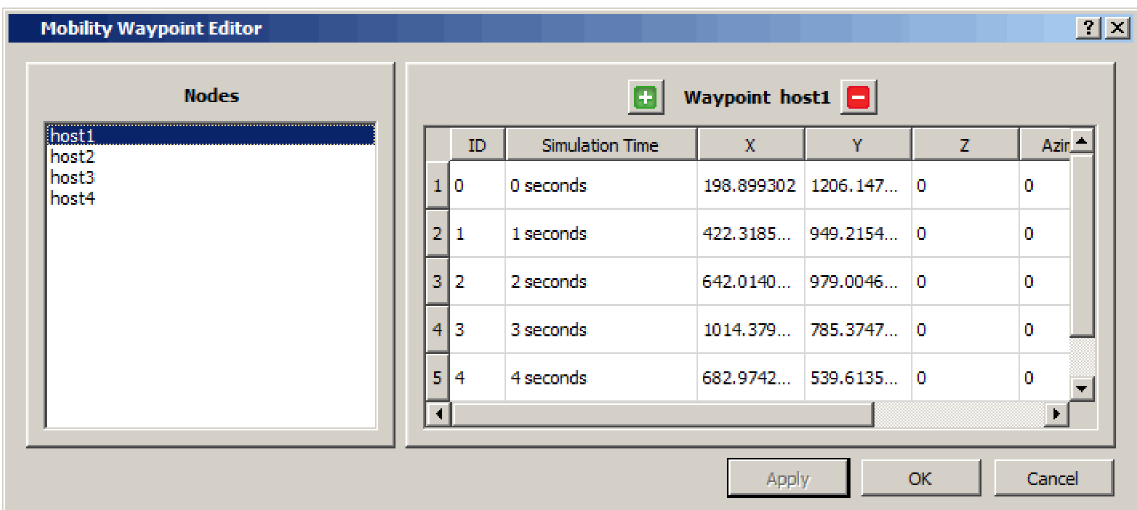


FIGURE 3-56. Mobility Waypoint Editor

- To display the waypoints for a node, select the node from the left panel of the **Mobility Waypoint Editor**. The waypoints for the selected node are displayed in the right panel.
- To specify a waypoint time, enter the numeric time value and select a time unit in the table.
- To modify the waypoint coordinates and orientation (azimuth and elevation), enter the new values in the table.
- To add a new waypoint, click the button. A new waypoint is added at the end of the waypoints table.
- To delete a waypoint, select the waypoint and click the button.

Note: All waypoints for a particular node should be in ascending order of time in the waypoints table. Also, the initial node position (position at time 0) is not listed in the waypoints table. The times associated with waypoints should be greater than 0.

3.4.6 Configuring Weather Patterns

You may specify weather patterns that move and affect propagation. Currently, the weather pattern model supports latitude/longitude coordinates.

A weather pattern is configured by drawing its shape on the canvas (in the form of a polygon).

To specify a weather pattern, perform the following steps:

1. Select the **Weather Pattern**  button from the **Other Components** toolbar of the **Standard Toolset**.
2. Draw the polygon representing the weather pattern by clicking on the canvas locations corresponding to the corners of the polygon. When you add a new point, lines is drawn from the last point to new point and the very first point to the new point.
3. After placing the last corner of the polygon, press the right mouse button.
4. You can start drawing another weather pattern polygon by clicking on the canvas locations corresponding to the corners of the new weather pattern.
5. To end the weather pattern insert mode and switch to select mode, select a different component in the toolset or click the **Select**, **Lasso**, or **Region Zoom** buttons on the toolbar or press the Escape key.

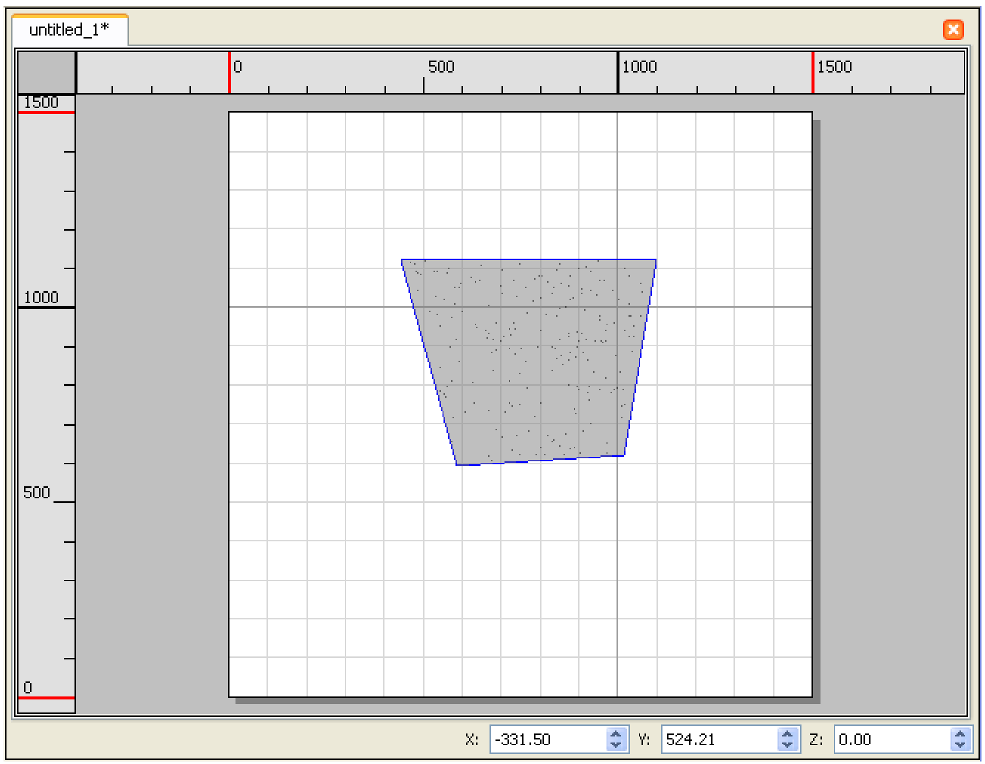


FIGURE 3-57. Drawing a Weather Pattern

Mobility can be added to a weather pattern in the same manner as a node (see Section 3.4.5). Note that after clicking on a weather object with the waypoint tool selected, a shadow of the weather object will follow the mouse pointer to indicate what area the weather will cover.

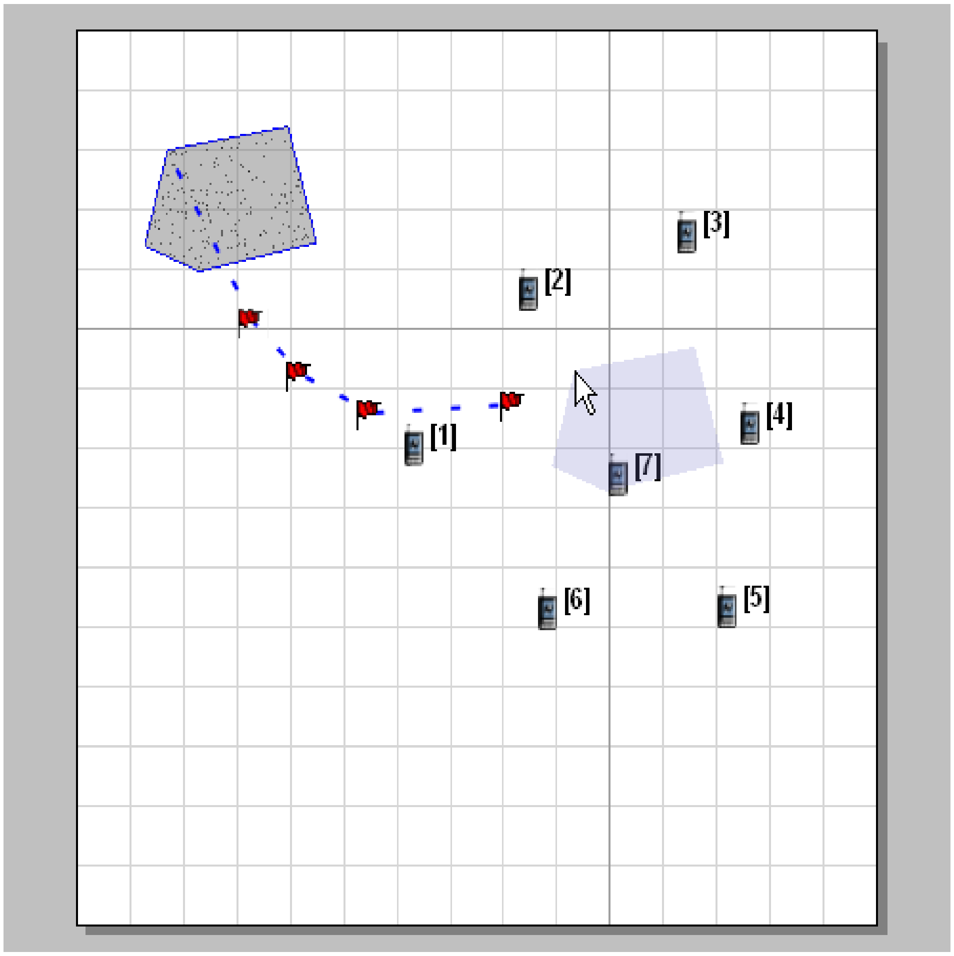


FIGURE 3-58. Adding Mobility to Weather Pattern

Properties of weather patterns can be specified using the **Weather Property** editor. Refer to *Wireless Model Library* for details.

3.4.7 Building Hierarchies

Hierarchies are used in EXata GUI to aid in the design of large network scenarios. A hierarchy represents another view of the canvas. If the hierarchy is unconstrained, it represents a view of the entire canvas. If the hierarchy is constrained, it represents a view of a section of the canvas. Typically, objects that have a logical or topological relationship among them are put into a hierarchy. The entire hierarchy itself is represented by a single icon in the scenario. Objects in a hierarchy can be viewed by opening the hierarchy, which can be done by double-clicking on the hierarchy icon on the canvas. Thus, hierarchies can be used to design scenarios in a modular fashion.

Note: Hierarchies can be nested, i.e., a hierarchy can contain another hierarchy. There is no limit on the level of nesting.

Hierarchies serve as a visual aid that enable viewing subsets of the scenario independently. Hierarchies do not impose any constraint on how different network objects can be connected. For example, a node that is in a hierarchy can be connected to a wireless subnet icon that is placed on the main canvas.

Creating Unconstrained Hierarchies

To create an unconstrained hierarchy, do the following:

1. Select the **Hierarchy**  button from the **Network Components** toolbar in the toolset.
2. Place a hierarchy icon anywhere on the canvas.
3. Open the hierarchy by double-clicking on the icon or by right-clicking and selecting **Open Hierarchy**. This opens the hierarchy window, which is another view of the canvas.
4. Place network components that are part of the hierarchy in the hierarchy window and connect them just as you would on the canvas.

Creating Constrained Hierarchies

To create a constrained hierarchy, do the following:

1. Select the **Constrained Hierarchy**  button from the **Network Components** toolbar in the toolset.
2. Position the cursor at one corner of the area to be covered by the hierarchy, drag the cursor to the diagonally opposite corner, and release the mouse. A gray rectangle is drawn on the canvas to represent the area covered by the hierarchy (see Figure 3-59).

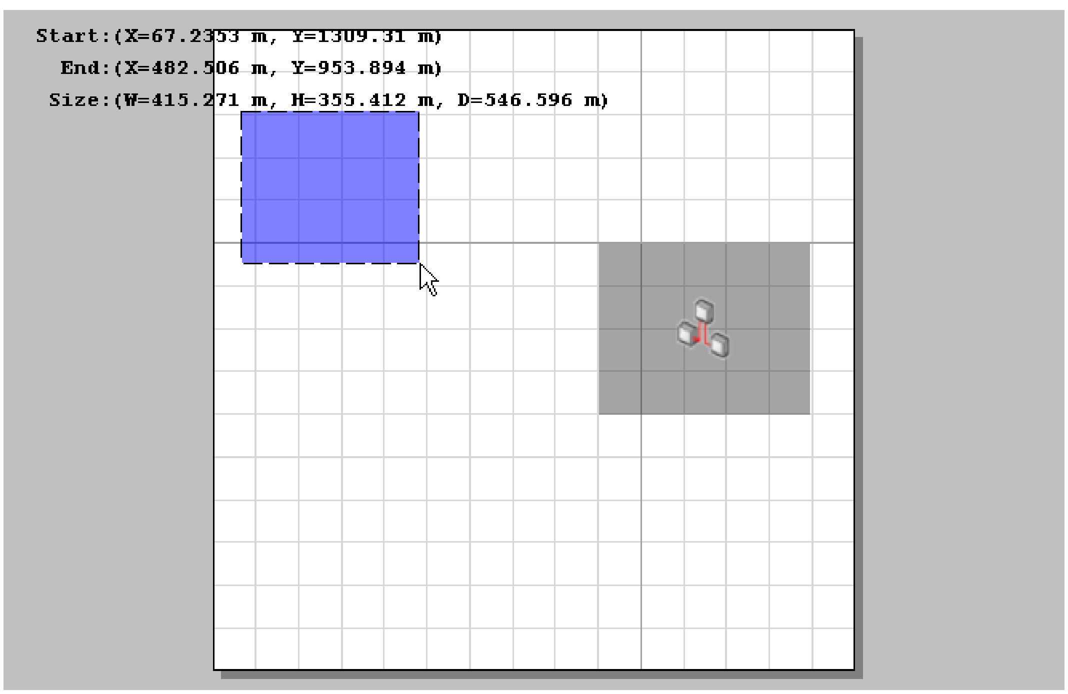


FIGURE 3-59. Creating Unconstrained Hierarchies

3. Open the hierarchy by double-clicking on rectangle or by right-clicking and selecting **Open Hierarchy**. This opens the hierarchy window. Only the part of the canvas that is covered by the hierarchy is shown in this window.
4. Place network components that are part of the hierarchy in the hierarchy window and connect them just as you would on the canvas.

Connecting to Objects across Hierarchies

To connect an object from outside the hierarchy to an object inside the hierarchy, draw a link from one of the objects to the other.

A link connecting two objects in different hierarchies is represented by two line segments: one line segment inside the hierarchy window from the connected object to the upper left corner of the hierarchy window and the other outside the hierarchy from the connected object to the hierarchy icon.

> **Note:** When drawing a line between objects across hierarchies, only the first segment of the line is visible.

Figure 3-60 shows a scenario consisting of three nodes (host1, host2, and host3) and a hierarchy (Hierarchy1). (The scenario itself is treated as a top-level hierarchy and is labeled Hierarchy0 by default.) The three nodes in Hierarchy0 are connected to a wireless subnet. Hierarchy1 itself has three nodes (host4, host5, and host6) which are connected to a wireless subnet.

Figure 3-60 shows the state when a point-to-point link is being drawn between host3 and host4. Note that only the line segment on the main canvas is visible. Figure 3-61 shows the state after the link has been drawn.

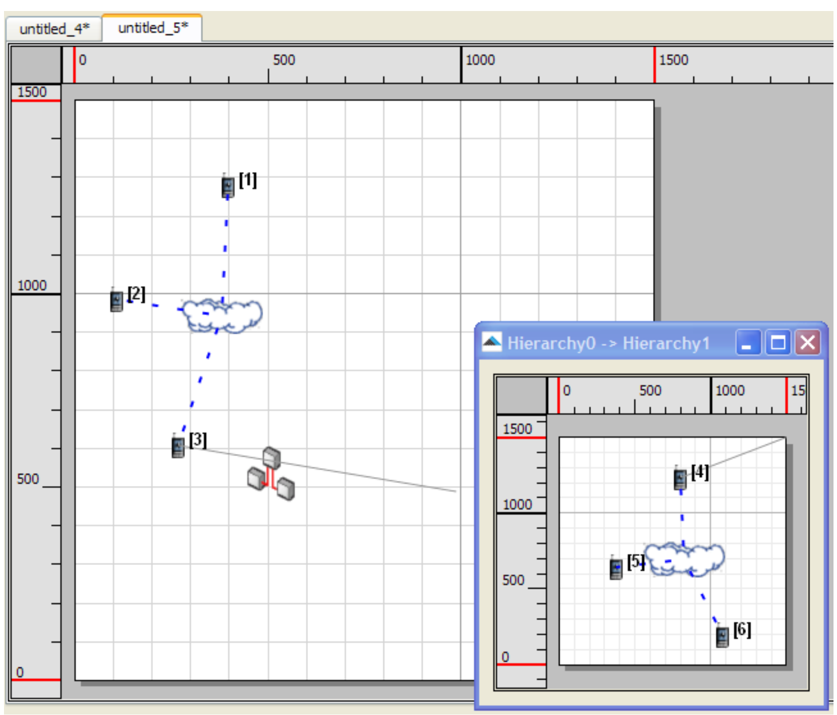


FIGURE 3-60. Creating a Link across Hierarchies

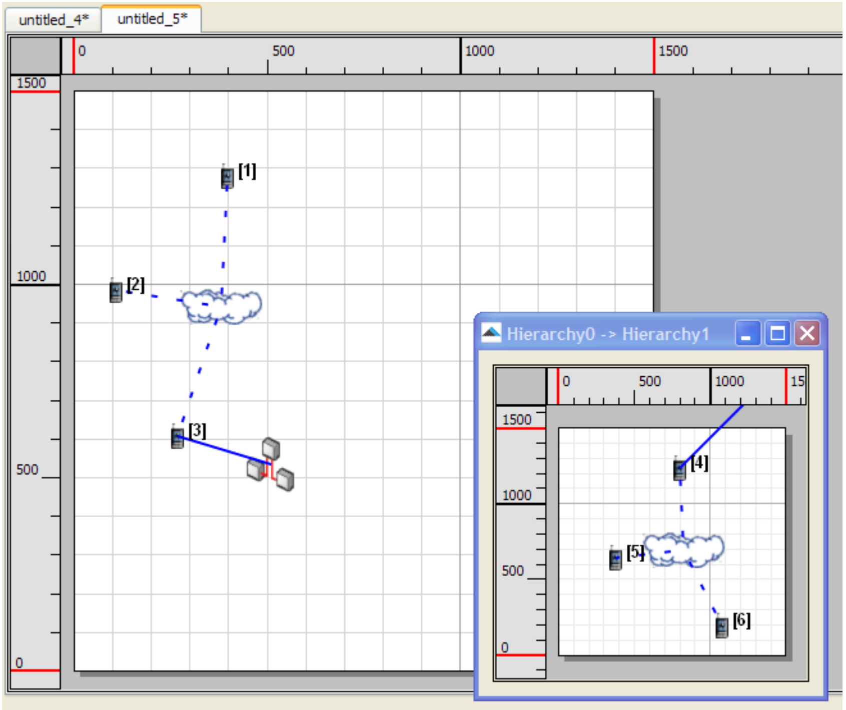


FIGURE 3-61. Representation of a Link across Hierarchies

Creating Nested Hierarchies

To created nested hierarchies, do the following:

1. Open a hierarchy and place a hierarchy icon in it.
2. Double-click on this hierarchy icon to open it and place network components in it. All operations can be performed on the nested hierarchy, including connecting to objects outside the hierarchy. Hierarchies can be nested to arbitrary depths.

Figure 3-62 shows an example of nested hierarchies. Hierarchy1 consists of three devices connected to a wireless subnet and a child hierarchy (Hierarchy2). Hierarchy2 consists of three devices connected to a wireless subnet. There is a point-point-link between a device in Hierachy1 and a device in Hierarchy2. There is also a point-to-point link between Hierarchy1 and the main (default) hierarchy.

Note: You can also close a hierarchy window by pressing the escape key, in addition to the usual ways of closing a window.

To set the properties of a hierarchy (including designating it as a constrained hierarchy), use the **Hierarchy Properties** editor, as explained in Section 3.4.7.1

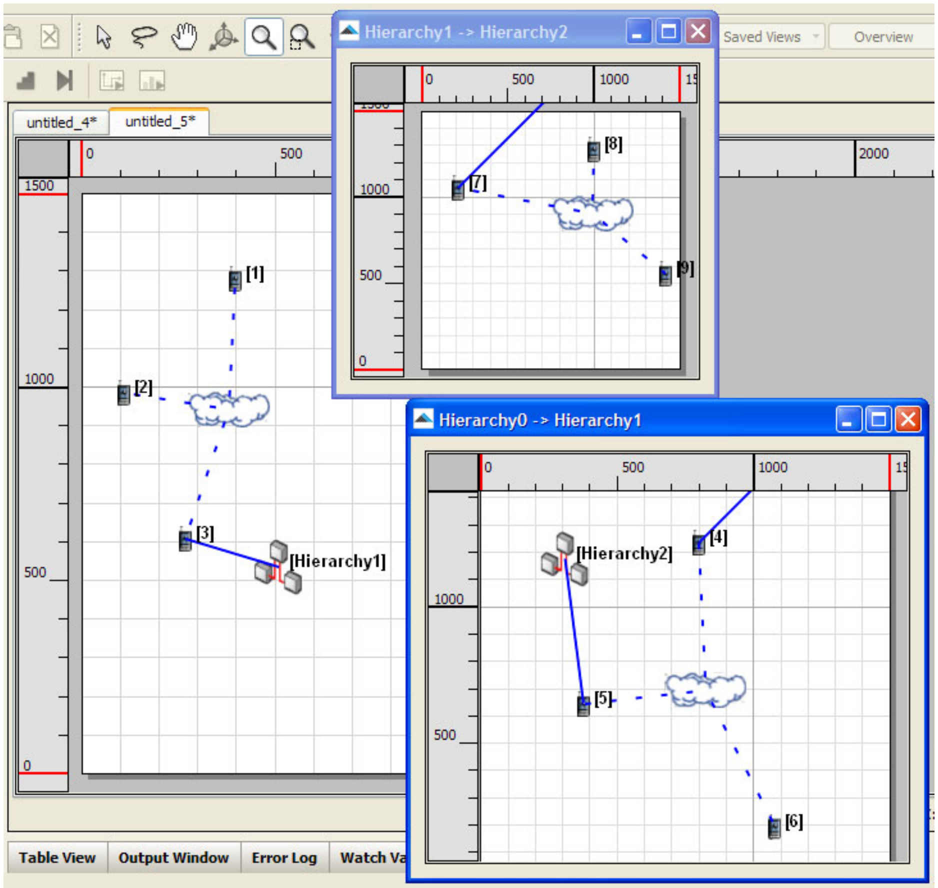


FIGURE 3-62. Nested Hierarchies

3.4.7.1 Hierarchy Properties Editor

The **Hierarchy Properties** editor is used to modify the properties of hierarchies.

The **Hierarchy Properties** editor can be opened in one of the following ways:

1. From the canvas, select the hierarchy icon, right-click, and choose **Properties**.
2. From the **Hierarchies** tab in the **Table View** panel, do one of the following:
   a. Double-click on the row for the hierarchy.
   b. Right-click on the row for the hierarchy and choose **Properties**.

The **Hierarchy Properties** editor is shown in Figure 3-63. Attributes such as the hierarchy name, background image, and icons can be specified using the hierarchy properties editor.

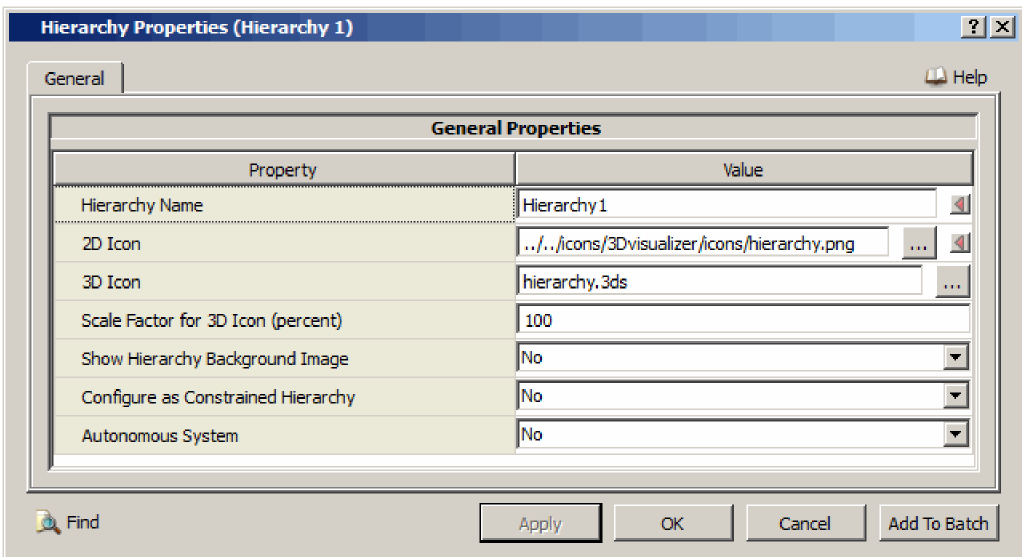


FIGURE 3-63. Hierarchy Properties Editor

Autonomous Systems

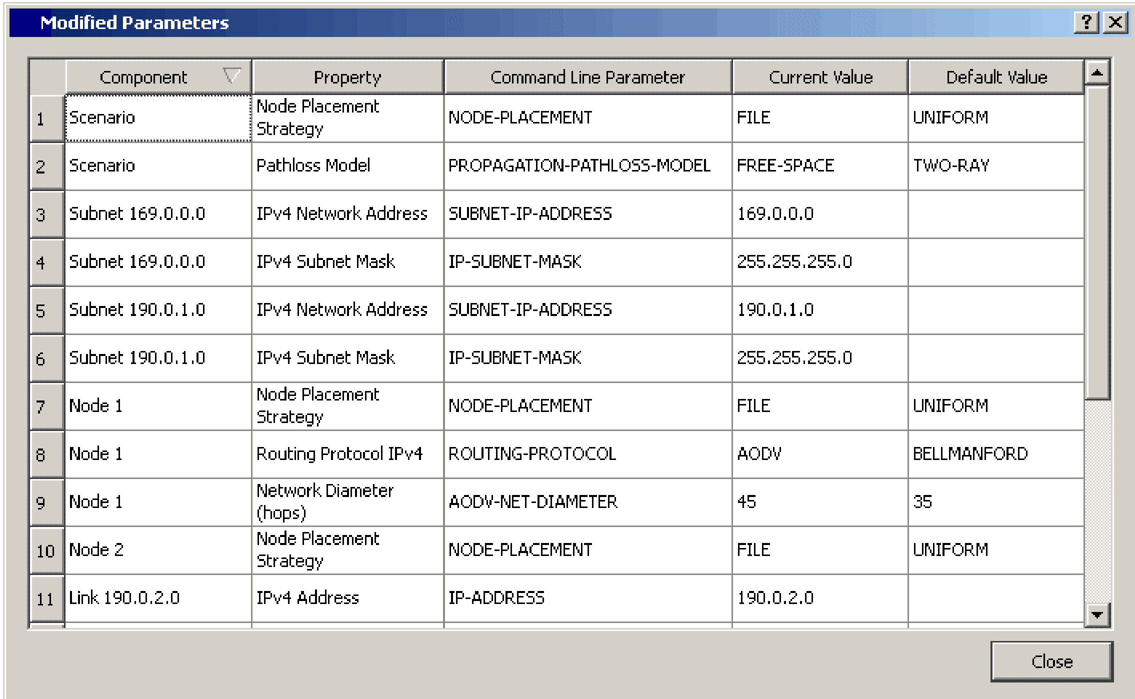
An autonomous system (AS) is part of the routing infrastructure of a large IP internetwork. An AS is essentially a portion of a large internetwork whose routing is administered by a single authority. These autonomous systems are connected by the Internet's core routers, which use an exterior routing protocol called Border Gateway Protocol (BGP) for communication among themselves. See *Multimedia and Enterprise Model Library* for details.

To designate a hierarchy as an autonomous system, set **Autonomous System** to *Yes*. The ID of the autonomous system can be specified in the AS-ID field.

3.4.8 Viewing Modified Parameters

The **Modified Parameters** window displays a list all scenario parameters that have been set to values that are different from their default values. (Chapter 4 describes how scenario parameters are set.)

To open the **Modified Parameters** window, select **Tools > Modified Parameters** from the menu bar.



Modified Parameters

	Component	Property	Command Line Parameter	Current Value	Default Value
1	Scenario	Node Placement Strategy	NODE-PLACEMENT	FILE	UNIFORM
2	Scenario	Pathloss Model	PROPAGATION-PATHLOSS-MODEL	FREE-SPACE	TWO-RAY
3	Subnet 169.0.0.0	IPv4 Network Address	SUBNET-IP-ADDRESS	169.0.0.0	
4	Subnet 169.0.0.0	IPv4 Subnet Mask	IP-SUBNET-MASK	255.255.255.0	
5	Subnet 190.0.1.0	IPv4 Network Address	SUBNET-IP-ADDRESS	190.0.1.0	
6	Subnet 190.0.1.0	IPv4 Subnet Mask	IP-SUBNET-MASK	255.255.255.0	
7	Node 1	Node Placement Strategy	NODE-PLACEMENT	FILE	UNIFORM
8	Node 1	Routing Protocol IPv4	ROUTING-PROTOCOL	AODV	BELLMANFORD
9	Node 1	Network Diameter (hops)	AODV-NET-DIAMETER	45	35
10	Node 2	Node Placement Strategy	NODE-PLACEMENT	FILE	UNIFORM
11	Link 190.0.2.0	IPv4 Address	IP-ADDRESS	190.0.2.0	

Close

FIGURE 3-64. Modified Parameters

The **Component** column lists the properties editor in which the parameter is set. The **Property** column displays the name of the parameter as it appears in the properties editor. The **Command Line Parameter** column shows the equivalent command line parameter for the property. The **Current Value** and **Default Value** columns list the current value and default value of the parameter.

To sort the data based on any of the columns, click the column's header to sort ascending, and click the same column header again to sort descending.

Double-clicking a row of the table opens the properties editor in which the parameter is set, enabling you to conveniently change its value.

3.4.9 Configuring Batch Experiments

The batch experiment capability allows you to configure several experiments based on a scenario and run them as a batch. The core elements of the scenario (network components and topology) are the same in all experiments. The experiments differ from each other in the values of one or more configurable parameters (*batch parameters*).

The **Batch Experiments panel** is used to configure batch experiments. To configure batch experiments, first configure the scenario. Then add the batch parameters to the batch parameter table in the **Batch Experiments** panel. Next, specify the values for each batch parameter. Note that each combination of batch parameter values constitutes an experiment.

Note: If the batch parameter is a dependent parameter of a controlling parameter, then the controlling parameter should be added first to the batch parameter table.

Figure 3-65 shows the **Batch Experiments** panel before adding any batch parameters.

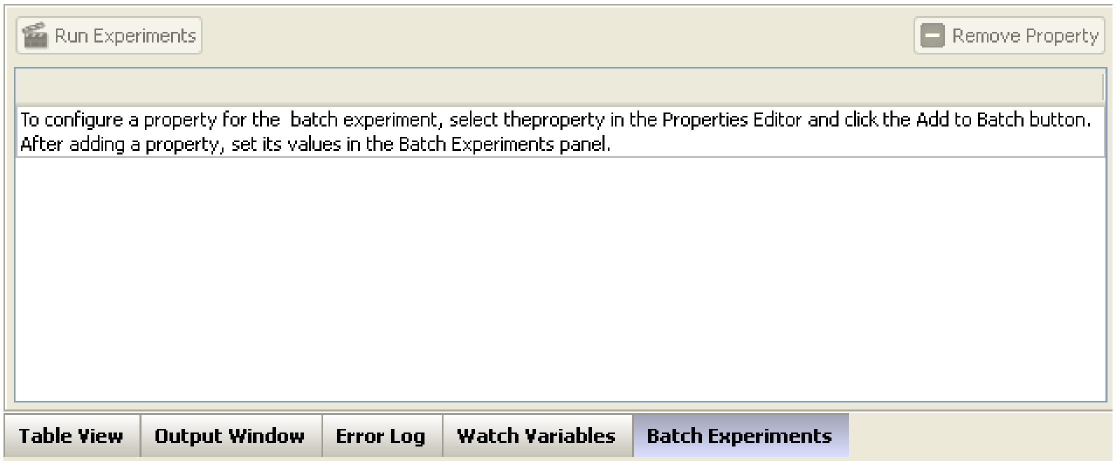


FIGURE 3-65. Batch Experiments Panel

To configure and run batch experiments, perform the following steps:

1. Configure a scenario, as described in Section 4.2.
2. Open the properties editor that contains the first batch parameter. For example, if you want to run experiments with different values for the AODV Hello Interval for a particular node, open the **Default Device Properties** editor for that node.
3. Open the tab and/or parameter group that contains the batch parameter. Select the batch parameter by clicking on it, and then clicking the **Add To Batch** button. This adds the parameter to the batch parameter table. If the selected parameter is dependent on a controlling parameter, then you will have to first add the controlling parameter to the batch parameter table.

   To add the node-level parameter AODV Hello Interval to a batch experiment:

   a. Go to **Default Device Properties Editor > Node Configuration > Routing Protocol**.
   b. Set **Routing Protocol IPv4** to *AODV*. This displays the AODV parameters.
   c. Select the parameter **Routing Protocol IPv4** and add it to the batch parameter table by clicking the **Add to Batch** button. Similarly, add the parameters **Enable Hello Messages** and **Hello Interval** to the batch parameter table.

Note: You can not add the parameter **Hello Interval** to the batch parameter table without first adding the parameters **Routing Protocol IPv4** and **Enable Hello Messages** because parameter **Hello Interval** is a dependent parameter of the controlling parameter **Enable Hello Messages,** which is dependent on controlling parameter **Enable Hello Messages**.

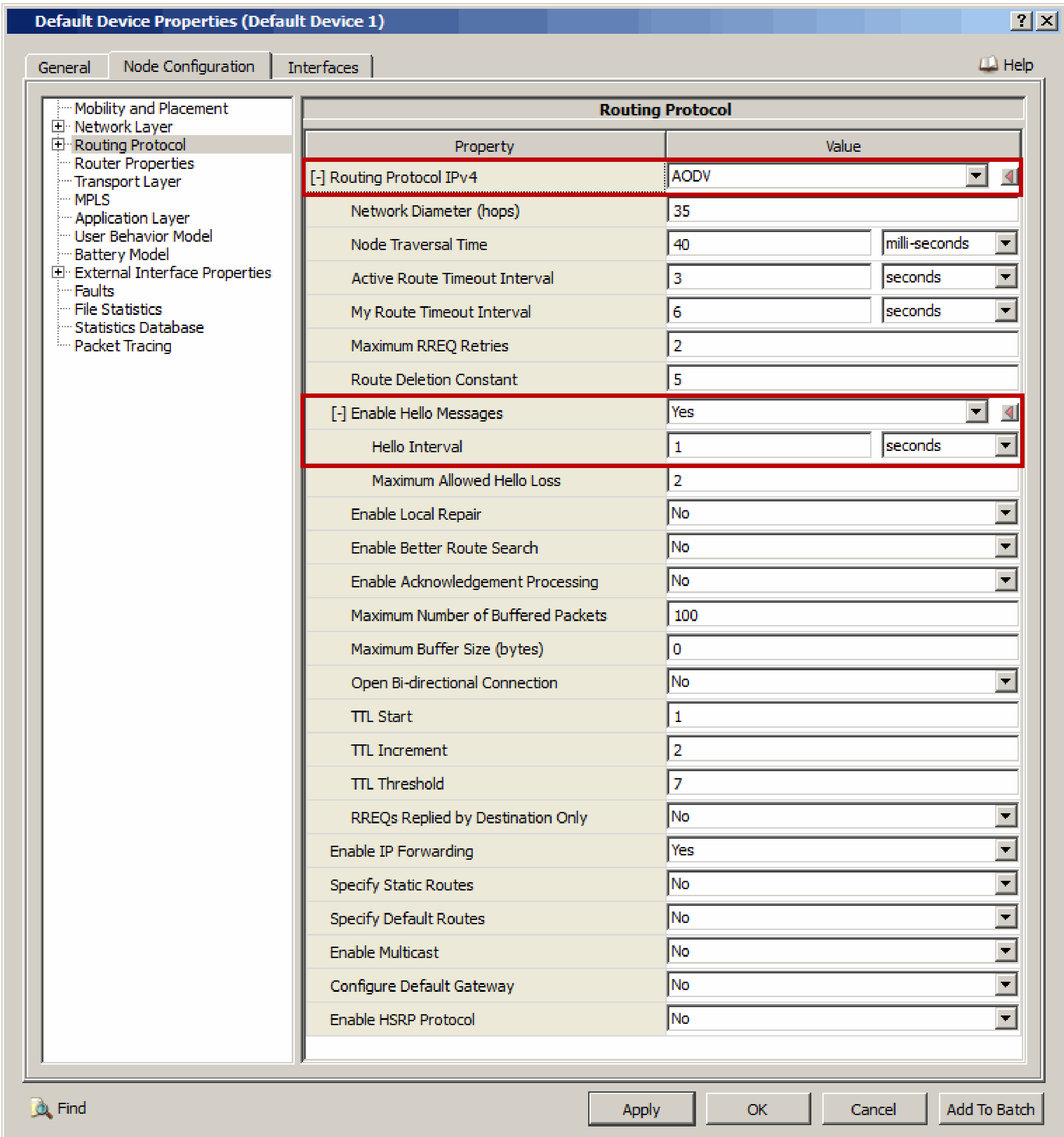


FIGURE 3-66. Adding Parameters to a Batch Experiment

4. Close the properties editor.
5. Add the other batch parameters to the batch parameter table in a similar way.

Figure 3-67 shows the batch parameter table after adding **Routing Protocol IPv4** and AODV parameters **My Route Timeout Interval**, **Enable Hello Messages**, and **Hello Interval**.

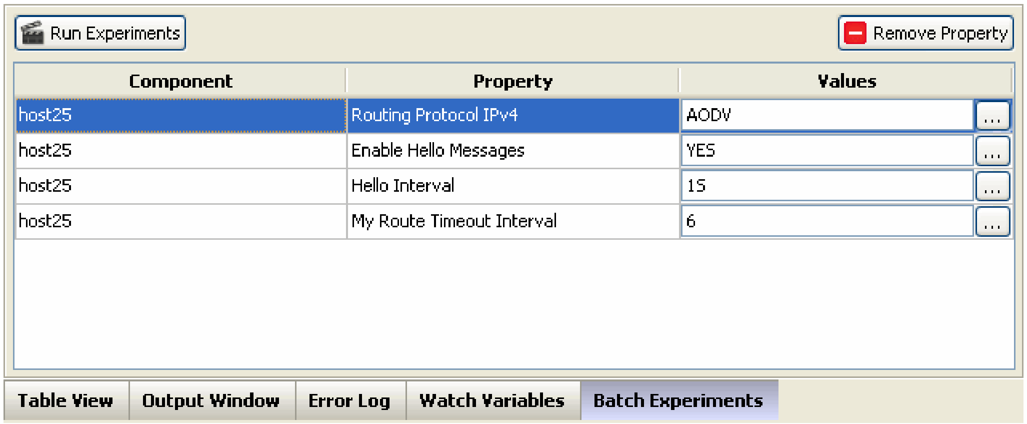


FIGURE 3-67. Batch Parameter Table

6. Next, specify the values for each batch parameter. To do this, select the batch parameter in the batch experiments table and click the ... button in the Value column. This opens an editor for specifying values of the batch parameter.

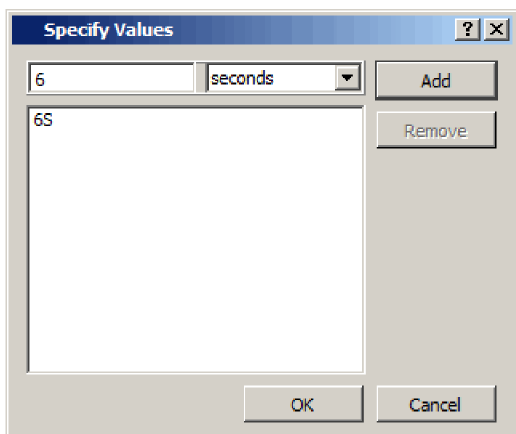


FIGURE 3-68. Specifying Values for a Batch Parameter

7. In the values editor, enter or select a value for the batch parameter and click the **Add** button. The added value is displayed in the list. Similarly, add the other values for the batch parameter.
8. To remove a value from the list, select the value from the list and click the **Remove** button.
9. After adding all values for the batch parameter, close the values editor.

10. Specify the values for all batch parameters in a similar way.
Figure 3-69 shows the batch parameter table after specifying multiple values for the AODV parameters **My Route Timeout Interval** and **Hello Interval**.

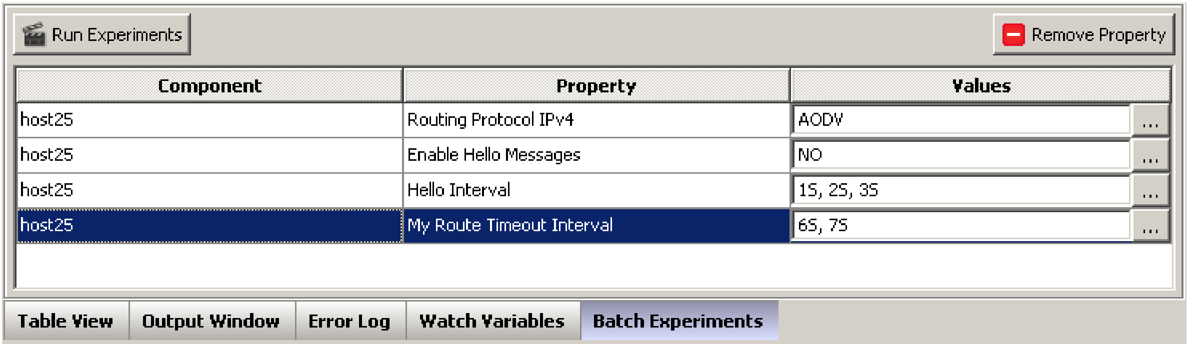


FIGURE 3-69. Batch Parameter Table with Multiple Values of Batch Parameters

11. To remove a parameter from the batch parameter table, select the parameter and click the **Remove Property** button.

12. Click the **Run Experiments** 

button.

13. In the **Batch Run Mode** window that is displayed, select either the **Interactive** or **Non-Interactive** button and then click **OK** to run the batch experiment. These two modes of batch execution are described below.

Interactive and Non-interactive Execution

If the batch experiment is run interactively, the experiments in the batch are loaded in Architect one at a time.The experiment can be run by pressing the **Play** button. The scenario is executed with animation. After one experiment has finished executing, the next experiment in the batch is loaded in a different tab and waits for user input to start running.

If the batch experiment is run non-interactively, all experiments in the batch are executed one after the other without user input. The scenario execution is not animated. However, the progress of the simulation is displayed in the **Output Window** of Architect.

Names of Experiments and Output Files

Experiments in a batch are called Experiment-1, Experiment-2, etc., and the files associated with each experiment are named accordingly. Thus, input files associated with the first experiment are called, Experiment-1.config, Experiment-1.app, Experiment-1.nodes, etc. The statistics file generated for the first experiment is called Experiment-1.<*date_time*>.stat, where <*date_time*> denotes the date and time when the experiment is run.

The files associated with a batch experiment are stored in a sub-folder called BatchRun in the scenario folder.

Note: The contents of the batch experiment folder, including all statistics files, are overwritten every time you run another batch experiment.

3.5 Customization

This section describes the customization features of Architect and contains the following sections:

- Creating Custom Network Object Models
- Creating Custom Hierarchy Models
- Creating and Customizing Toolsets

3.5.1 Creating Custom Network Object Models

Using the **Device Model Editor**, custom models for network objects (devices, wired and wireless subnets, satellites, and switches) can be created. Properties can be assigned to the custom model and a button can be added in the toolset for the model, using which the model can be conveniently used in scenarios. Properties of custom network object models can also be modified using the **Device Model Editor**.

To open the **Device Model Editor**, select **Tools > Device Model Editor**. The left panel displays the list of existing network object models (see Figure 3-70).

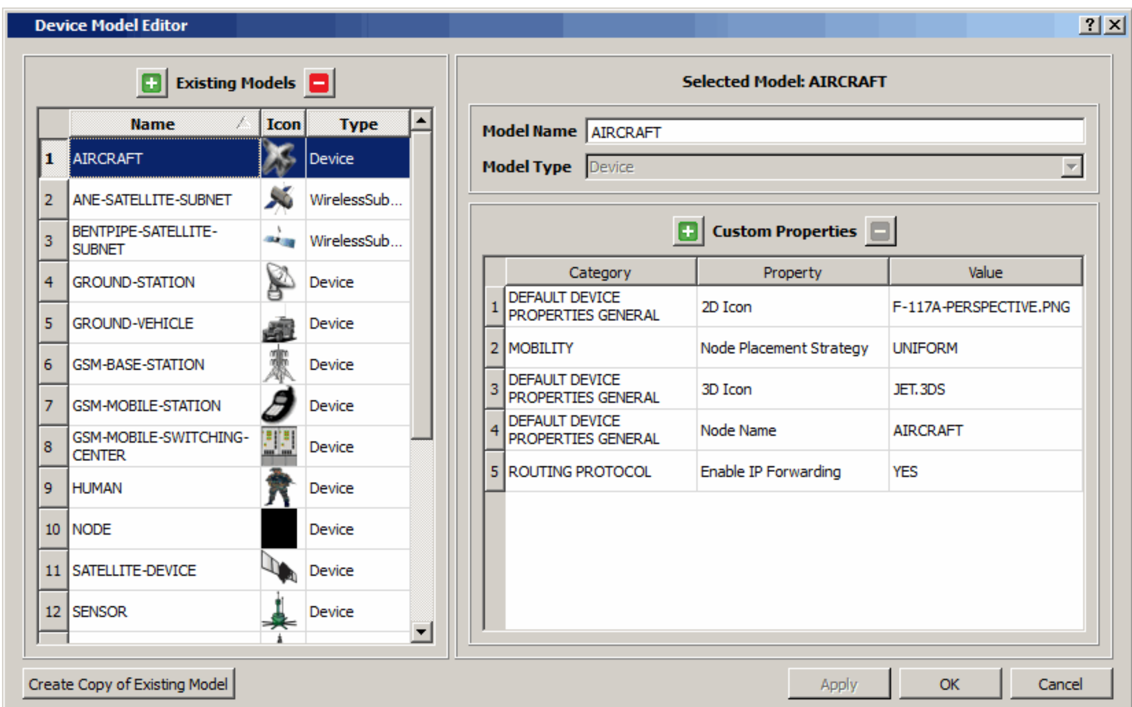


FIGURE 3-70. Device Model Editor

Creating a New Custom Network Object Model

To create a new custom network object model, perform the following steps:

1. Open the **Device Model Editor**.
2. Click the button in the left panel and enter the model name in the **Model Name** field.

3. In the **Model Type** field, select the type of the model (device, wired subnet, wireless subnet, satellite, or switch) from the drop-down list.

   A *Device* model type has the same set of properties as the Default Device selected from the **Devices** toolbar of the toolset. A model of type *WiredSubnet*, *WirelessSubnet*, or *Satellite* has the same set of properties as the Wired Subnet, Wireless Subnet, or Satellite, respectively, selected from the **Network Components** toolbar of the toolset. A *Switch* model type has the same set of properties as the Switch selected from the **Devices** toolbar of the toolset.

   Default values for all properties are used for the custom model. Non-default values for properties can be set as described in the following steps.

4. To assign non-default values to the properties of the custom model, click the  button next to **Custom Properties**. This opens the **Device Model Property Editor**.

   The properties displayed by the editor depend on the type of the model. For example, if **Model Type** is set to *Device*, then the **Device Model Property Editor** (see Figure 3-71) is the same as the **Default Device Properties** editor (except that the Interfaces tab is disabled).

5. To associate an icon with the custom model, in the **2D Icon** field, enter the name (including the full path) of the image file to be associated with the model.

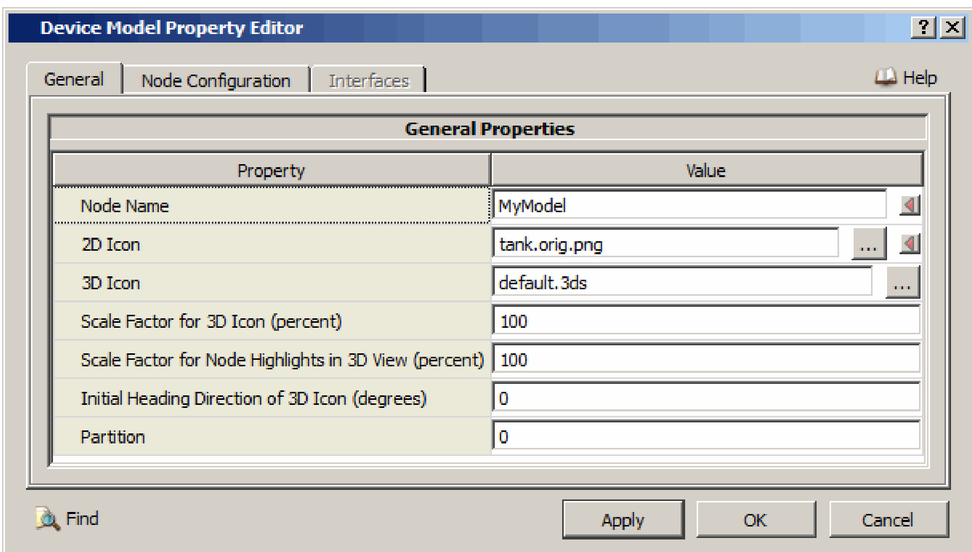


FIGURE 3-71. Device Model Property Editor for Model of Type Device

6. Set the properties to their desired values in the **Device Model Property Editor**. See Section 3.3 for details on setting properties. Click **Apply** or **OK** to save the assigned values.

   All properties that were modified in the previous step are displayed in the right panel along with their values. Properties that were not modified are not displayed.

   Note: Default values are used for all properties that are not displayed in the right panel.

Figure 3-72 shows the Device Model Editor after creating a customized wireless subnet (called MAC-SUBNET) with **RADIO TYPE** and **MAC PROTOCOL** set to *MACDOT11E* and *PHY-ABSTRACT*, respectively.

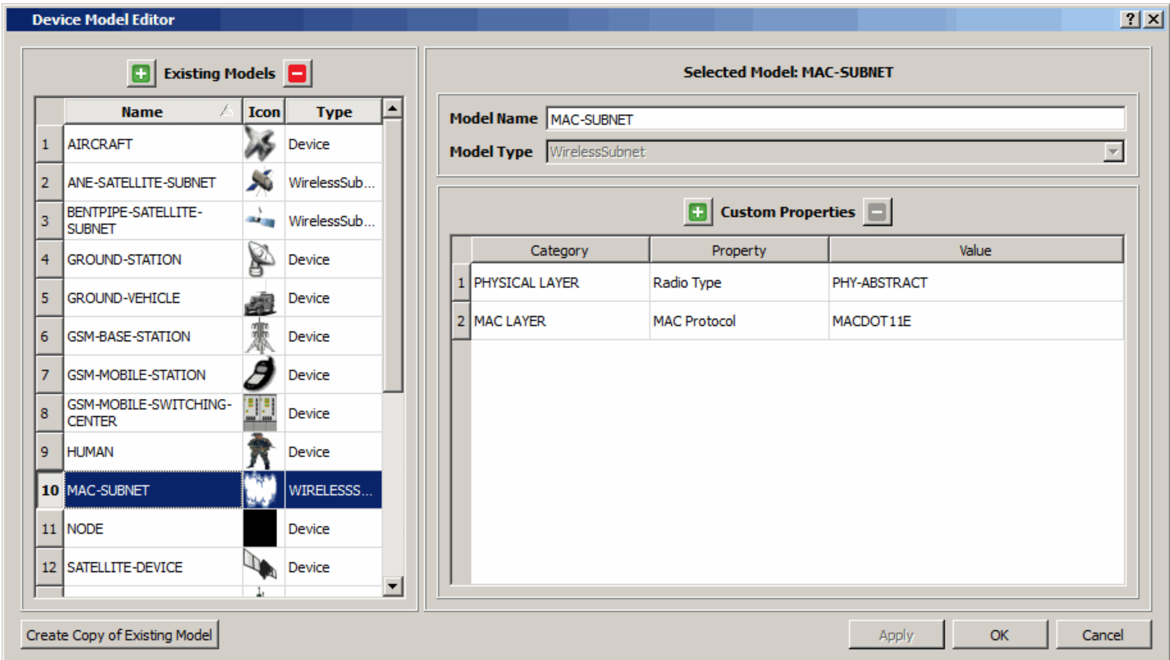


FIGURE 3-72. Properties of Custom Wireless Subnet Model

7. To change the value of a modified property, double-click on the property in the right panel. This opens the **Device Model Property Editor** in which a new value can be assigned to the property.
8. To remove a property from the table, select the property and click the button in the right panel.

Note: If a modified property is deleted from the table, the default value is used for that property.

9. To remove a model from the list, click the button in the left panel.
10. Click **Apply** or **OK** to save your changes.

The new model can be made available for use in scenarios by associating a button with it in the toolset (see Section 3.5.3). The new model can also be used in creating hierarchy models (see Section 3.5.2).

Modifying and Deleting Custom Network Object Models

To modify or delete a custom network object model, perform the following steps:

1. Open the **Device Model Editor**.
2. To modify a network object model, select the model in the left panel. In the right panel, add a new property, delete a property, or modify a property as described in the previous section.
3. To delete a network object model, select the model in the left panel and click the button in the left panel.

3.5.2 Creating Custom Hierarchy Models

To create a custom hierarchy model, perform the following steps.

1. Select **Tools > Hierarchy Model Editor**. This opens the **Hierarchy Model Editor** (see Figure 3-73).
2. The left panel displays the existing hierarchy models.
   a. To change the properties of an existing model, select it from the list.
   b. To create a new hierarchy model, click the button and enter the name in the **Name** field.

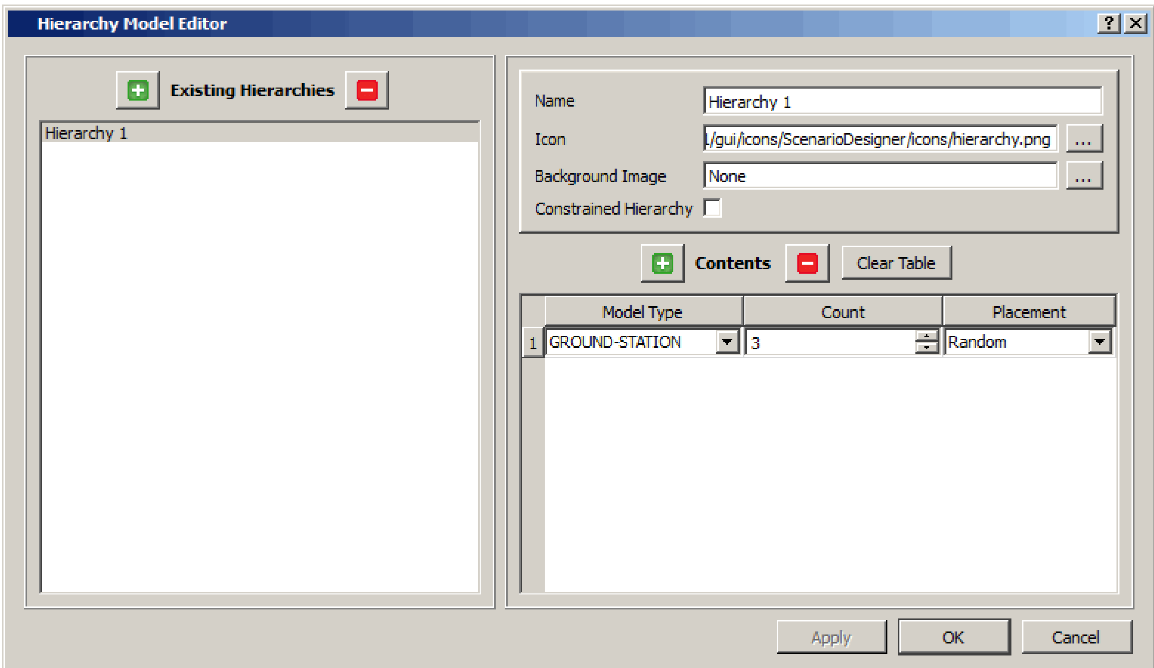


FIGURE 3-73. Hierarchy Model Editor

3. In the **Icon** field, enter the name (including the full path) of the icon file to be associated the hierarchy model. You can also open a file selection window by pressing the ... button and select the icon file by navigating to it.
4. Similarly, in the **Background File** field, specify the name of the image file to be used as the background of the hierarchy.
5. Check the **Constrained Hierarchy** box if you wish to make a constrained hierarchy.
6. Specify the components of the hierarchy in the components table. Each row in the table specifies a device type, the number of devices of the type, and the placement policy used for them.
   a. Add a row in the component table by clicking the button.
   b. Select the type of device from the pull-down list in the **Device Type** column. Enter the number of devices of that type in the **Count** column. Select the placement policy for these devices from the pull-down list in the **Placement** column.
   c. Add as many devices as desired by adding rows to the components table.
7. To remove a device from the table, select it and click the button.

8. To remove a hierarchy model from the list, select the model name in the left panel and click the button.
9. Click **Apply** or **OK** to save your changes.

The new hierarchy model can be made available for use in scenarios by associating a button with it in the toolset (see Section 3.5.3).

Figure 3-74 shows the Hierarchy Model Editor that creates a new hierarchy (called Sample-Hierarchy) with 10 aircraft, two ground stations, and 20 humans.

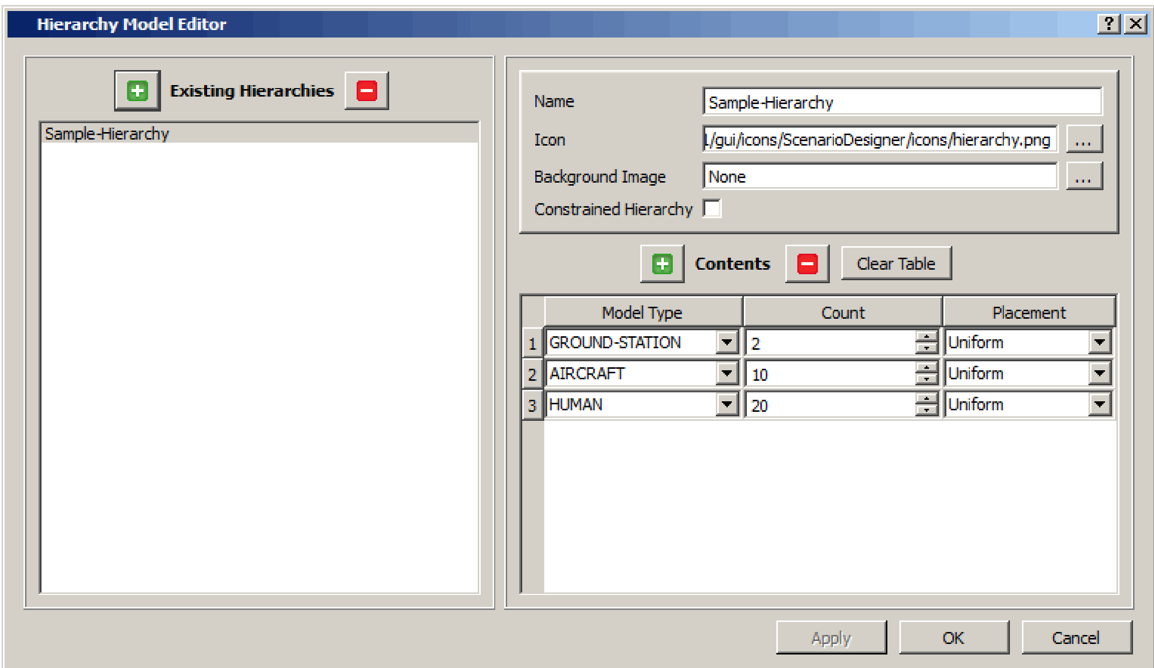


FIGURE 3-74. Sample Hierarchy

3.5.3 Creating and Customizing Toolsets

Architect has a predefined Standard toolset which can be customized. In addition, you can create new toolsets.

The **Standard** toolset can be modified in the following ways:

- New toolbars can be added to the toolset.
- New buttons can be added to any of the toolbars.
- Any toolbar or any button on any toolbar can be removed.

Note: Although buttons and toolbars can be removed from the **Standard** toolset, it is strongly recommended that users do not modify the **Standard** toolset other than adding buttons and toolbars.

To create a new toolset or customize an existing toolset, perform the following steps.

1. Select **Tools > Toolset Editor**. This opens the **Toolset Editor** (see Figure 3-75).
   The **Toolset Editor** has two tabs, **Toolset List** and **Customize Toolbar**.

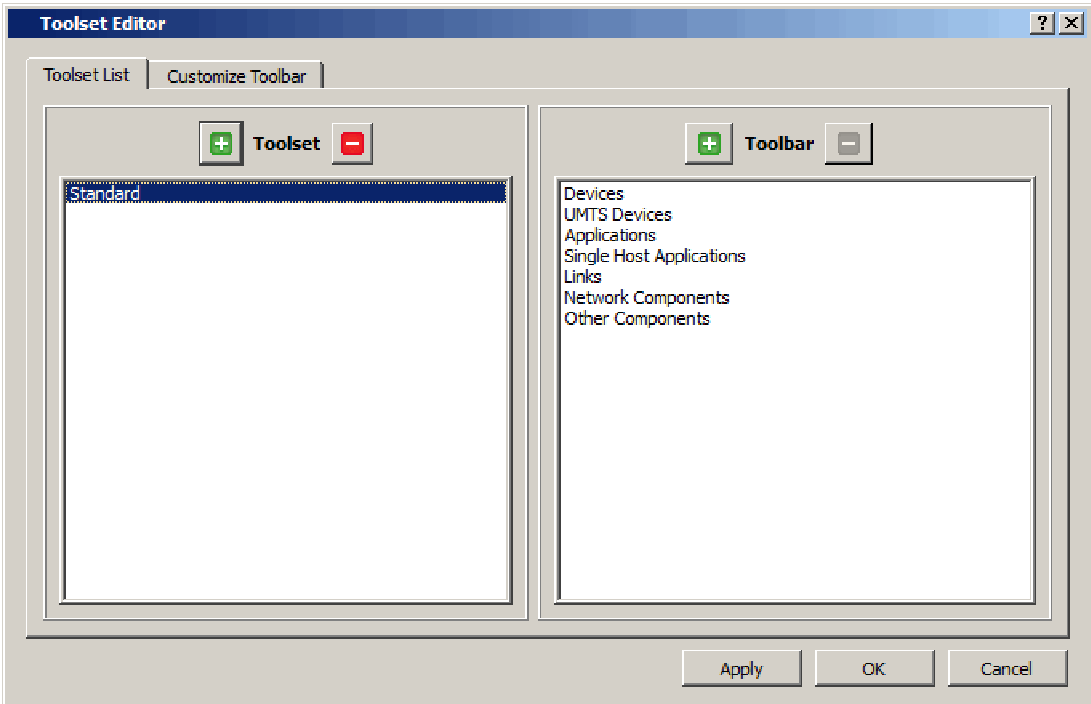


FIGURE 3-75. Toolset Editor: Toolset List Tab

2. Open the **Toolset List** tab. The left panel displays the list of available toolsets and the right panel displays the list of toolbars in the toolset selected in the left panel.

   a. To add a toolset, click the button above the toolset (left) panel. This adds a toolset to the list with the default name Toolset0, Toolset1, etc.

   Note: The new toolset is a copy of the **Standard** toolset and has all the toolbars of the Standard toolset (Devices, Applications, Single Host Applications, Links, Network Components, and Other Components). The new toolset can be modified as described in the following steps.

   b. To change the name of a toolset, either double-click on it or right click on it and select **Rename**, and type the new name.

   c. To delete a toolset from the list, select it in the list and click the button above the toolset panel.

   d. To add a toolbar to a toolset, select the toolset in the toolset panel. The toolbars in the selected toolset are displayed in the toolbar (right) panel. Click the button above the toolbar panel to add a new toolbar. This adds a toolbar to the list with the default name Toolbar0, Toolbar1, etc.

   e. To change the name of a toolbar, either double-click on it or right click on it and select **Rename**, and type the new name.

   f. To delete a toolbar from a toolset, select it in the list and click the button above the toolbar panel.

3. To customize a toolbar, open the **Customize Toolbar** tab. The left panel displays tool categories and the tools (buttons) in each category which are available for placing in toolbars. The right panel displays toolbars, and for each toolbar the tools that are currently in the toolbar.

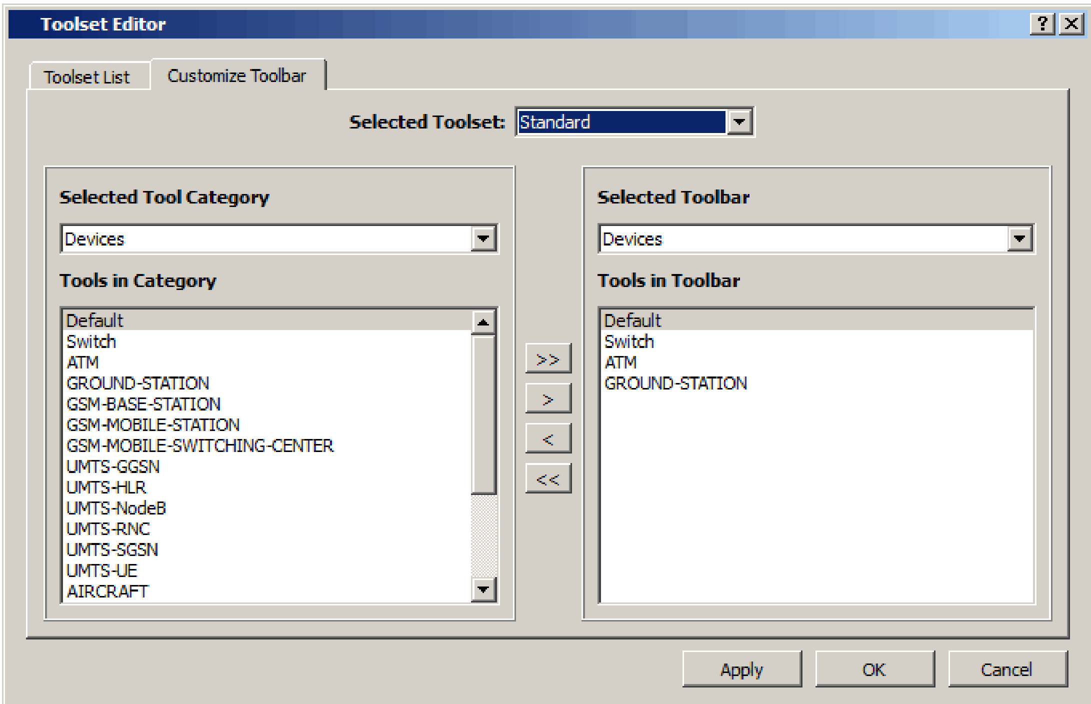


FIGURE 3-76. Toolset Editor: Customize Toolbar Tab

a. From the drop-down list at the top, select the toolset that you want to customize (e.g., Standard).

b. In the toolbar (right) panel, select the toolbar to customize from the drop-down list. The tools in the selected toolbar are displayed below it.

c. In the tool category (left) panel, select a tool category from the drop-down list. The following tool categories are available: Devices, Applications, Single Host Applications, Links, Network Components, and Other Components. The tools in the selected category are displayed below it.

Note: Each tool category contains a predefined list of items. You can add custom tools to the categories by using the **Device Model Editor** (see Section 3.5.1) and the **Hierarchy Model Editor** (see Section 3.5.2). A custom model created by using the **Device Model Editor** with Model Type Device, WiredSubnet, Satellite, or Switch is added to the Devices category, while a model with Model Type WirelessSubnet is added to the Network Components category. A custom hierarchy created using the **Hierarchy Model Editor** is added to the Network Components category.

d. To add a tool to the selected toolbar, select a tool in the left panel and click the > button. To add all tools in the tool category to the toolbar, click the >> button.

e. To remove a tool from a toolbar, select the tool in the right panel and click the < button. To remove all tools from the toolbar, click the << button. Figure 3-77 shows the customize toolbar for a standard toolset for network components.

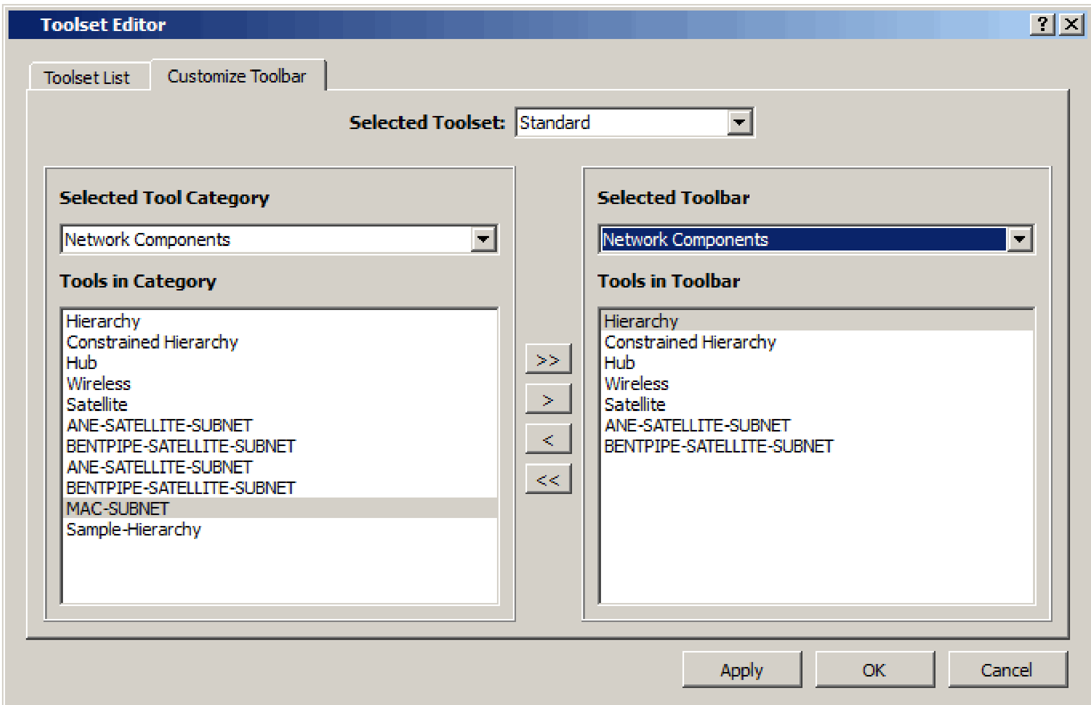


FIGURE 3-77. Toolset Editor

Figure 3-78 shows the results of creating a custom network object model (MAC-Subnet) using the **Device Model Editor** (Section 3.5.1), creating a custom hierarchy model (Sample-Hierarchy) using the **Hierarchy Model Editor** (Section 3.5.2), then adding them to the **Network Components** toolbar of the **Standard**

Toolset using the **Toolset Editor**. A hierarchy of the type Sample-Hierarchy has been placed on the canvas and opened to display its components.

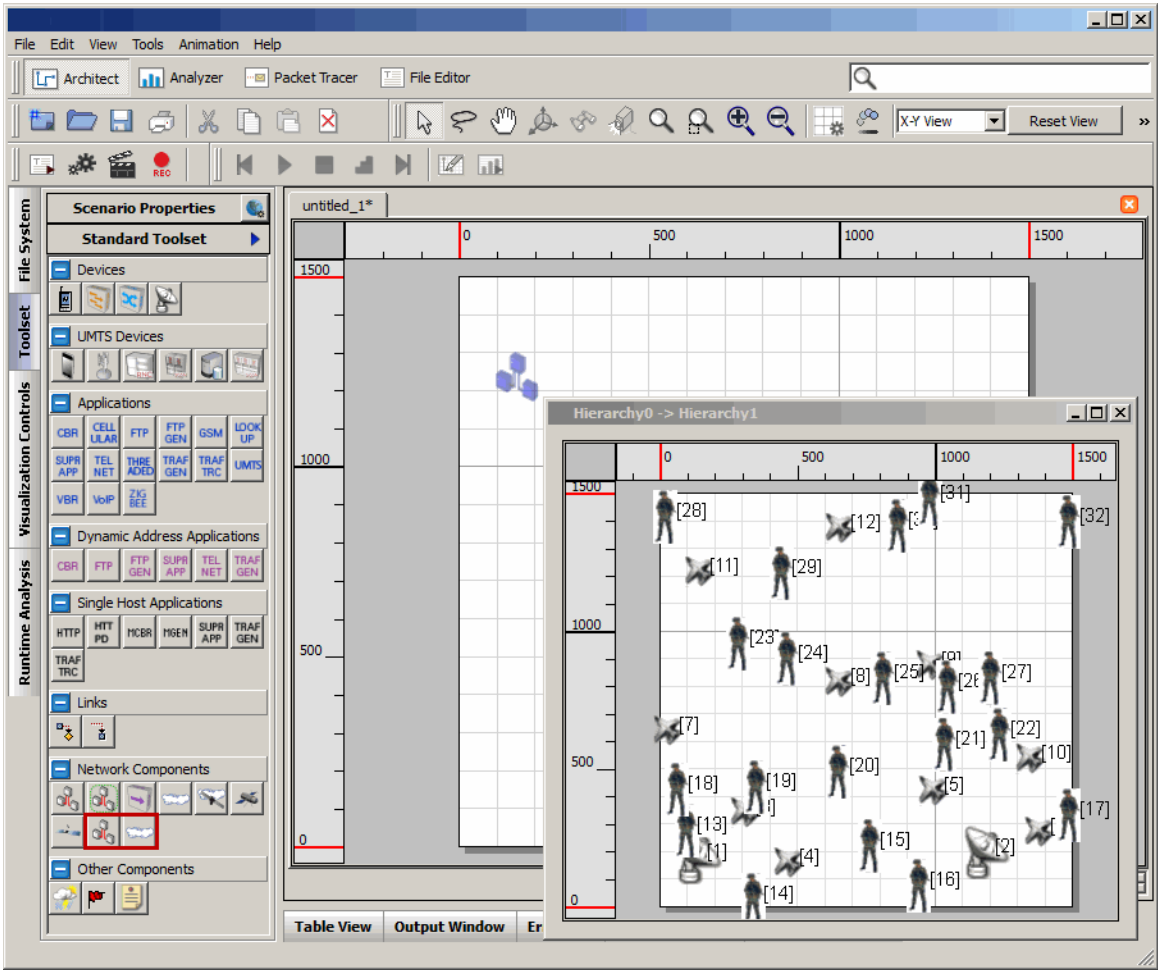


FIGURE 3-78. Customized Toolset Example

4 Modeling Networks

This chapter describes how to create networks models in EXata (in both command line and GUI) and how to use EXata for emulation.

Section 4.1 describes conventions used in this chapter.

Section 4.2 gives a detailed description of configuring network scenarios. It describes how to configure high-level parameters in command line and in Architect. References to model libraries are provided to configure model-dependent parameters.

Section 4.3 gives details of configuring the emulation environment in EXata.

Section 4.4 describes the parameters for configuring EXata for a multi-core/multi-processor environment.

Section 4.5 describes the parameters for achieving trade-off between simulation speed and accuracy of results.

Section 4.6 gives an overview of advanced network modeling features. These features are described in detail in the model libraries.

4.1 Conventions Used

4.1.1 Conventions for Command Line Configuration

In this document, most parameters are described using a tabular format described below. The parameter description tables have three columns labeled “Parameter”, “Value”, and “Description”. Table 4-1 shows the format of parameter tables. Table 4-3 shows examples of parameter descriptions in this format.

TABLE 4-1. Parameter Table Format

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

<u>*Parameter Column*</u>

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).

- ***<Dependency>:*** This entry specifies the condition for the parameter to be included in the input file. Usually, the condition is some other parameter being set to a certain value). If the only condition for including the parameter is to select the model, then this entry is omitted.

  Examples of dependencies are:

  Dependency: `MAC-DOT-11-ASSOCIATION = DYNAMIC`

  Dependency: `ANTENNA-MODEL-TYPE ≠ OMNIDIRECTIONAL`

- ***<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.

Note: The parameter designation is relative to the dependency. For example, if a parameter must be included if some condition is true, then the condition is listed in the *<Dependency>* field and the parameter designation is set to *Required*.

- ***<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 4-2 shows the values a parameter can take for each type.

TABLE 4-2. Parameter Types

Type	Description
Integer	Integer value Examples: `2`, `10`
Real	Real value Examples: `15.0`, `-23.5`, `2.0e8`
String	String value Examples: `TEST`, `SWITCH1`
Time	Time value expressed in EXata time syntax (see Section 2.2.2) Examples: `1.5S`, `200MS`, `10US`

Filename	Name of a file in EXata filename syntax (see Section 2.2.6)
	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)

TABLE 4-2. Parameter Types (Continued)

Type	Description
Path	Path to a directory in EXata path syntax (see Section 2.2.6) 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`, `2000:0:0:1::1`
IPv4 Address	IPv4 address Examples: `192.168.2.1`
IPv6 Address	IPv6 address Examples: `2000:0:0:1::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 4-3.

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 4-3 shows examples of parameter descriptions using the format described above.

TABLE 4-3. 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 4-30 for a description of mobility models.
`BACKOFF-LIMIT` *Dependency:* `USE-BACKOFF = YES` *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.

TABLE 4-3. Example Parameter Table (Continued)

Parameter	Values	Description
`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.

4.1.2 Conventions 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 4-1.

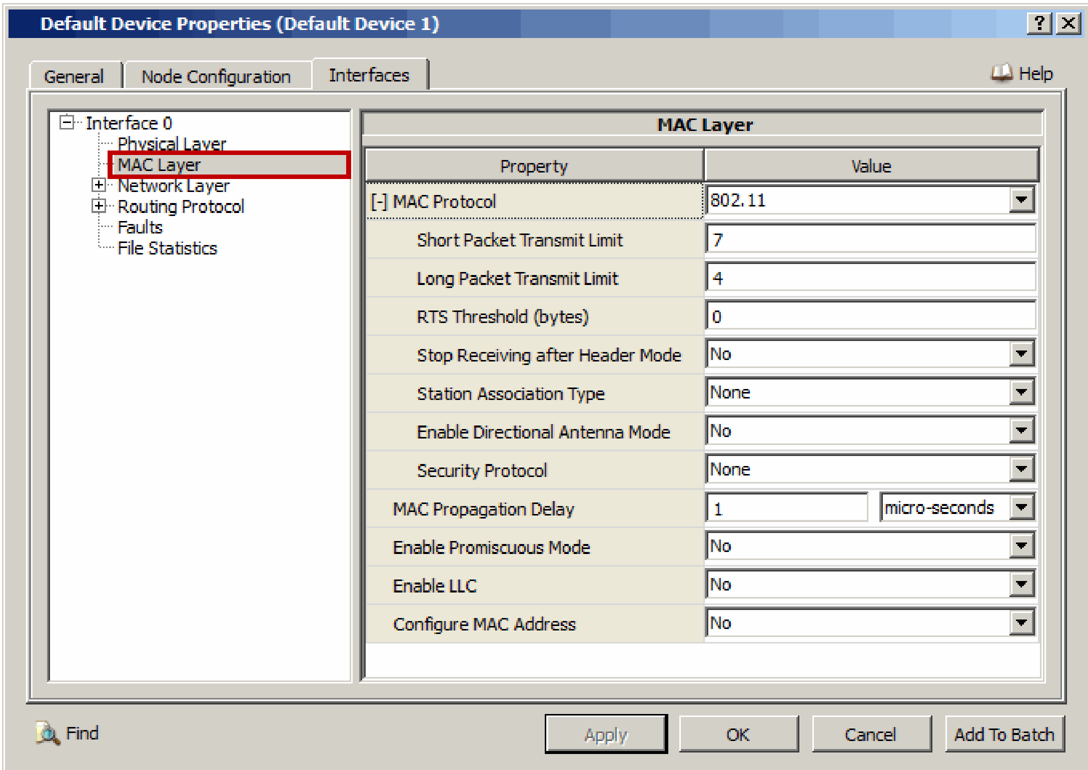


FIGURE 4-1. Path to a Parameter Group

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 as 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 as Access Point** to *Yes*.
4. Set **Enable Power Save Mode** to *Yes*.

The above path is shown in Figure 4-2.

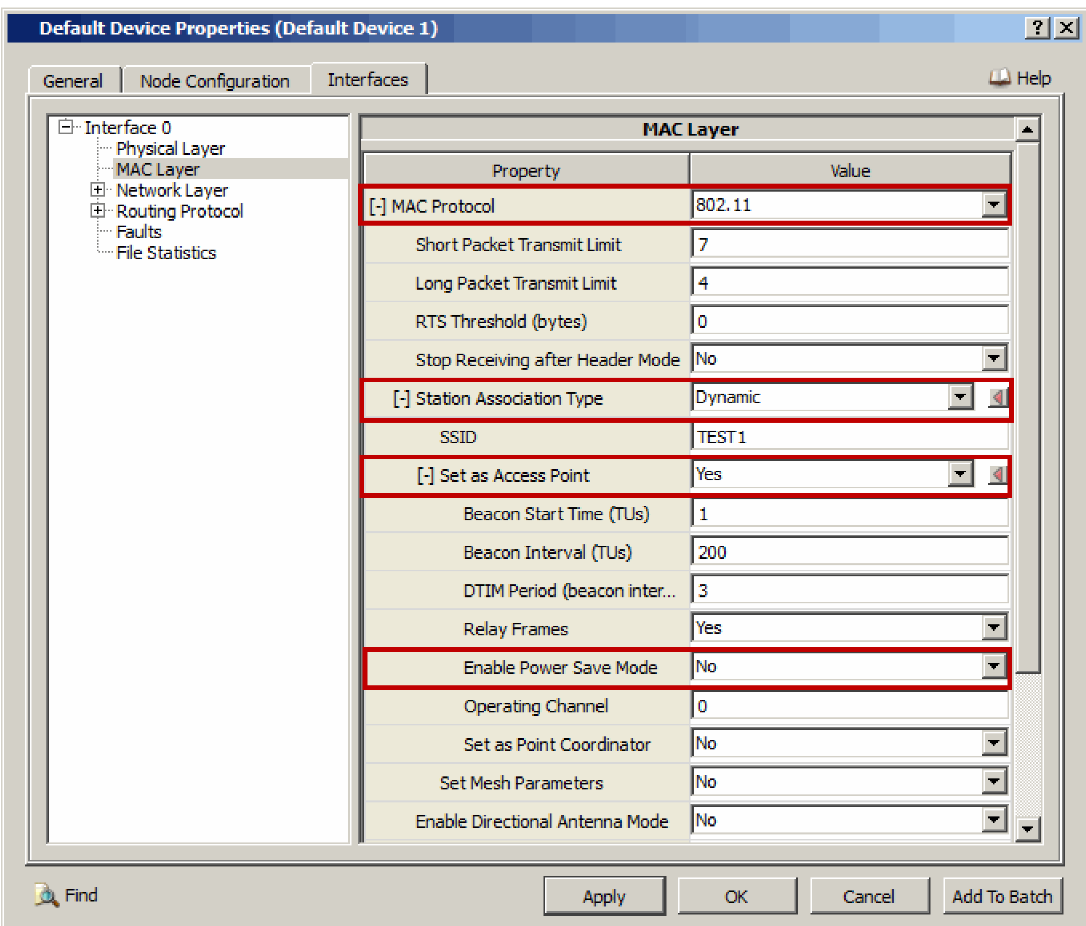


FIGURE 4-2. Path to a Specific Parameter

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 more information on a GUI parameter, see the description of the equivalent command line parameter in the command line configuration section. The mapping table also indicates the levels at which the parameter can be configured in the GUI.

The format of a parameter mapping table is shown in Table 4-4

TABLE 4-4. Example Mapping Table

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

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 4-5 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 4-5.

TABLE 4-5. 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 4-6 is an example of a parameter mapping table.

TABLE 4-6. Example Mapping Table

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Short Packet Transmit Limit	Subnet, Interface	`MAC-DOT11-SHORT-PACKET-TRANSMIT-LIMIT`
Long Packet Transmit Limit	Subnet, Interface	`MAC-DOT11-SHORT-PACKET-TRANSMIT-LIMIT`
RTS Threshold	Subnet, Interface	`MAC-DOT11-RTS-THRESHOLD`

4.2 Configuring Scenarios

Several input files together constitute a scenario description (see Section 2.1.1.1). The primary input files are: scenario configuration (.config) file, initial node position (.nodes) file, and application configuration (.app) file. Other input files may be required by some scenarios. This section describes how to create input files for a typical scenario.

Notes:

1. Only the high-level parameters are described in this chapter. Model-specific parameters are described in the model libraries. To complete a scenario configuration, you must include the model-specific parameters in the input file(s), and, in some cases, create supplemental input files. For example, you can select the routing protocol to be Open Shortest Path First (OSPF) version 2 by setting the parameter `ROUTING-PROTOCOL` to `OSPFv2` in the scenario configuration (.config) file (see Section 4.2.8.3.2). To complete the scenario configuration, you will need to include OSPFv2-specific parameters in the scenario configuration file. You may also need to create an OSPFv2 configuration file. These OSPFv2-specific parameters and the format of the OSPFv2 configuration file are described in *Multimedia and Enterprise Model Library*.
2. Unless otherwise stated, parameters described in this section should be included in the scenario configuration (.config) file.
3. A parameter declaration should be on a single line by itself, using the format described in Section 2.2.9.1.
4. Comments can be entered anywhere in the input files (see Section 2.2.1).
5. In the scenario configuration file, parameters can be entered in any order.

It may be convenient to start with input files for one of the scenarios distributed with EXata and modify them for your scenario. Make copies of the supplied input files and work on the copies. Keep the original input files for reference. For example, to start with the sample scenario in the folder EXATA_HOME/scenarios/default, do the following:

1. Copy default.config to new.config.
2. Copy default.app to new.app.
3. Copy default.nodes to new.nodes
4. In new.config, change the value of the parameter `APP-CONFIG-FILE` to new.app.
5. In new.config, change the value of the parameter `NODE-POSITION-FILE` to new.nodes.
6. Make modifications to new.config, new.nodes and new.app, as needed.

The sample scenario configuration file, default.config, contains descriptive information about the parameters that can be configured. It also contains several available options for a large number of parameters, all of which are commented, except for the parameter currently selected. In order to select a different option than the one already selected, add a "#" symbol in front of the current selection, and remove it from the option to be used instead.

This section gives a brief overview of how to develop a simulation scenario for the Command Line Interface. Details are provided in later sections. Only high-level scenario parameters are described in this chapter. Configuration parameters required for specific scenario components, such as protocols, are described in the model libraries. Note that the following is not a comprehensive list of steps involved in developing a scenario: some scenarios may need additional steps to configure. Section 4.6 gives an overview of some advanced features in network modeling which are described in detail in the model libraries.

4.2.1 General Parameters

This set of parameters define the general attributes of the experiment, such as experiment name, length of simulation, and seed for random number generation.

4.2.1.1 Command Line Configuration

To configure general simulation parameters for the command line interface, include the parameters listed in Table 4-7 in the scenario configuration (.config) file..

TABLE 4-7. General Parameters

Parameter	Value	Description
`EXPERIMENT-NAME` *Optional* *Scope:* Global	String *Default:* `exata`	Name of the experiment. Names of output files (.stat and.trace files) are based on the experiment name unless the command-line parameter is specified (see Section 2.1.1.2).
`EXPERIMENT-COMMENT` *Optional* *Scope:* Global	String	Comment used as an experiment annotation. This comment is reproduced at the beginning of the trace file generated by running the experiment, and can be used to distinguish trace files from different runs.
`SIMULATION-TIME` *Required* *Scope:* Global	Time *Range:* `> 0S`	Length of simulation.
`SEED` *Required* *Scope:* Global	Integer *Range:* `> 0`	Seed used to generate random number streams.
`GUI-CONFIG-LOCKED` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `NO`	Indicates whether changes can be made to the scenario when it is opened using the GUI. If this parameter is set to `YES`, then the scenario can be opened and run using the GUI, but any changes made to the scenario in Architect's Design Mode can not be saved. (If the scenario is modified, it can be saved under a different name.)

4.2.1.2 GUI Configuration

To set the general simulation parameters in the GUI, do the following:

1. Go to **Scenario Properties Editor > General > General Settings**.
2. Set the parameters listed in Table 4-8.

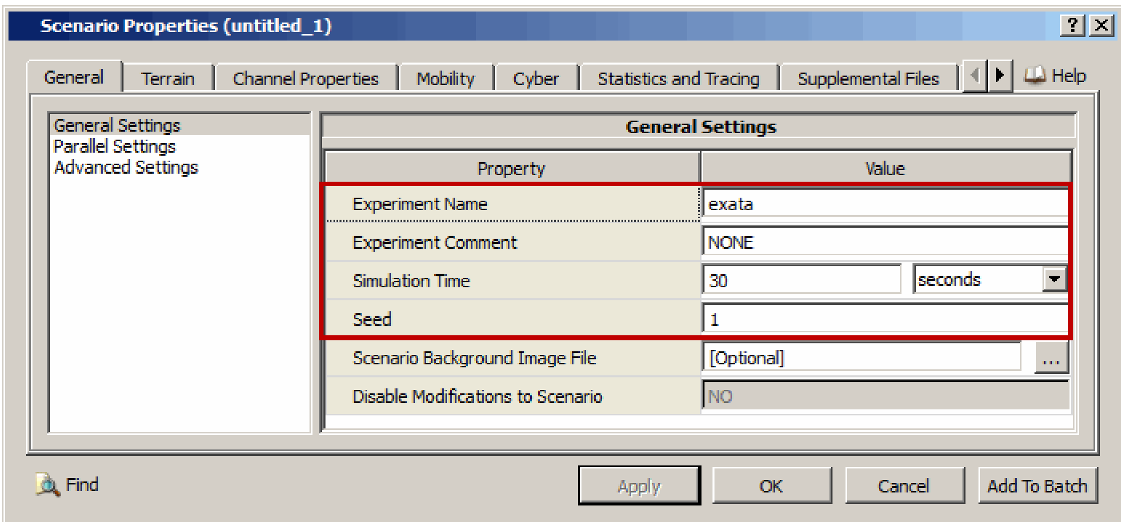


FIGURE 4-3. Setting General Simulation Parameters

TABLE 4-8. Command Line Equivalent of General Simulation Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Experiment Name	Global	`EXPERIMENT-NAME`
Experiment Comment	Global	`EXPERIMENT-COMMENT`
Simulation Time	Global	`SIMULATION-TIME`
Seed	Global	`SEED`

3. To display a background image on the canvas, set **Scenario Background Image File** to the name of image file.

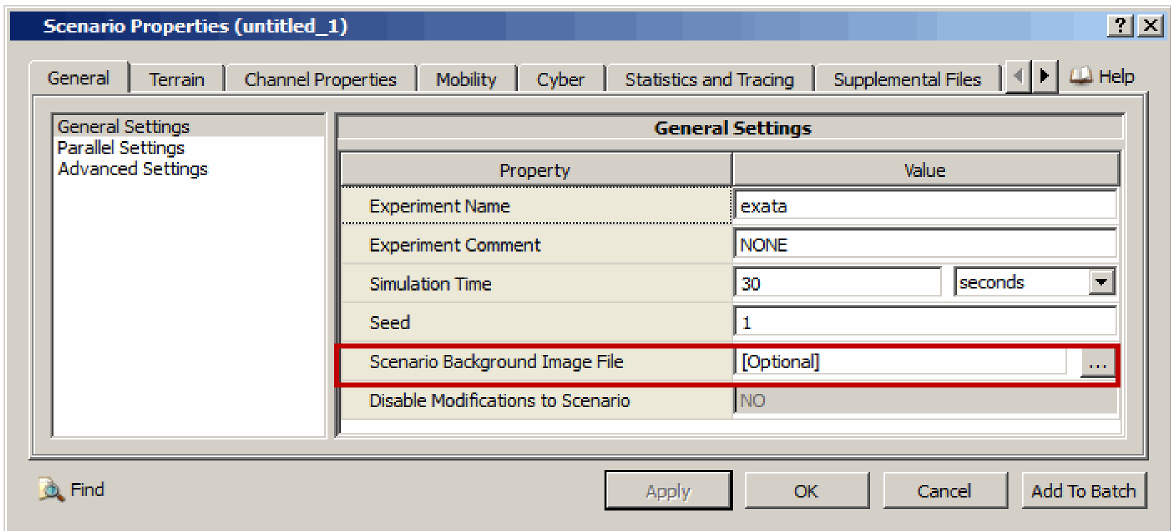


FIGURE 4-4. Setting Background Image

4.2.2 Terrain Specification

This set of parameters defines the coordinate system to be used, the terrain size to simulate, and the terrain features. Terrain features, such as ground elevation at different points and the dimensions of buildings, affect the strength of signals transmitted by nodes. To accurately model the extent of signal attenuation, EXata makes use of the terrain specification.

4.2.2.1 Command Line Configuration

To configure terrain properties for the command line interface, include the parameters listed in Table 4-9 in the scenario configuration (.config) file.

TABLE 4-9. Terrain Parameters

Parameter	Value	Description
`COORDINATE-SYSTEM` *Optional* *Scope:* Global	List: • `CARTESIAN` • `LATLONALT` *Default:* `CARTESIAN`	Coordinate system used in the scenario. `CARTESIAN` : Cartesian coordinate system is used. `LATLONALT` : Latitude-Longitude-Altitude system is used.
`TERRAIN-DIMENSIONS` *Dependency:* `COORDINATE-SYSTEM` = `CARTESIAN` *Required* *Scope:* Global	Pair of real numbers in the format (x, y) *Unit:* meters	Dimensions of the terrain. The x-dimension and y-dimension of the terrain (in meters) are specified as a pair of real numbers (> 0.0) separated by a comma and enclosed in parentheses. *Example*: `(1000, 1500)`

TABLE 4-9. Terrain Parameters (Continued)

Parameter	Value	Description
TERRAIN-SOUTH-WEST-CORNER *Dependency:* COORDINATE-SYSTEM = LATLONALT *Required* *Scope:* Global	Lat-Lon pair in the format (x, y) *Unit:* degrees	Coordinates of the south-west corner of the terrain. The latitude and longitude of the terrain (in degrees) are specified as a pair of real numbers (> 0.0) separated by a comma and enclosed in parentheses. *Example*: (34.99, -120.00)
TERRAIN-NORTH-EAST-CORNER *Dependency:* COORDINATE-SYSTEM = LATLONALT *Required* *Scope:* Global	Lat-Lon pair in the format (x, y) *Unit:* degrees	Coordinates of the north-east corner of the terrain. The latitude and longitude of the terrain (in degrees) are specified as a pair of real numbers (> 0.0) separated by a comma and enclosed in parentheses. *Example*: (35.0, -119.99)
TERRAIN-DATA-TYPE *Optional* *Scope:* Global	List: • NONE • CARTESIAN • DEM • DTED *Default:* NONE	Format used for terrain elevation data. If TERRAIN-DATA-TYPE is set to NONE, then terrain elevation data are not used in the simulation. See Table 4-10 for a description of terrain elevation data formats.
URBAN-TERRAIN-TYPE *Optional* *Scope:* Global	List: • NONE • QUALNET-URBAN-TERRAIN *Default:* NONE	Interface for urban terrain features (such as buildings and roads) data. If URBAN-TERRAIN-TYPE is set to NONE, then urban terrain features data are not used in the simulation. See Table 4-11 for a description of interfaces for urban terrain features data.
TERRAIN-FEATURES-SOURCE *Dependency:* URBAN-TERRAIN-TYPE = QUALNET-URBAN-TERRAIN *Optional* *Scope:* Global	List: • FILE • SHAPEFILE	Format used for urban terrain features (such as buildings and roads) data. If this parameter is not included in the scenario configuration (.config) file, urban terrain features data are not used in the simulation. See Table 4-12 for a description of formats used for urban terrain features data.

Table 4-10 describes the different formats for terrain elevation data. See the corresponding model library for the description of each format.

TABLE 4-10. Terrain Elevation Data Formats

Command Line Name	GUI Name	Description	Model Library
`CARTESIAN`	Cartesian	Cartesian terrain data type. The Cartesian terrain format is intended to provide terrain data for small areas specified in Cartesian coordinates.	Wireless
`DEM`	USGS DEM	Digital Elevation Model (DEM) data type. DEM files are produced by the USGS. EXata supports only the 1 degree files, corresponding to DTED level 1, with elevation points in a grid at approximately 100 meters spacing.	Wireless
`DTED`	DTED	Digital Terrain Elevation Data (DTED) data type. DTED data are available from a variety of sources and in various resolutions. All resolutions contain grids of elevation points. DTED level 0 is spaced at about 1000 meters per data point, DTED level 1 at 100 meters, DTED level 2 at 30 meters, DTED level 3 at 10 meters, DTED level 4 at 3 meters, and DTED level 5 at 1 meter.	Wireless

Table 4-11 describes the different interfaces for urban terrain features. See the corresponding model library for the description of each interface.

TABLE 4-11. Urban Terrain Features Interfaces

Command Line Name	GUI Name	Description	Model Library
`QUALNET-URBAN-TERRAIN`	QualNet Format	QualNet urban terrain interface. This interface takes as input urban features data in one of several data formats and converts it into QualNet Terrain Format during initialization.	Wireless

Table 4-12 describes the formats for urban terrain data. See the corresponding model library for the description of each interface.

TABLE 4-12. Urban Terrain Data Formats

Command Line Name	GUI Name	Description	Model Library
`FILE`	QualNet Terrain File	Qualnet terrain format. This is a proprietary XML format provided by Scalable Network Technologies to define urban terrain features, such as buildings, roads, parks, etc.	Wireless
`SHAPEFILE`	Shapefile	ESRI shapefile. ESRI shapefiles are a popular means of describing geospatial data.	Wireless

4.2.2.2 GUI Configuration

To set the terrain parameters in the GUI, do the following:

1. Go to **Scenario Properties Editor > Terrain**.
2. Set **Coordinate System** to *Cartesian* or *Latitude-Longitude*

Note: You can not change the coordinate system after placing objects on the canvas.

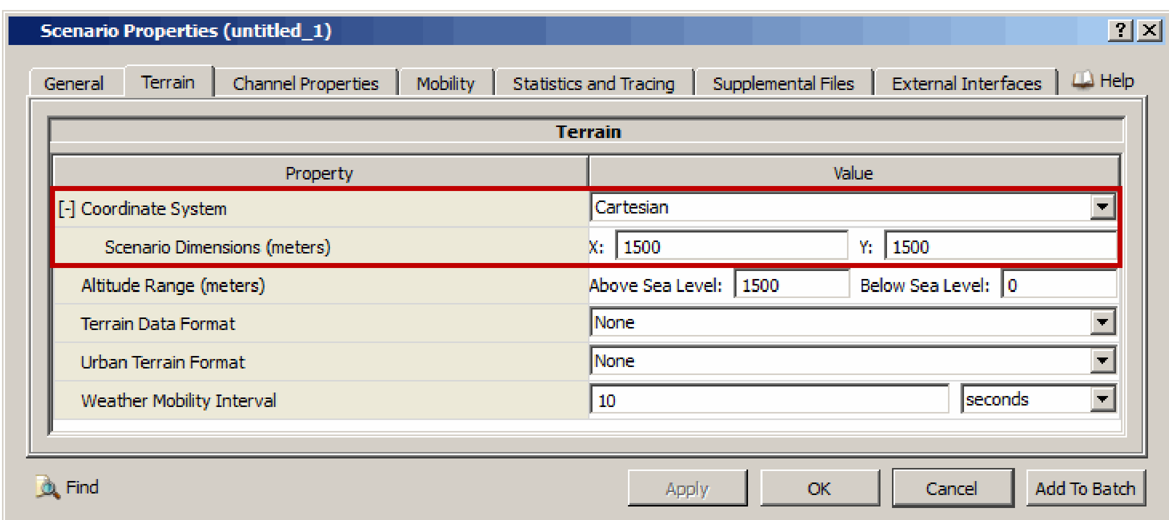


FIGURE 4-5. Setting Coordinate System to Cartesian

TABLE 4-13. Command Line Equivalent of Coordinate System Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Coordinate System	Global	`COORDINATE-SYSTEM`

a. If **Coordinate System** is set to *Cartesian*, then enter values for the *X* and *Y* components of the **Scenario Dimensions** parameter (see Figure 4-5).

TABLE 4-14. Command Line Equivalent of Cartesian Coordinate System Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Scenario Dimensions	Global	`TERRAIN-DIMENSIONS`

b. If **Coordinate-System** is set to *Latitude-Longitude*, then enter values for the *Lat* and *Lon* components (latitude and longitude, respectively) of the **SW Corner** and **NE Corner** parameters.

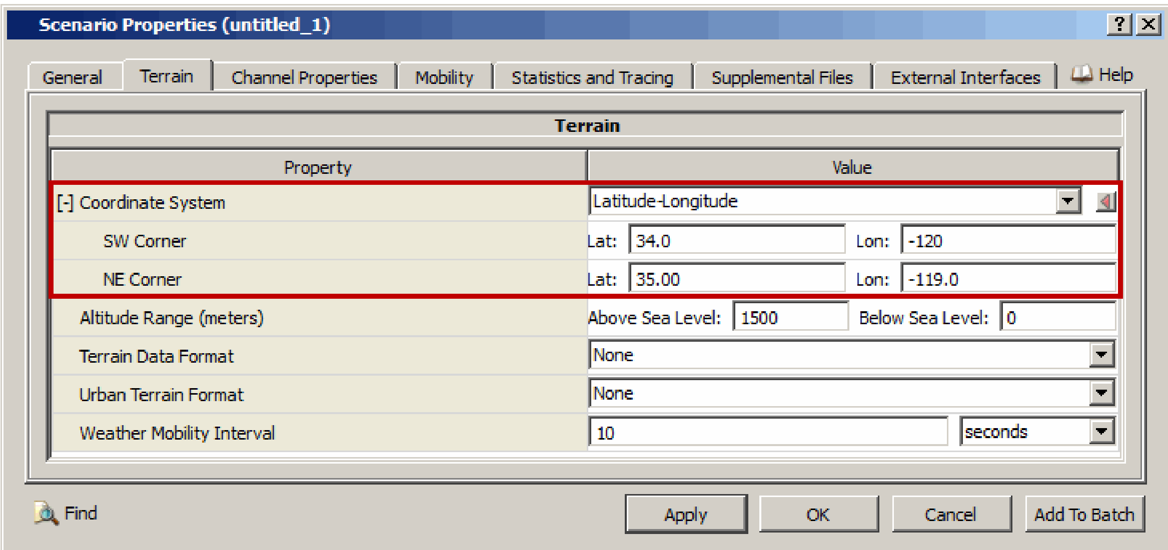


FIGURE 4-6. Setting Coordinate System to Latitude-Longitude

TABLE 4-15. Command Line Equivalent of Latitude-Longitude System Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
SW Corner	Global	`TERRAIN-SW-CORNER`
NE Corner	Global	`TERRAIN-NE-CORNER`

3. Set the *Above Sea Level* and *Below Sea Level* components of the parameter **Altitude Range**. This parameter specifies the range of altitudes that can be displayed in the GUI.

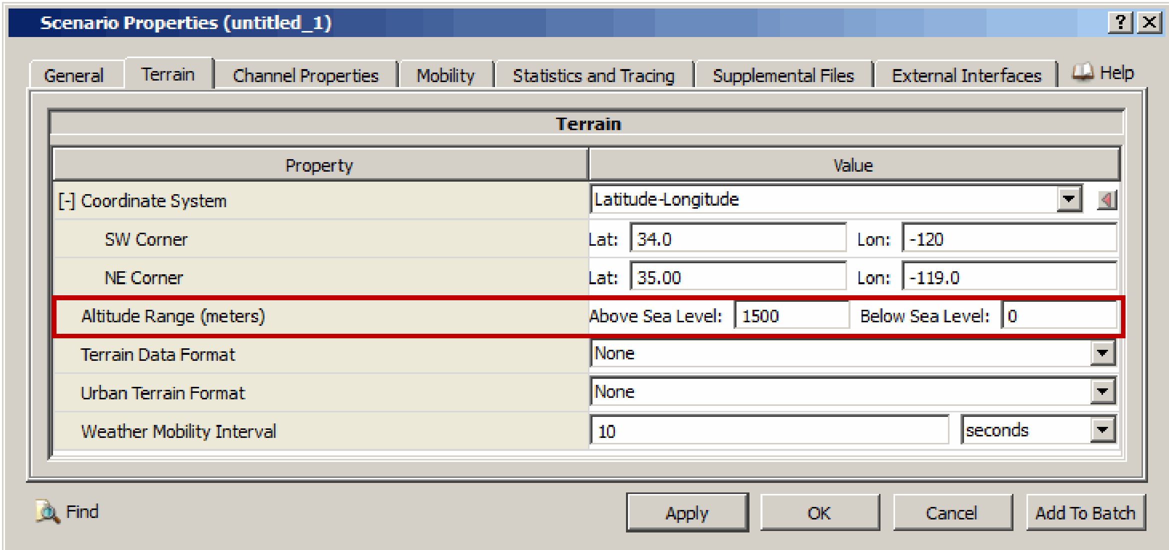


FIGURE 4-7. Setting Altitude Range Parameters

Note: Parameter **Altitude Range** does not have an equivalent in the command line interface.

4. Set the terrain elevation data parameters as follows.

   a. Select the terrain elevation data format by setting **Terrain Data Format** to the appropriate value. The available terrain elevation data formats are described in Table 4-10.

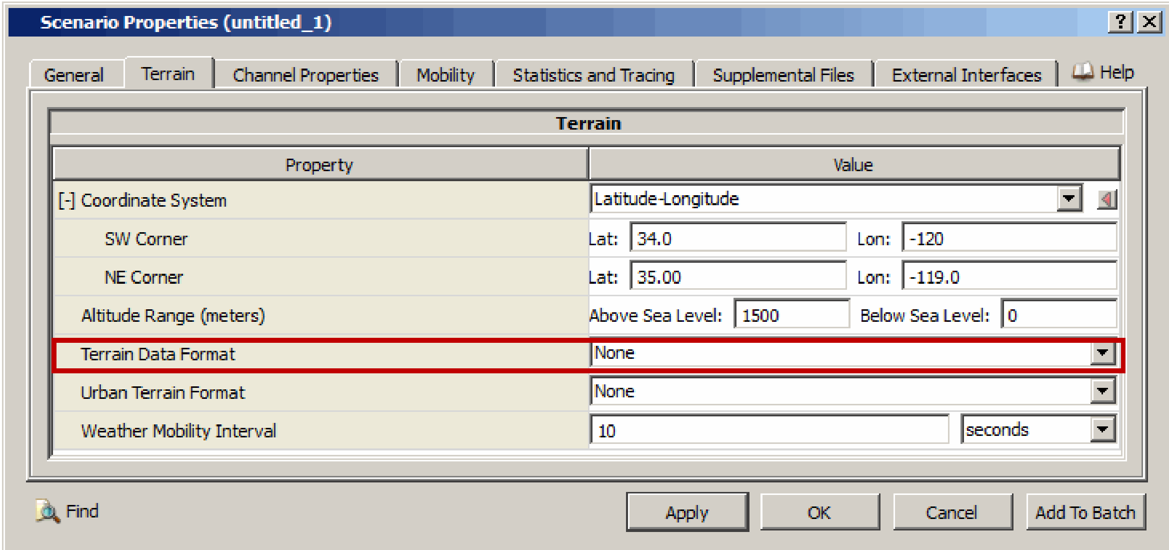


FIGURE 4-8. Selecting Terrain Elevation Data Format

TABLE 4-16. Command Line Equivalent of Terrain Elevation Data Format Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Terrain Data Format	Global	`TERRAIN-DATA-TYPE`

b. Set the dependent parameters for the selected terrain data format. See the model library referenced in Table 4-10 for details.

5. Set the urban terrain features parameters as follows.

a. Select the urban terrain features interface by setting **Urban Terrain Format** to the appropriate value. The available urban terrain features interfaces are described in Table 4-11.

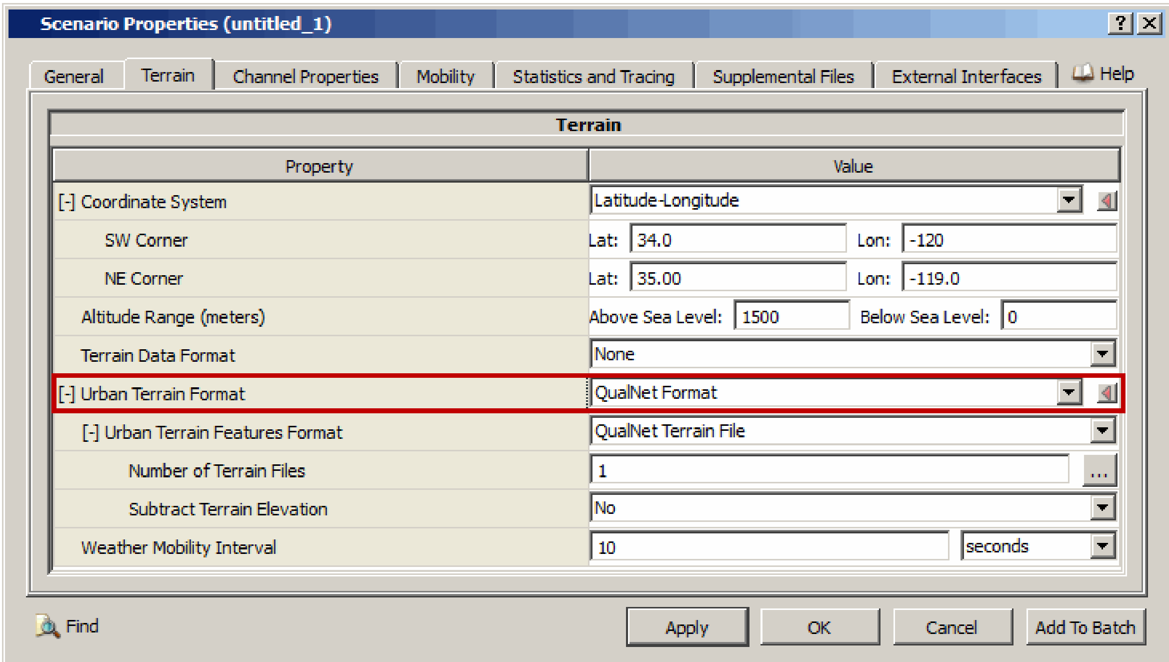


FIGURE 4-9. Selecting Urban Terrain Features Interface

TABLE 4-17. Command Line Equivalent of Urban Terrain Features Interface Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Urban Terrain Format	Global	`URBAN-TERRAIN-TYPE`

b. If **Urban Terrain Format** is set to *QualNet Format*, then select urban terrain data format by setting **Urban Terrain Features Format** to the appropriate value. The available urban terrain data formats are described in Table 4-12.

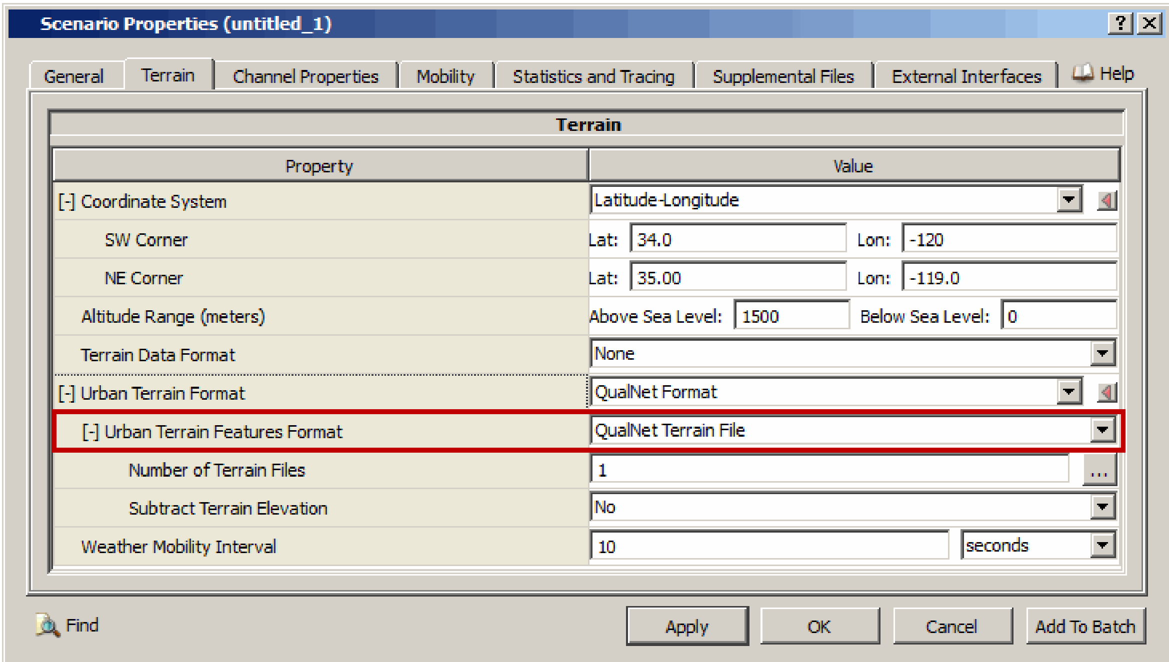


FIGURE 4-10. Setting Urban Terrain Features Interface Parameters

TABLE 4-18. Command Line Equivalent of Urban Terrain Features Interface Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Urban Terrain Features Format	Global	TERRAIN-FEATURES-SOURCE

c. Set the dependent parameters for the selected urban terrain features format. See the model library referenced in Table 4-12 for details.

4.2.3 Node Placement

This section describes the parameters and models that determine the initial positions of nodes.

4.2.3.1 Command Line Configuration

In command line, nodes are created as part of the topology specification (see Section 4.2.5).

To configure node placement parameters for the command line interface, include the parameters listed in Table 4-19 in the scenario configuration (.config) file.

TABLE 4-19. Node Placement Parameters

Parameter	Value	Description
NODE-PLACEMENT *Required* *Scope:* Global, Node	List: • FILE • GRID • GROUP • RANDOM • UNIFORM	Specification of policy to place nodes in a scenario. See Table 4-20 for a description of node placement models.
MOBILITY-GROUND-NODE *Dependency:* TERRAIN-DATA-TYPE ≠ NONE *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* NO	Indication whether the node's altitude is read from a terrain file. This parameter is used only if a terrain file is also specified in the scenario (see Section 4.2.2). YES : At each point along the node's path as it moves on the terrain, the node's z-coordinate (for Cartesian system) or altitude (for Lat-Lon-Alt system) is the same as the altitude specified in the terrain file for that point. The value read from the terrain file overrides the z-coordinate or altitude determined by the mobility model. NO : The node's z-coordinate or altitude is determined by the mobility model.

Table 4-20 describes the different node placement models in EXata. See the corresponding model library for the description of each model and its parameters.

TABLE 4-20. Node Placement Models

Command Line Name	GUI Name	Description	Model Library
FILE	File	File-based node placement policy. The initial node positions are read from a file.	Wireless
GRID	Grid	Grid node placement policy. The terrain is divided into a number of squares. One node is placed at each grid point.	Wireless
GROUP	N/A	Group-based node placement policy. This node placement model is used with the Group mobility model (see Section 4.2.6).	Wireless

TABLE 4-20. Node Placement Models (Continued)

Command Line Name	GUI Name	Description	Model Library
`RANDOM`	Random	Random node placement policy. Nodes are placed on the terrain randomly.	Wireless
`UNIFORM`	Uniform	Uniform node placement policy. The terrain is divided into a number of equal-sized square cells. One node is placed in each cell randomly.	Wireless

4.2.3.2 GUI Configuration

In the GUI, nodes can be placed manually (see Section 4.2.3.2.1) or automatically by using the **Node Placement Wizard** (see Section 4.2.3.2.2). When the **Node Placement Wizard** is used, File, Random, Uniform, or Grid placement strategy can be specified. If the File placement strategy is specified, the initial node positions are read from a file. If the Random, Uniform, or Grid placement strategy is specified, Architect places nodes on the canvas using a strategy similar to the Random, Uniform, and Grid placement strategies in command line configuration (see Table 4-20). A combination of placement strategies can be used to place different groups of nodes. Once nodes have been placed on the canvas, either manually or automatically, Architect creates a node position file which contains the initial positions of all nodes. When the scenario is run, the simulator uses the File node placement model (see Table 4-20) with the node position file created by Architect.

4.2.3.2.1 Placing Nodes Manually

To manually place a device, click on the desired device in the **Devices** toolbar. Then click on the canvas to place the device. You can move a device to another location on the canvas by selecting and moving it to the desired location.

In addition, you can move a device to a specific position by selecting the device on the canvas, and entering the desired coordinates in the **Position** Indicators.

4.2.3.2.2 Using Node Placement Wizard

To place nodes through the **Node Placement Wizard**, do the following:

1. Select **Tools > Node Placement**. This opens the **Node Placement Wizard** shown in Figure 4-11.

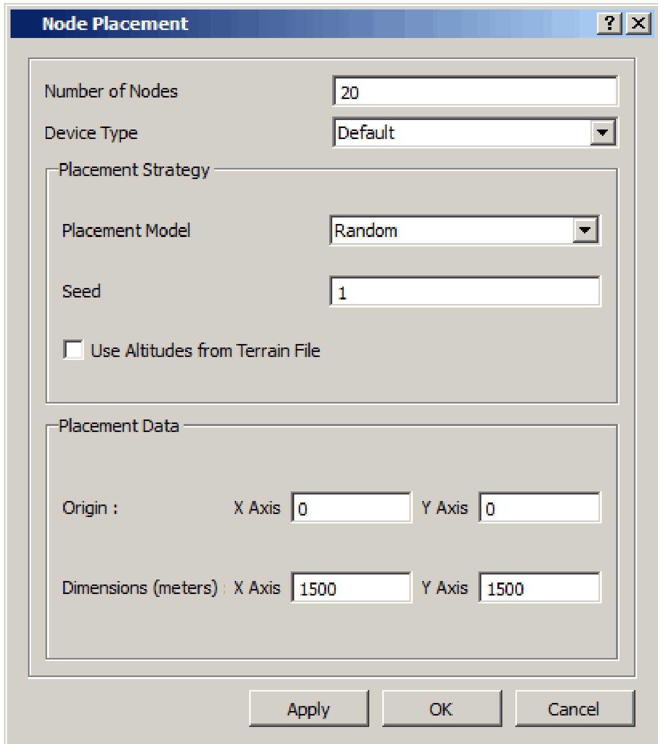


FIGURE 4-11. Node Placement Wizard

2. In the **Number of Nodes** field, enter the desired number of nodes.
3. In the **Device Type** field, select the type of device from the list. All devices that appear in the **Devices** toolbar in the Toolset also appear in the list.
4. *Under* **Placement Strategy***, select the* **Placement Model** *from the list. See* Table 4-20 for a description of the placement models.
   a. If **Placement Model** is *File*, then specify the name of the position file in the field **Node Placement File to Import**.
   b. If **Placement Model** is *Grid*, then specify the grid unit in the **Grid Unit** field.
   c. If **Placement Model** is *Random* or *Uniform*, then specify a value for the seed in the **Seed** field. This seed is used to generate random numbers that determine the position of nodes on the canvas.
5. Select the check box **Use Altitudes from Terrain File** if terrain elevation data are used in the scenario and the nodes' altitudes should be read from the terrain file (see Section 4.2.2).
6. Under **Placement Data**, specify the coordinates of the origin and the dimensions of the area in which the nodes are placed (for Cartesian coordinate system) or the latitude and longitude of the south-west and north-east corners of the placement area (for Latitude-Longitude coordinate system).
7. *Click* **Apply** *or* **OK***.*

TABLE 4-21. Command Line Equivalent of Node Placement Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Placement Model	Global	`NODE-PLACEMENT`
Use Altitudes from Terrain File	Global, Node	`MOBILITY-GROUND-NODE`

4.2.3.2.3 Configuring Individual Node Placement Parameters

To configure node placement parameters for a specific node, do the following:

1. Go to **Default Device Properties Editor > Node Configuration > Mobility and Placement**.
2. If terrain elevation data are used in the scenario (see Section 4.2.2) and the node's altitude should be read from the terrain file, then set **Use Altitudes from Terrain File** to *Yes*.

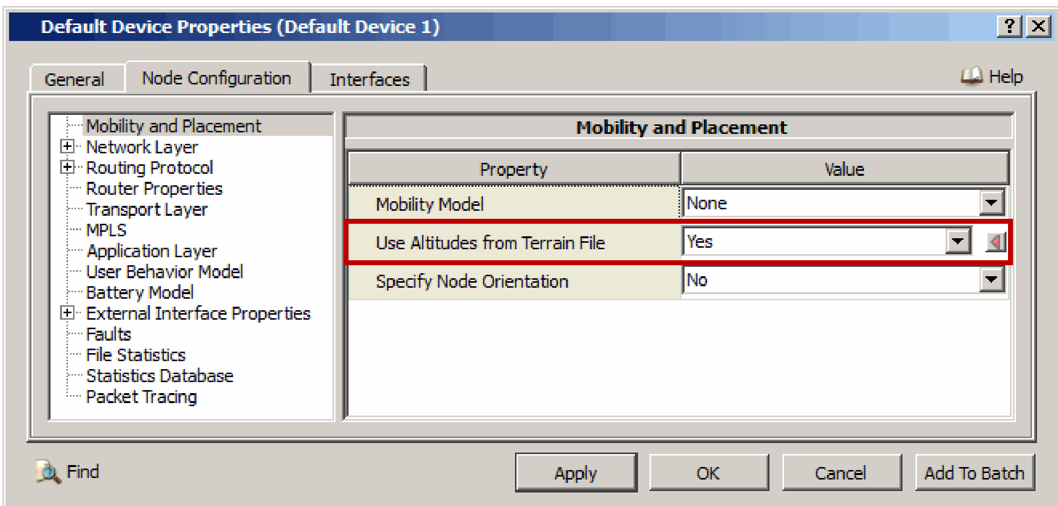


FIGURE 4-12. Setting Node Altitude Parameters

TABLE 4-22. Command Line Equivalent of Node Altitude Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Use Altitudes from Terrain File	Global, Node	`MOBILITY-GROUND-NODE`

3. To specify the node orientation, set **Specify Node Orientation** to *Yes* and set the dependent parameters listed in Table 4-23.

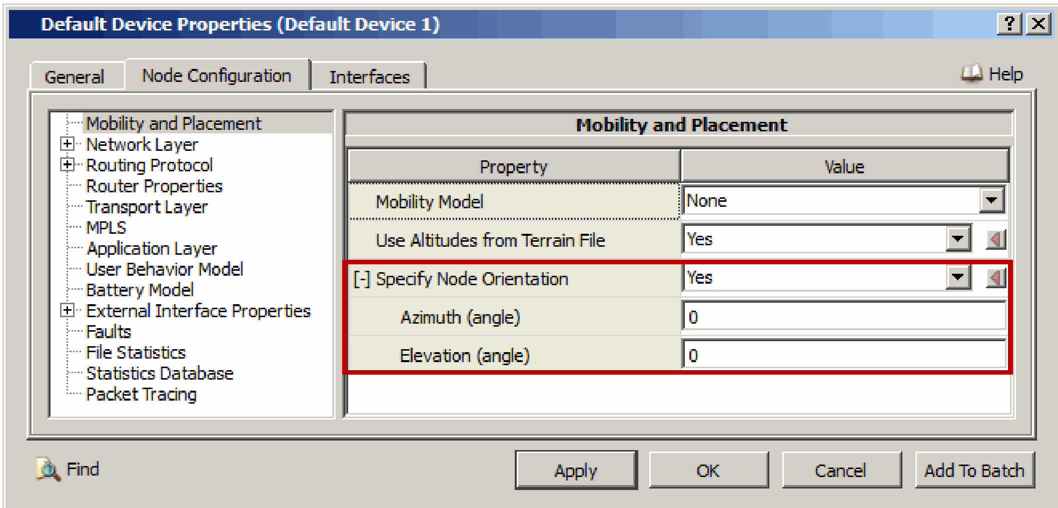


FIGURE 4-13. Setting Node Orientation

TABLE 4-23. GUI Node Orientation Parameters

GUI Parameter	Description
Azimuth	Initial (at time 0) azimuth of the node.
Elevation	Initial (at time 0) elevation of the node.

Note: GUI parameters **Azimuth** and **Elevation** do not have direct equivalents in the command line interface. In command line, the initial azimuth and elevation of a node are specified with the initial node position in the node placement file.

4.2.4 Node Properties

Node properties are node-specific attributes of nodes. Some of these are display attributes (such as icons) that are used only by the GUI and do not have equivalent parameters for the command line interface.

4.2.4.1 Command Line Configuration

The configure node properties for the command line interface, include the parameters listed in Table 4-24 in the scenario configuration (.config) file.

TABLE 4-24. Node Properties Parameters

Parameter	Value	Description
HOSTNAME *Optional* *Scope:* Node	String	Name associated with the node. This parameter is used to associate a meaningful name with a node and is printed in the statistics file. This name is also displayed in EXata GUI if the scenario is loaded in the GUI.

4.2.4.2 GUI Configuration

To configure node properties in the GUI, do the following:

1. Go to **Default Device Properties Editor > General**.
2. Set the node name parameters listed in Table 4-25.

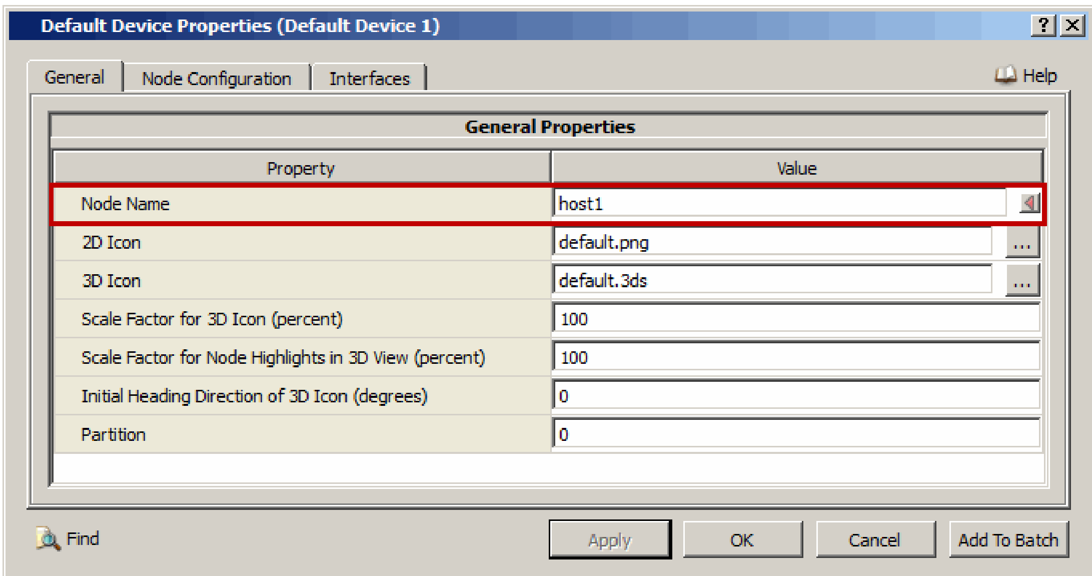


FIGURE 4-14. Setting General Node Parameters

TABLE 4-25. Command Line Equivalent of Node Name Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Node Name	Node	`HOSTNAME`

3. Set the icon parameters listed in Table 4-26.

Notes: 1. Icon parameters can be configured for the following GUI components: Default Device, Switch, ATM Device, Hierarchy, Wired Subnet, Wireless Subnet, and Satellite. These parameters are listed in the **General** tab of the corresponding Properties Editor.

2. Icon parameters do not have equivalents in the command line interface.

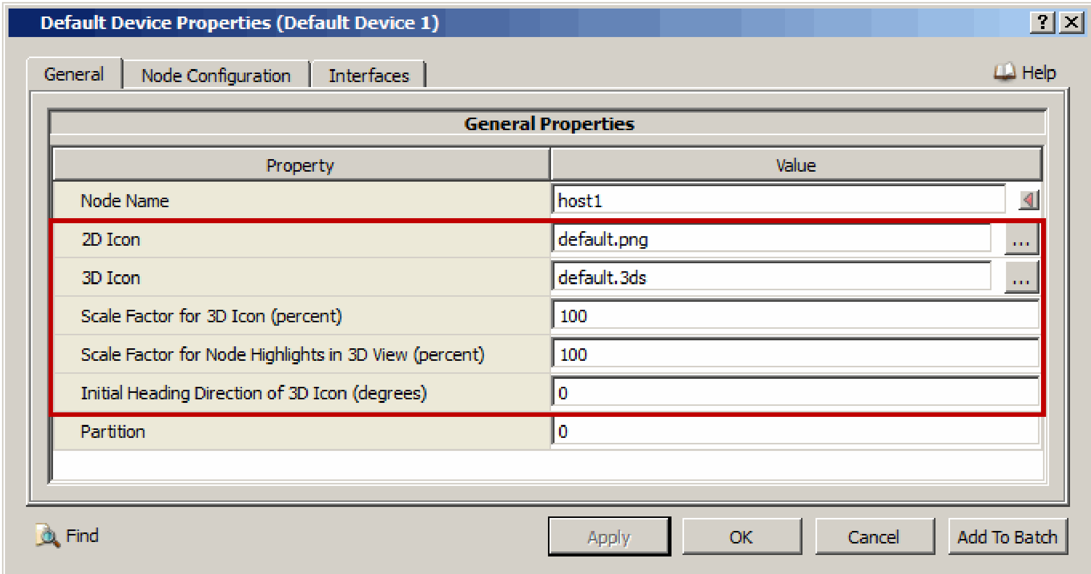


FIGURE 4-15. Setting Icon Parameters

TABLE 4-26. Icon Parameters

GUI Parameter	Scope of GUI Parameter	Description
2D Icon	Node, Switch, ATM Device, Hierarchy, Wired, Subnet, Wireless Subnet, Satellite	Image file used to display the component in the GUI when a 2-dimensional plane of view is selected.
3D Icon	Node, Switch, ATM Device, Hierarchy, Wired, Subnet, Wireless Subnet, Satellite	Image file used to display the component in the GUI when the 3D view is selected.
Scale Factor for 3D Icon	Node, Switch, ATM Device, Hierarchy, Wired, Subnet, Wireless Subnet, Satellite	Scale factor used for displaying the 3D icon. By default, this parameter is set to 100. The size of the icon can be increased or decreased by setting this to a higher or lower value. **Note:** The size of the icon does not change in 2D view.

TABLE 4-26. Icon Parameters (Continued)

GUI Parameter	Scope of GUI Parameter	Description
Scale Factor for Node Highlights in 3D View	Node	Scale factor for the size of node highlights (colored triangles) in 3D view (see Section 6.2). By default, this parameter is set to 100. The size of the triangles can be increased or decreased by setting this to a higher or lower value. **Note:** The size of the triangles does not change in 2D view.
Initial Heading Direction of 3D Icon	Node	Initial direction along which the node icon is aligned in 3D view. (0 degrees means the icon is aligned along the Y axis in the XY plane.)

4.2.5 Topology Specification

The next step is to instantiate nodes and specify the network topology of the scenario, i.e., identify the wired and wireless links between nodes and the subnets that each node belongs to. For the command line interface, nodes are also created as part of the topology specification.

Note: In a scenario, background traffic can be configured on point-to-point links. The effect of the background traffic is to reduce the effective bandwidth available to other (application and control) traffic in the scenario. Refer to *Developer Model Library* for details.

4.2.5.1 Command Line Configuration

For the command line interface, the keywords `SUBNET`, `LINK` and `ATM-LINK` are used to instantiate the nodes and define the network topology. Every node in the scenario must appear in at least one `SUBNET`, `LINK` or `ATM-LINK` declaration.

4.2.5.1.1 Subnets

The `SUBNET` keyword is used to describe a network composed of two or more nodes. It also instantiates the network interfaces on those nodes to connect them to the network.

The following syntax is used to define a subnet:

```
SUBNET <Subnet Address> {<List of Nodes>}
```

where

`<Subnet Address>`	Address of the subnet in the EXata N syntax (see Section 2.2.4).
`<List of Nodes>`	List of node IDs in the subnet, separated by commas. A range of nodes using the keyword `thru` can also be specified.

The following are examples of declarations to create IPv4 subnets:

```
SUBNET N8-192.168.1.0 {1, 2, 5, 9}
SUBNET N16-1.0.0 {3 thru 7}
SUBNET N24-1.0.0.0 {2, 4, 5 thru 7, 10}
```

The following is an example of a declaration to create IPv6 subnets:

```
SUBNET N64-2000:0:0:1:: {1, 3 thru 5, 8 thru 10}
```

To create a subnet in a dual-IP network, both IPv4 and IPv6 addresses need to be listed. For example:

```
SUBNET N8-192.168.1.0 N64-2000:0:0:1:: {1, 2, 3}
```

A subnet can be wired or wireless. Whether a subnet is a wired subnet or a wireless subnet is determined by the MAC protocol specified for the subnet (see Section 4.2.8.2).

Interface Addresses

A node can be in more than one subnet and has an interface on each of the subnets of which it is a member. The interface addresses can be derived from the subnet address used in the subnet declaration, as explained in Section 2.2.4.

As an example, consider the following declarations in the scenario configuration (.config) file:

```
SUBNET N8-1.0 {1 thru 3}
SUBNET N8-2.0 {2 thru 4}
```

In the above example, nodes 2 and 3 are in both subnets and have interfaces to both of them. Nodes 1 and 4 have one interface each. The following table lists the interface addresses of the four nodes.

Node ID	Subnet Address	Interface Address
1	N8-1.0	0.0.1.1
2	N8-1.0	0.0.1.2
2	N8-2.0	0.0.2.1
3	N8-1.0	0.0.1.3
3	N8-2.0	0.0.2.2
4	N8-2.0	0.0.2.3

4.2.5.1.2 Links

A point-to-point link is a communication channel that is used to connect two communicating devices, like two nodes in a network. In EXata, a point-to-point link can be one of three types:

- **Wired**: A wired point-to-point link models a wired physical connection between two devices.
- **Wireless**: A wireless point-to-point link models a wireless communication medium between two nodes using some form of energy like radio frequency (RF), infrared light, etc., to information transfer. The link is assumed to be error-free
- **Microwave**: A microwave point-to-point link is a special type of wireless link for which the propagation characteristics can be configured.

The `LINK` keyword is used to define a point-to-point link. A link is a special case of a subnet with exactly two nodes. The keyword `ATM-LINK` is similar to the keyword `LINK` and is used to define a point-to-point link for ATM networks.

The following syntax is used to define a point-to-point link:

```
LINK <Subnet Address> {<Node 1>, <Node 2>}
```

where

`<Subnet Address>`	Address of the link in the EXata N syntax (see Section 2.2.4).
`<Node 1>, <Node 2>`	Node IDs of nodes at the end points of the link.

Interface Addresses

A node can be connected to more than one link and/or subnet. A node has an interface to each link or subnet to which it is connected. The interface addresses are derived from the link or subnet address as explained in Section 4.2.5.1.1.

Link Properties

The configure link properties for the command line interface, include the parameters listed in Table 4-27 in the scenario configuration (.config) file.

TABLE 4-27. Link Properties Parameters

Parameter	Value	Description
`LINK-PHY-TYPE` *Optional* *Scope:* Global, Subnet, Interface	List: • `WIRED` • `WIRELESS` • `MICROWAVE` *Default:* `WIRED`	Link type.

Set the MAC protocol for the link by setting the parameter `LINK-MAC-PROTOCOL`, as described in Section 4.2.8.2.1. In addition, if `LINK-PHY-TYPE` is set to `MICROWAVE`, then set the microwave link parameters as described in the Microwave Links section of *Wireless Model Library*.

4.2.5.2 GUI Configuration

In the GUI, a wireless subnet is modeled by placing a wireless subnet (cloud) icon on the canvas and connecting it to the nodes that belong to the subnet. The link between a wireless subnet icon and a node is a logical connection, not a physical link.

A wired subnet is modeled by placing a wired subnet (hub) icon on the canvas and connecting it to the nodes that belong to the subnet.

Point-to-point connections between devices are modelled by direct links between them.

> **Note:** Every scenario has a default wireless subnet. Any node placed on the canvas that is not connected to any other device or subnet belongs to the default wireless subnet. The default wireless subnet is not represented by any icon on the canvas.

For details of creating subnets and links in the GUI, see Section 3.2.1.

By default, a point-to-point link created in the GUI is a wired link. To change the link properties, do the following:

1. Go to **Point-to-Point Link Properties Editor > General**.
2. Set the parameters listed in Table 4-28.

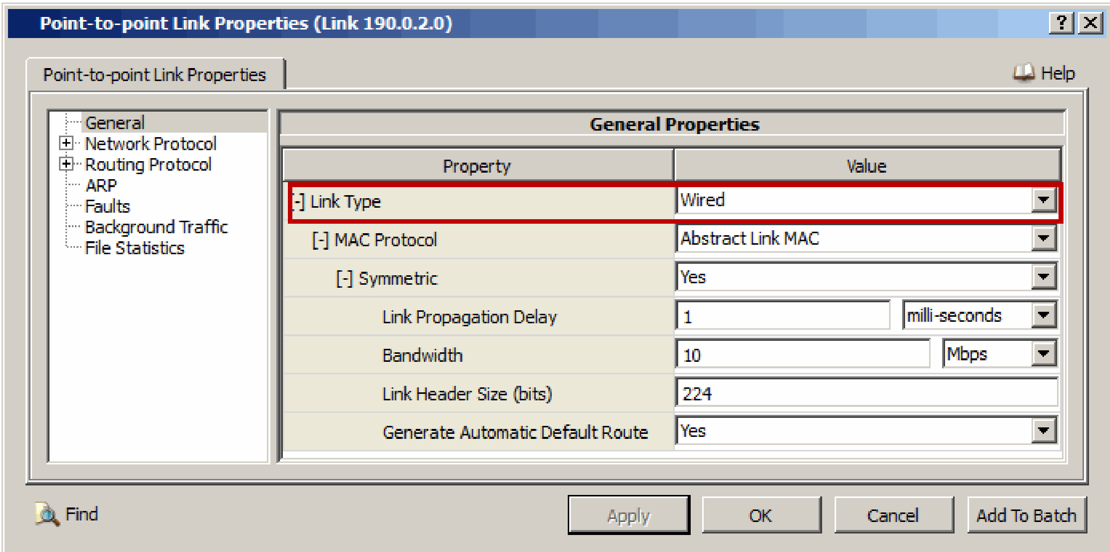


FIGURE 4-16. Setting Link Properties

TABLE 4-28. Command Line Equivalent of Link Properties Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Link Type	Point-to-point Link	`LINK-PHY-TYPE`

3. Set the dependent parameters for the selected link model. Set the MAC protocol as described in Section 4.2.8.2.2. In addition, if **Link Type** is set to *Microwave*, then set the microwave link parameters as described in the Microwave Links section of *Wireless Model Library*.

4.2.6 Mobility Specification

A mobility model determines how each node moves during the simulation, i.e., how the positions of nodes at different simulation times are calculated.

4.2.6.1 Command Line Configuration

The configure mobility characteristics for the command line interface, include the parameters listed in Table 4-29 in the scenario configuration (.config) file.

TABLE 4-29. Mobility Parameters

Parameter	Value	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 4-30 for a description of mobility models.
MOBILITY-POSITION-GRANULARITY *Dependency:* MOBILITY ≠ NONE *Optional* *Scope:* Global, Node	Real *Range:* > 0.0 *Default:* 1.0 *Unit:* meters	Distance by which a node moves in a single step. **Note:** The mobility position granularity also impacts the frequency of pathloss updates. Finer position granularity leads to more position updates, more pathloss updates, and generally greater accuracy. Courser granularity leads to fewer position and pathloss updates, and less accuracy, but faster execution. See Section 4.5 for details.

Table 4-30 describes the different mobility models in EXata. See the corresponding model library for the description of each model and its parameters.

TABLE 4-30. Mobility Models

Command Line Name	GUI Name	Description	Model Library
FILE	File	File-based mobility model The node positions at different simulation times are read from a file. The node moves from one position to the next in a straight line at a constant speed.	Wireless

TABLE 4-30. Mobility Models (Continued)

Command Line Name	GUI Name	Description	Model Library
`GROUP`	Group Mobility	Group-based mobility model. In this model, groups of nodes move together. The entire group moves following the Random Waypoint model, and each node moves within the group area, also following the Random Waypoint model. **Note:** This mobility model can only be used if the Group placement model is used to place nodes (see Section 4.2.3).	Wireless
`RANDOM-WAYPOINT`	Random Waypoint	Random Waypoint mobility model. The node selects a random position, moves towards it in a straight line at a constant speed that is randomly selected from a range, and pauses at that destination. The node repeats this process. throughout the simulation.	Wireless

4.2.6.2 GUI Configuration

In Architect, the mobility model can be specified for individual nodes. In addition, waypoints can be set up for nodes on the canvas.

Note: If the File node placement model is used to place nodes on the canvas (see Section 4.2.3), then any mobility waypoint information (i.e., positions of nodes after time 0) contained in the node position file is also imported. If the node position file contains waypoints for a node, then the mobility model for that node is automatically set to File-based mobility. Waypoint markers are placed on the canvas using the waypoint information in the node position file. The mobility model for a node can be modified, as described in Section 4.2.6.2.1.

4.2.6.2.1 Specifying Mobility Model

To specify a mobility model for a node, do the following:

1. Go to **Default Device Properties Editor > Node Configuration > Mobility and Placement**.
2. Set the parameters listed in Table 4-31. The available mobility models are described in Table 4-30.

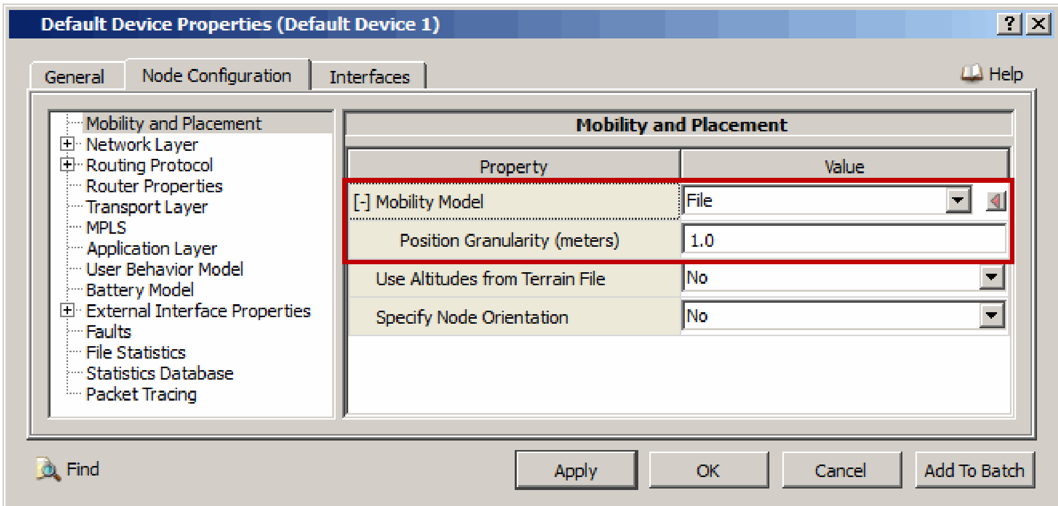


FIGURE 4-17. Setting Mobility Properties

TABLE 4-31. Command Line Equivalent of Mobility Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Mobility Model	Node	`MOBILITY`
Position Granularity	Node	`MOBILITY-POSITION-GRANULARITY`

3. Set the dependent parameters for the selected mobility model. See the model library referenced in Table 4-30 for details.

> **Note:** To use the File-based mobility model, the node position file must be specified in the **Node Placement Wizard** (see Section 4.2.3.2.2).

4.2.6.2.2 Setting Mobility Waypoints on Canvas

Setting mobility waypoints for nodes on the canvas is described in Section 3.4.5. Waypoint attributes can be modified by using the **Mobility Waypoint Editor**, as described in Section 3.4.5.1.

When mobility waypoints are set for a node, the **Mobility Model** parameter (see Figure 4-17) is automatically set to *File* and the mobility waypoint attributes (location and time) are added to the node position file. When the scenario is run, the simulator uses the File-based mobility model (see Table 4-30) with the node position file created by Architect.

> **Note:** Mobility waypoint markers associated with a node are not automatically deleted even if the mobility model for the node is changed, i.e., parameter **Mobility Model** is set to a value other than *File*.

4.2.7 Channel Properties

This section describes the parameters for setting up channels and the related pathloss, fading, and shadowing models in a wireless scenario.

4.2.7.1 Command Line Configuration

To configure channel properties for the command line interface, include the parameters listed in Table 4-32 in the scenario configuration (.config) file.

TABLE 4-32. Channel Configuration Parameters

Parameter	Value	Description
`PROPAGATION-CHANNEL-FREQUENCY` *Optional* (see note) *Scope:* Global *Instances:* channel index	Real *Range:* `> 0.0` *Unit:* Hz	Channel frequency. Include as many instances of this parameter as the number of channels in the simulation. **Note:** This parameter must be specified for a wireless scenario.
`PROPAGATION-CHANNEL-NAME` *Optional* *Scope:* Global *Instances:* channel index	String *Default:* :See note	Name assigned to the channel. A channel name must start with an upper case or lower case letter and can not contain any of the following characters: `comma space ( ) { } [ ]` Channel names must be unique. **Note:** If the name is not specified for a channel, then the default name assigned to channel with index *x* is `channelx`.
`PROPAGATION-PATHLOSS-MODEL` *Dependency:* `PROPAGATION-CHANNEL-FREQUENCY` is included *Required* *Scope:* Global *Instances:* channel index	List: • `COST231-HATA` • `COST231-WALFISH-IKEGAMI` • `FREE-SPACE` • `ITM` • `OKUMURA-HATA` • `PATHLOSS-MATRIX` • `STREET-M-TO-M` • `STREET-MICROCELL` • `SUBURBAN` • `TIREM` • `TWO-RAY` • `URBAN-MODEL-AUTOSELECT`	Pathloss model for the channel. See Table 4-33 for a description of the pathloss models. **Note:** This parameter must be specified for each channel in a wireless scenario.

TABLE 4-32. Channel Configuration Parameters (Continued)

Parameter	Value	Description
`PROPAGATION-SHADOWING-MODEL` *Dependency:* `PROPAGATION-CHANNEL-FREQUENCY` is included *Optional* *Scope:* Global *Instances:* channel index	List: • `NONE` • `CONSTANT` • `LOGNORMAL` *Default:* `NONE`	Shadowing model for the channel. If this parameter is set to `NONE`, no shadowing model is used for the channel. See Table 4-34 for a description of the shadowing models.
`PROPAGATION-FADING-MODEL` *Optional* *Scope:* Global *Instances:* channel index	List: • `NONE` • `FAST-RAYLEIGH` • `RAYLEIGH` • `RICEAN` *Default:* `NONE`	Fading model for the channel If this parameter is set to `NONE`, no fading model is used for the channel. See Table 4-35 for a description of the fading models.
`PROPAGATION-ENABLE-CHANNEL-OVERLAP-CHECK` *Optional* *Scope:* Global *Instances:* channel index	List: • `YES` • `NO` *Default:* `NO`	Enables the Inter-channel Interference model. If the Inter-channel Interference model is enabled, both co-channel and inter-channel interference are taken into account; otherwise, only co-channel interference is taken into account. Refer to *Wireless Model Library* for details of the Inter-channel Interference model.
`PROPAGATION-SPEED` *Optional* *Scope:* Global *Instances:* channel index	Real *Range:* `> 0.0` *Default:* `3.0e8` *Unit:* meters/sec	Signal propagation speed.
`PROPAGATION-LIMIT` *Optional* *Scope:* Global *Instances:* channel index	Real *Default:* `-110.0` *Unit:* dBm	Threshold for delivering signals to nodes. Signals with power below this limit (before accounting for antenna gain at the receiver) are not delivered to nodes. This parameter is meant for optimizing simulation performance. A lower value of the parameter results in a more accurate simulation but at the expense of longer execution time. See Section 4.5 for details.
`PROPAGATION-MAX-DISTANCE` *Optional* *Scope:* Global *Instances:* channel index	Real *Range:* ≥ `0.0` *Default*: `0.0` (see note) *Unit:* meters	Maximum distance for which a node's transmission is considered for communication or interference. **Note:** If this parameter is set to a value ≤ `0.1`, then it is not considered in the estimation of a node's propagation range, i.e., the maximum distance is effectively infinity. In this case, the propagation range is determined only by the parameter `PROPAGATION-LIMIT`. This parameter is important for optimization (speed/accuracy tradeoff). See Section 4.5 for details.

TABLE 4-32. Channel Configuration Parameters (Continued)

Parameter	Value	Description
`PROPAGATION-COMMUNICATION-PROXIMITY` *Optional* *Scope:* Global *Instances:* channel index	Real *Range:* `> 0.0` *Default*: `400.0` *Unit:* meters	Communication proximity used to calculate the frequency of pathloss updates (see Section 4.5). This parameter should be set to the approximate optimistic radio range. This parameter is important for optimization (speed/accuracy tradeoff). See Section 4.5 for details.
`PROPAGATION-PROFILE-UPDATE-RATIO` *Optional* *Scope:* Global *Instances:* channel index	Real *Range:* `[0.0, 1.0]` *Default*: `0.0`	Update ratio used to calculate the frequency of pathloss updates (see Section 4.5). A larger value of this parameter results in a more aggressive optimization. This parameter is important for optimization (speed/accuracy tradeoff). See Section 4.5 for details.

Table 4-33 describes the different pathloss models in EXata. See the corresponding model library for the description of each model and its parameters.

TABLE 4-33. Pathloss Models

Command Line Name	GUI Name	Description	Model Library
`COST231-HATA`	COST231-HATA	COST 231-Hata pathloss model. This model can be used for urban, suburban, or open areas. It is a refinement of the Okumura-Hata pathloss model.	Urban Propagation
`COST231-WALFISH-IKEGAMI`	COST231-WALFISH-IKEGAMI	COST 231-Walfish-Ikegami pathloss model. This model can be used for urban or metropolitan areas.	Urban Propagation
`FREE-SPACE`	Free Space	Friis free-space pathloss model. The model assumes an omni-directional line-of-sight propagation path. The signal strength decays with the square of the distance between the transmitter and receiver.	Wireless
`ITM`	Irregular Terrain Model	Irregular Terrain Model, also known as the Longley-Rice model. This model uses the information from a terrain data file to calculate line-of-sight between nodes, ground reflection characteristics, and pathloss.	Wireless
`OKUMURA-HATA`	OKUMURA-HATA	Okumura-Hata pathloss model for macro-cellular systems. This model can be used for urban, suburban, or open areas.	Urban Propagation

TABLE 4-33. Pathloss Models (Continued)

Command Line Name	GUI Name	Description	Model Library
PATHLOSS-MATRIX	Pathloss Matrix	Matrix-based pathloss model. This model uses a four-dimensional matrix of pathloss values indexed by source node, destination node, simulation time, and channel number.	Wireless
STREET-M-TO-M	Street M-To-M	Street mobile-to-mobile pathloss model. This model calculates pathloss between a source and a destination in an urban canyon communicating across several building obstacles.	Urban Propagation
STREET-MICROCELL	Street Microcell	Street micro-cell pathloss model. This model calculates the path-loss between transmitter-receiver pairs that are located in adjacent streets in an urban canyon.	Urban Propagation
SUBURBAN	Suburban	Suburban pathloss model. This model characterizes propagation in a suburban environment and takes into account the effects of terrain and foliage on signals.	Urban Propagation
TIREM	TIREM	Terrain Integrated Rough Earth Model. This model considers terrain effects, transmitter and receiver attributes such as antenna height and frequency, and atmospheric and ground constants. This model is distributed by the Joint Spectrum Center of the Department of Defense and is interfaced with EXata. This model requires a terrain data file.	TIREM Advanced Propagation
TWO-RAY	Two Ray	Two-ray pathloss model. The two-ray pathloss model considers a line-of-sight path and a reflection from flat earth in pathloss calculation.	Wireless
URBAN-MODEL-AUTOSELECT	Urban Model Autoselect	Urban pathloss model auto-selection feature. This feature automatically selects the most appropriate model based on the relative situation of the communicating nodes and the communication environment. The model is updated during the simulation as the nodes move.	Urban Propagation

Table 4-34 describes the different shadowing models in EXata. See the corresponding model library for the description of each model and its parameters.

TABLE 4-34. Shadowing Models

Command Line Name	GUI Name	Description	Model Library
`CONSTANT`	Constant	Constant shadowing model. This model uses a constant shadowing offset.	Wireless
`LOGNORMAL`	Lognormal	Lognormal shadowing model. This model uses a lognormal distribution for the shadowing value.	Wireless

Table 4-35 describes the different fading models in EXata. See the corresponding model library for the description of each model and its parameters.

TABLE 4-35. Fading Models

Command Line Name	GUI Name	Description	Model Library
`FAST-RAYLEIGH`	Fast Rayleigh	Fast Rayleigh fading model. The fast Rayleigh fading model is a statistical model to represent the fast variation of signal amplitude at the receiver due to the motion of the transmitter/receiver pair.	Wireless
`RAYLEIGH`	Rayleigh	Rayleigh fading model. Rayleigh fading model is a statistical model to represent the fast variation of signal amplitude at the receiver. In wireless propagation, Rayleigh fading occurs when there is no line of sight between the transmitter and receiver.	Wireless
`RICEAN`	Ricean	Ricean fading model. This model can be used for scenarios where there is line of sight communication and the line of sight signal is the dominant signal seen at the receiver.	Wireless

4.2.7.2 GUI Configuration

To configure channel properties in the GUI, do the following:

1. Go to **Scenario Properties Editor > Channel Properties**.
2. Set **Number of Channels** to the desired value.

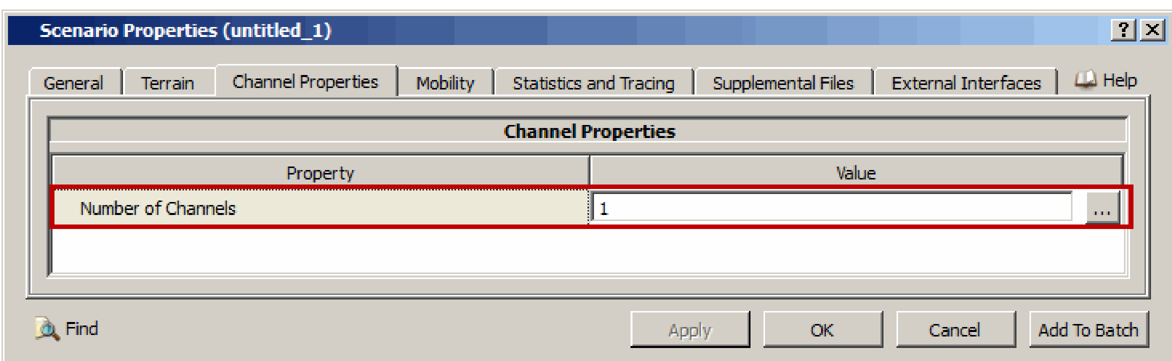


FIGURE 4-18. Setting Number of Channels

3. Click on the **Open Array Editor** ... button in the **Value** column. This opens the Array Editor.
4. In the left panel of the Array Editor, select the index of the channel to be configured.

   For each channel, set the parameters listed in Table 4-36.

   The available pathloss models are described in Table 4-33.

   The available shadowing models are described in Table 4-34.

   The available fading models are described in Table 4-35.

   The speed and accuracy trade-off parameters (**Propagation Limit**, **Maximum Propagation Distance**, **Propagation Communication Proximity**, *and* **Propagation Profile Update Ratio** *are described in* Section 4.5.

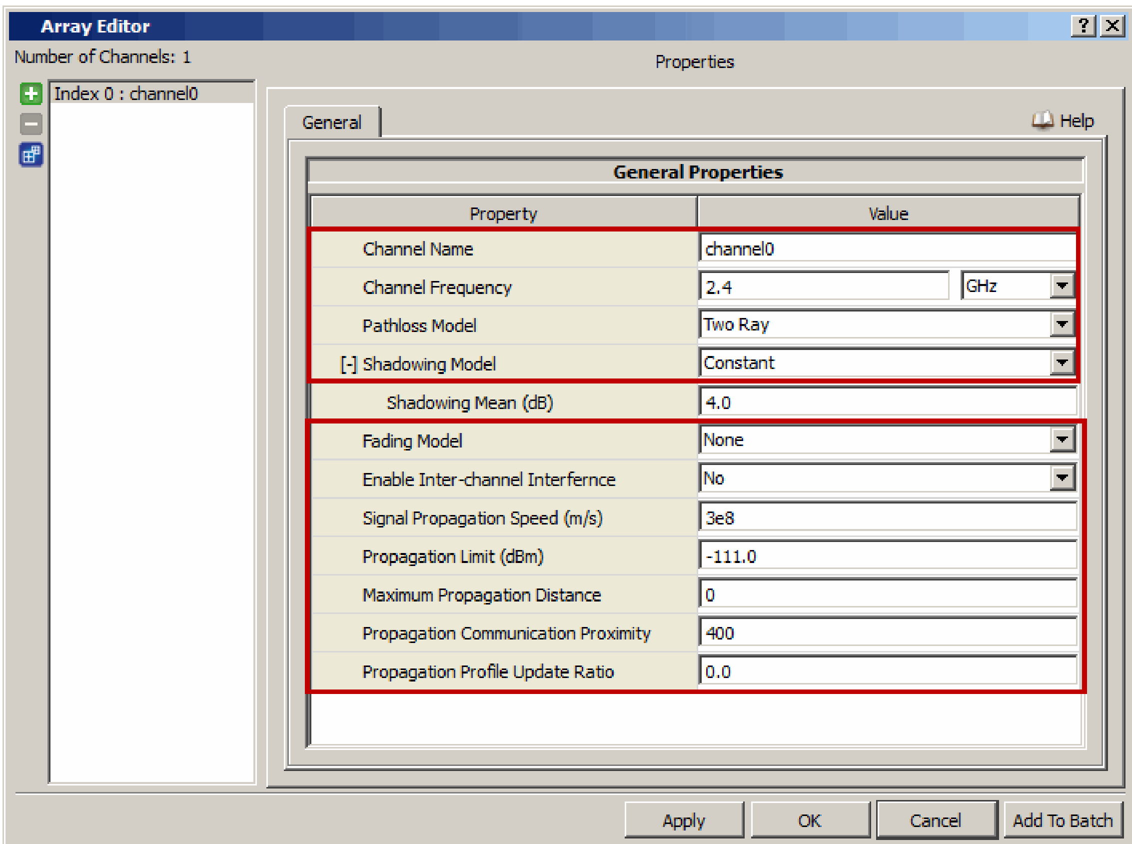


FIGURE 4-19. Setting Channel Properties

TABLE 4-36. Command Line Equivalent of Channel Configuration Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Channel Name	Global	`PROPAGATION-CHANNEL-NAME`
Channel Frequency	Global	`PROPAGATION-CHANNEL-FREQUENCY`
Pathloss Model	Global	`PROPAGATION-PATHLOSS-MODEL`
Shadowing Model	Global	`PROPAGATION-SHADOWING-MODEL`

TABLE 4-36. Command Line Equivalent of Channel Configuration Parameters (Continued)

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Fading Model	Global	`PROPAGATION-FADING-MODEL`
Enable Inter-channel Interference	Global	`PROPAGATION-ENABLE-CHANNEL-OVERLAP-CHECK`
Signal Propagation Speed	Global	`PROPAGATION-SPEED`
Propagation Limit	Global	`PROPAGATION-LIMIT`
Maximum Propagation Distance	Global	`PROPAGATION-MAX-DISTANCE`
Propagation Communication Proximity	Global	`PROPAGATION-COMMUNICATION-PROXIMITY`
Propagation Profile Update Ratio	Global	`PROPAGATION-PROFILE-UPDATE-RATIO`

5. Set the dependent parameters for the selected pathloss, shadowing, and fading models. See the model library referenced in Table 4-33, Table 4-34, and Table 4-35 for details. Refer to Wireless Model Library for the Inter-channel Interference model.

4.2.8 Configuring the Protocol Stack

Network protocols are usually divided into layers as defined by the OSI Reference Model. These layers form a protocol stack. EXata primarily models the TCP/IP protocol stack, which consists of the following five layers (from top to bottom): Application Layer, Transport Layer, Network Layer, MAC/Data Link Layer, and Physical Layer. At each layer, multiple protocols are available.

This section describes how to configure protocols at each layer. In EXata, both wired and wireless networks implement the top four layers. For wired networks, the Physical Layer functionality is integrated with the MAC Layer. For wireless networks, the Physical Layer needs to be configured by specifying the antenna model and the radio model.

4.2.8.1 Physical Layer Configuration

For each wireless interface in the scenario, Physical Layer properties need to be configured, which consist of the following:

- Listenable and listening channels
- Radio model
- Antenna model
- Thermal noise parameters
- Radio energy models

4.2.8.1.1 Listenable and Listening Channels

Listenable and listening channels identify which channels a node can potentially listen to during the simulation and which channel it is configured to listen to when simulation starts.

4.2.8.1.1.1 Command Line Configuration

To set the listenable and listening channels for the command line interface, include the parameters listed in Table 4-37 in the scenario configuration (.config) file.

TABLE 4-37. Listenable and Listening Channel Parameters

Parameter	Value	Description
`PHY-LISTENABLE-CHANNELS` *Dependency:* Scenario is a wireless scenario *Optional* (see note) *Scope:* All	Comma-separated list of channel names and frequencies	List of listenable channels, i.e., the list of channels that the node can potentially listen to during the simulation. Channels in the list can be specified by their names or frequencies (in Hz). However, if two or more channels have the same frequency, then those channels cannot be specified using the frequency. **Note:** For a wireless scenario either (but not both) `PHY-LISTENABLE-CHANNEL-MASK` or `PHY-LISTENABLE-CHANNELS` must be specified.
`PHY-LISTENABLE-CHANNEL-MASK` *Dependency:* Scenario is a wireless scenario *Optional* (see note) *Scope:* All	String of 0's and 1's	Listenable channel mask. A channel mask is a string of 0's and 1's. The length of the string is equal to the number of channels in the scenario. Each bit in the mask corresponds to a channel: the left-most bit corresponds to the first channel (with index 0), the next bit corresponds to the second channel, and so on. If a bit value in the listenable channel mask is 1, the node can potentially listen to the corresponding channel during the simulation; otherwise, the node can not listen to the corresponding channel at any time during the simulation. **Note:** For a wireless scenario either (but not both) `PHY-LISTENABLE-CHANNEL-MASK` or `PHY-LISTENABLE-CHANNELS` must be specified.

TABLE 4-37. Listenable and Listening Channel Parameters (Continued)

Parameter	Value	Description
PHY-LISTENING-CHANNELS *Dependency:* Scenario is a wireless scenario *Optional* (see note) *Scope:* All	String of 0's and 1's	List of listening channels, i.e., the list of channels that the node is configured to listen to at the start of simulation. Channels in the list can be specified by their names or frequencies (in Hz). However, if two or more channels have the same frequency, then those channels cannot be specified using the frequency. A channel can be in the list of listening channels only if it is also in the list of listenable channels. **Note:** For a wireless scenario either (but not both) PHY-LISTENING-CHANNEL-MASK or PHY-LISTENING-CHANNELS must be specified.
PHY-LISTENING-CHANNEL-MASK *Dependency:* Scenario is a wireless scenario *Optional* (see note) *Scope:* All	String of 0's and 1's	Listening channel mask. If a bit value is 1, the node is configured (at the start of simulation) to listen to the corresponding channel; otherwise, the node is not configured to listen to the corresponding channel when the simulation starts. A bit for a channel in the listening mask can be set to 1 only if the corresponding bit in the listenable mask is also set to 1. **Note:** For a wireless scenario either (but not both) PHY-LISTENING-CHANNEL-MASK or PHY-LISTENING-CHANNELS must be specified.

Example:

Consider the following configuration that uses channel names:

```
PROPAGATION-CHANNEL-FREQUENCY[0] 2.4e6
PROPAGATION-CHANNEL-NAME[0] wifi-1
PROPAGATION-CHANNEL-FREQUENCY[1] 2.5e6
PROPAGATION-CHANNEL-NAME[1] wimax-1
PROPAGATION-CHANNEL-FREQUENCY[2] 2.6e6
PROPAGATION-CHANNEL-NAME[2] wifi-2
SUBNET N8-1.0 {1 thru 5}
SUBNET N8-2.0 {5 thru 10}
SUBNET N8-3.0 {10 thru 15}
PHY-LISTENABLE-CHANNELS          wifi-1,2.5e6,wifi-2
[N8-1.0] PHY-LISTENING-CHANNELS  wifi-2
[N8-2.0] PHY-LISTENING-CHANNELS  wimmax-1
[N8-3.0] PHY-LISTENING-CHANNELS  wifi-1
```

All three subnets can potentially listen to all three channels. At the start of simulation, subnet N8-1.0 is configured to listen to channel 2 (whose name is wifi-2), subnet N8-2.0 is configured to listen to channel 1 (whose name is wimax-1), and subnet N8-3.0 is configured to listen to channel 0 (whose name is wifi-1).

An alternative configuration for the same scenario using channel masks is as follows:

```
PROPAGATION-CHANNEL-FREQUENCY[0] 2.4e6
PROPAGATION-CHANNEL-FREQUENCY[1] 2.5e6
PROPAGATION-CHANNEL-FREQUENCY[2] 2.6e6
SUBNET N8-1.0 {1 thru 5}
SUBNET N8-2.0 {5 thru 10}
SUBNET N8-3.0 {10 thru 15}
PHY-LISTENABLE-CHANNEL-MASK          111
[N8-1.0] PHY-LISTENING-CHANNEL-MASK  001
[N8-2.0] PHY-LISTENING-CHANNEL-MASK  010
[N8-3.0] PHY-LISTENING-CHANNEL-MASK  100
```

4.2.8.1.1.2 GUI Configuration

The listenable and listening channels can be modified by using the PHY Channel Editor, as described below. See Table 4-37 for a description of listenable and listening channels.

Note: Before setting the listenable and listening channels, the number of channels must be specified using the **Scenario Properties Editor** (see Section 4.2.7).

To configure listenable and listening channels in the GUI, do the following:

1. Go to one of the following locations:

- To set properties for a specific subnet, go to **Wireless Subnet Properties Editor > Physical Layer**.
- To set properties for a specific interface of a node, go to one of the following locations:
  - **Interface Properties Editor > Interfaces > Interface # > Physical Layer**
  - **Default Device Properties Editor > Interfaces > Interface # > Physical Layer**

In this section, we show how to configure listenable and listening channels in the Wireless Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. In the **Value** field for the parameter **Listenable Channels**, click on the ... button. This opens the **PHY Channel List Editor**.

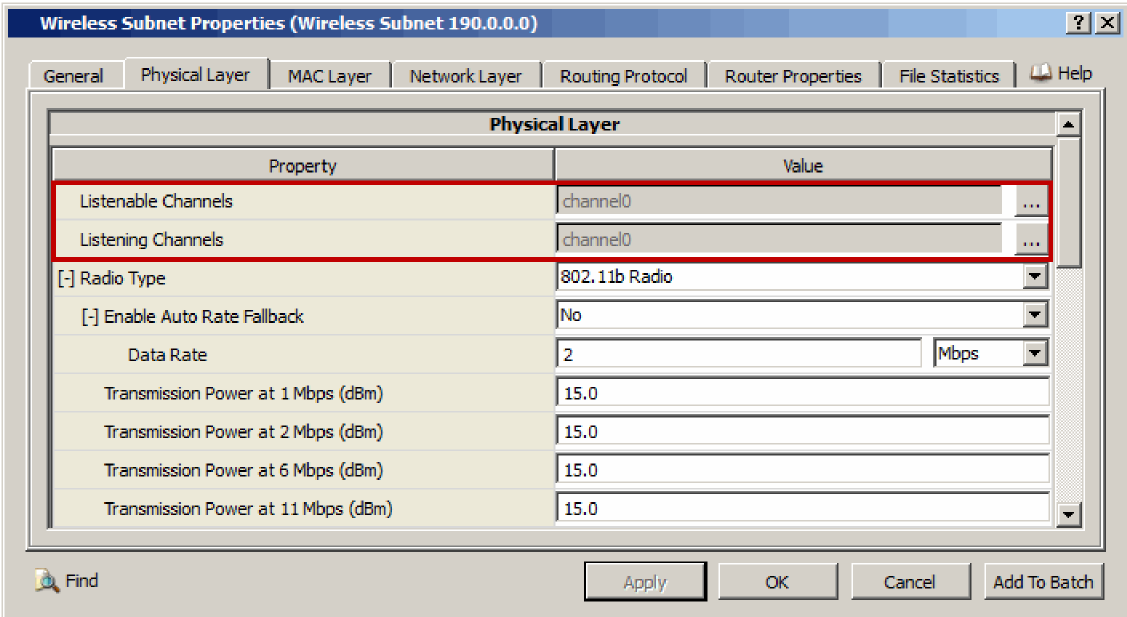


FIGURE 4-20. Specifying Listenable and Listening Channels

TABLE 4-38. Command Line Equivalent of Listenable and Listening Channel Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Listenable Channels	Subnet, Interface	`PHY-LISTENABLE-CHANNELS`
Listening Channels	Subnet, Interface	`PHY-LISTENING-CHANNELS`

3. The **PHY Channel List Editor** displays the frequency and name for each channel. To include a channel in the list of listenable channels, select the check box in the **Enabled** column. Press **Apply** or **OK**. Figure 4-21 shows the **PHY Channel List Editor** for a scenario with four channels.

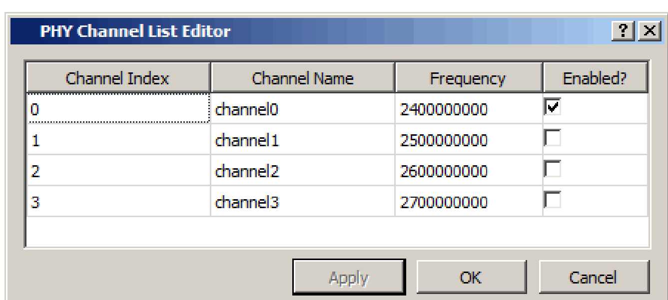


FIGURE 4-21. PHY Channel List Editor

4. Set the parameter **Listening Channels** in a similar way.

4.2.8.1.2 Radio Models

A radio model characterizes the transmitter/receiver behavior at an interface.

4.2.8.1.2.1 Command Line Configuration

To configure the radio model for the command line interface, include the parameters listed in Table 4-39 in the scenario configuration (.config) file.

TABLE 4-39. Radio Configuration Parameters

Parameter	Value	Description
`PHY-MODEL` *Dependency:* Scenario is a wireless scenario *Required* *Scope:* All	List: • `PHY802.11a` • `PHY802.11b` • `PHY802.11n` • `PHY802.15.4` • `PHY802.16` • `PHY-ABSTRACT` • `PHY-GSM` • `PHY-LTE` • `SATELLITE-RSV`	Name of the radio model. See Table 4-40 for a description of radio models.

Table 4-40 describes the different radio models. See the corresponding model library for a detailed description of each model and its parameters.

TABLE 4-40. Radio Models

Command Line Name	GUI Name	Description	Model Library
`PHY802.11a`	802.11a/g Radio	Models the IEEE 802.11a PHY specification. This radio operates in the 5 GHz frequency band, uses Orthogonal Frequency Division Multiplexing (OFDM) and supports the following data rates (in Mbits/s): 6, 9, 12, 18, 24, 36, 48, 54.	Wireless
`PHY802.11b`	802.11b Radio	Models the IEEE 802.11b PHY specification. This radio operates in the 2.4 GHz frequency band, uses Direct Sequence Spread Spectrum (DSSS) and supports the following data rates (in Mbits/s): 1, 2, 5.5, 11.	Wireless
`PHY802.11n`	802.11n Radio	Models the IEEE 802.11n PHY specification. IEEE 802.11n improves the effective throughput of end-user applications on 802.11 devices.	Wireless
`PHY802.15.4`	802.15.4 Radio	Models the IEEE 802.15.4 PHY specification. This radio uses different waveforms in different frequency bands to support different data rates.	Sensor Networks
`PHY802.16`	802.16 Radio	Models the IEEE 802.16 PHY specification. This radio uses OFDM and uses the following modulation and encoding combinations: QPSK 1/2, QPSK 3/4, 16QAM 1/2, 16QAM 3/4, 64QAM 1/2, 64QAM 2/3, and 64QAM 3/4.	Advanced Wireless

TABLE 4-40. Radio Models (Continued)

Command Line Name	GUI Name	Description	Model Library
PHY-ABSTRACT	Abstract	Abstract PHY model. This is a generic PHY model and can be used to simulate different PHYs. This model simulates a PHY that is capable of carrier sensing and is able to work with both BER-based and SNR threshold-based reception models.	Wireless
PHY-GSM	GSM	Models the GSM Physical Layer.	Cellular
PHY-LTE	LTE	Models the LTE Physical Layer.	LTE
SATELLITE-RSV	Satellite RSV PHY	Models the Aloha satellite model with Reed-Solomon/Viterbi (RSV) support. This is a model for satellites in geosynchronous orbits. Both bent-pipe and process payload modes are modeled.	Wireless

4.2.8.1.2.2 GUI Configuration

To set radio configuration parameters in the GUI, do the following:

1. Go to one of the following locations:
    - To set properties for a specific subnet, go to **Wireless Subnet Properties Editor > Physical Layer.**
    - To set properties for a specific interface of a node, go to one of the following locations:
        - **Interface Properties Editor > Interfaces > Interface # > Physical Layer**
        - **Default Device Properties Editor > Interfaces > Interface # > Physical Layer**.

    In this section, we show how to configure the radio model in the Wireless Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. Select the radio model by setting the parameter **Radio Type**. The available radio models are described in Table 4-40.

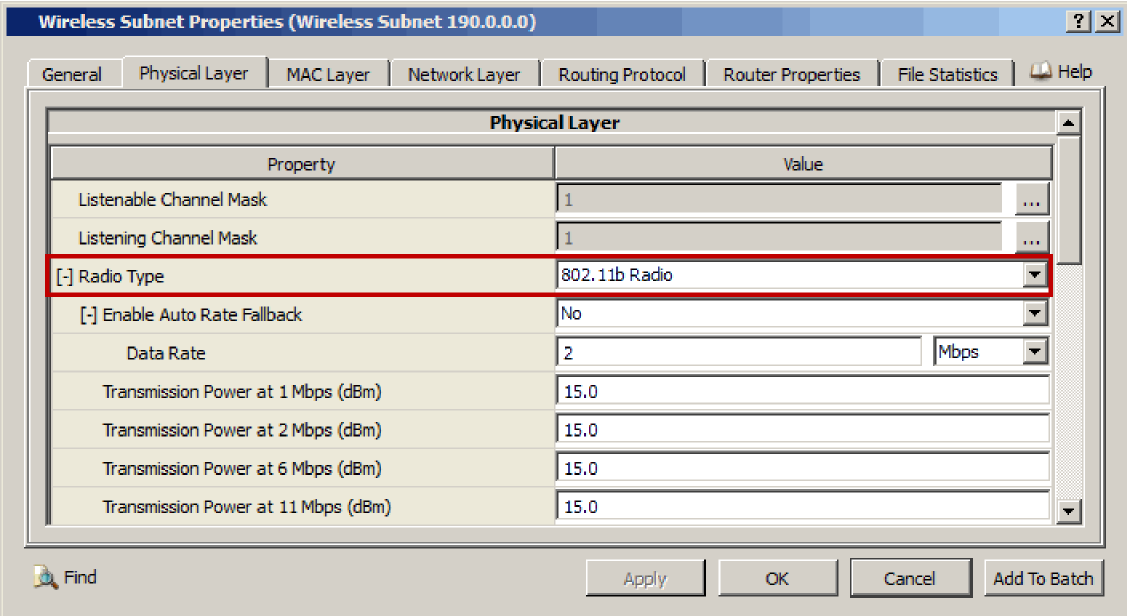


FIGURE 4-22. Setting Radio Configuration Parameters

TABLE 4-41. Command Line Equivalent of Radio Configuration Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Radio Type	Subnet, Interface	`PHY-MODEL`

3. Set the dependent parameters for the selected radio model. See the model library referenced in Table 4-40 for details.

4.2.8.1.3 Antenna Models

Antenna models represent the characteristics of radio antennas. Antennas are the electronic devices which generate waveforms at the transmitter side and detect and receive waveforms at the receiver side. EXata models omni-directional antennas, static patterned antennas, and smart antennas.

The major characteristics of an antenna are the gains it provides during transmission and receiving, and the efficiency or loss due to various mechanical and electronic factors. An omni-directional antenna is the simplest one and provides the same antenna gain regardless of the direction of the signal transmitted or received. A patterned antenna has different gains in different directions. The value of the gain in different directions follow a gain pattern. A smart antenna is a special type of patterned antenna and can be one of two types: switched beam or steerable. A switched beam antenna can utilize multiple antenna patterns. A steerable antenna can rotate the antenna to achieve the maximum gain.

4.2.8.1.3.1 Command Line Configuration

Standard antenna models (omni-directional, patterned, steerable, and switched-beam) or custom antenna models can be used in a scenario. Custom antenna models are defined in an Antenna Models file.

To configure antenna models for the command line interface, include the parameters listed in Table 4-42 in the scenario configuration (.config) file.

TABLE 4-42. Antenna Configuration Parameters

Parameter	Value	Description
`ANTENNA-MODEL-CONFIG-FILE` *Optional* *Scope:* All	Filename	Name of file containing user-defined antenna models. See Section 4.6.4 for details.
`ANTENNA-MODEL` *Optional* *Scope:* All	List: • `OMNIDIRECTIONAL` • `PATTERNED` • `STEERABLE` • `SWITCHED-BEAM` • *Value of any occurrence of parameter* `ANTENNA-MODEL-NAME` *in the file* `ANTENNA-MODEL-CONFIG-FILE` *Default:* `OMNIDIRECTIONAL`	Antenna model name. See Table 4-43 for a description of the antenna models. It is also possible to pre-configure antenna parameters and assign names to the configurations. The file `ANTENNA-MODEL-CONFIG-FILE` is used for this purpose. The name of a user-defined antenna configuration can be specified as the value of the parameter `ANTENNA-MODEL`. See Section 4.6.4 for details.

Table 4-43 describes the different antenna models. See the corresponding model library for a detailed description of each model and its parameters.

TABLE 4-43. Antenna Models

Command Line Name	GUI Name	Description	Model Library
`OMNIDIRECTIONAL`	Omnidirectional	Omnidirectional antenna model. This is the model for the basic antenna, which yields the same antenna gain irrespective of the signal direction.	Wireless
`PATTERNED`	Patterned	Patterned antenna model. The patterned antenna transmits or receives according to a particular pattern specified in the pattern file.	Wireless

TABLE 4-43. Antenna Models (Continued)

Command Line Name	GUI Name	Description	Model Library
`STEERABLE`	Steerable	Steerable antenna model. The steerable antenna can rotate the antenna and uses the direction that yields the maximum antenna gain.	Wireless
`SWITCHED-BEAM`	Switched Beam	Switched-beam antenna model. The switched-beam antenna can switch among multiple antenna patterns and uses the pattern that yields the maximum antenna gain.	Wireless

4.2.8.1.3.2 GUI Configuration

This section describes how to configure antenna models in the GUI. You can specify a standard antenna model, a pre-configured antenna model, or a custom antenna model. Pre-configured antenna models are imported into the scenario by specifying an antenna models file or by using the Antenna Model Editor. Custom antenna models are created using the Antenna Model Editor. Refer to *Wireless Model Library* for details of using the Antenna Model Editor to import, create, and modify antenna models.

Specifying Antenna Models File

To use pre-configured antenna models in a scenario, the name of the file that contains the antenna models must be specified. To specify the name of the antenna models file, do the following (or import this file using the Antenna Model Editor, as described in *Wireless Model Library*):

1. Go to **Scenario Properties Editor > Supplemental Files**.
2. Set the parameter **Antenna Models File** to the name of the Antenna Models file.

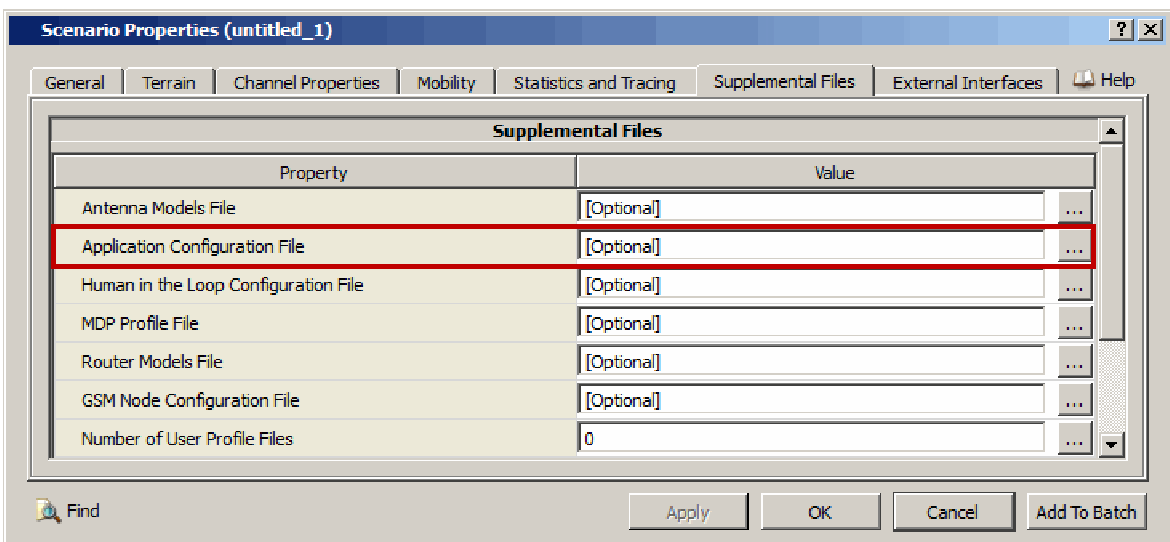


FIGURE 4-23. Specifying the Antenna Models File

TABLE 4-44. Command Line Equivalent of Antenna Models File Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Antenna Models File	Global	`ANTENNA-MODEL-CONFIG-FILE`

Configuring Antenna Model for an Interface

To configure the antenna model for an interface, do the following:

1. Go to one of the following locations:
   - To set properties for a specific subnet, go to **Wireless Subnet Properties Editor > Physical Layer**.
   - To set properties for a specific interface of a node, go to one of the following locations:
     - **Interface Properties Editor > Interfaces > Interface # > Physical Layer**
     - **Default Device Properties Editor > Interfaces > Interface # > Physical Layer**.

   In this section, we show how to configure the antenna model in the Wireless Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. To specify a standard antenna model, set **Specify Antenna Model From File** to *No and* set the antenna parameters listed in Table 4-45. *The available antenna models are described in* Table 4-43. Set the dependent parameters for the selected antenna model. See the model library referenced in Table 4-43 for details.

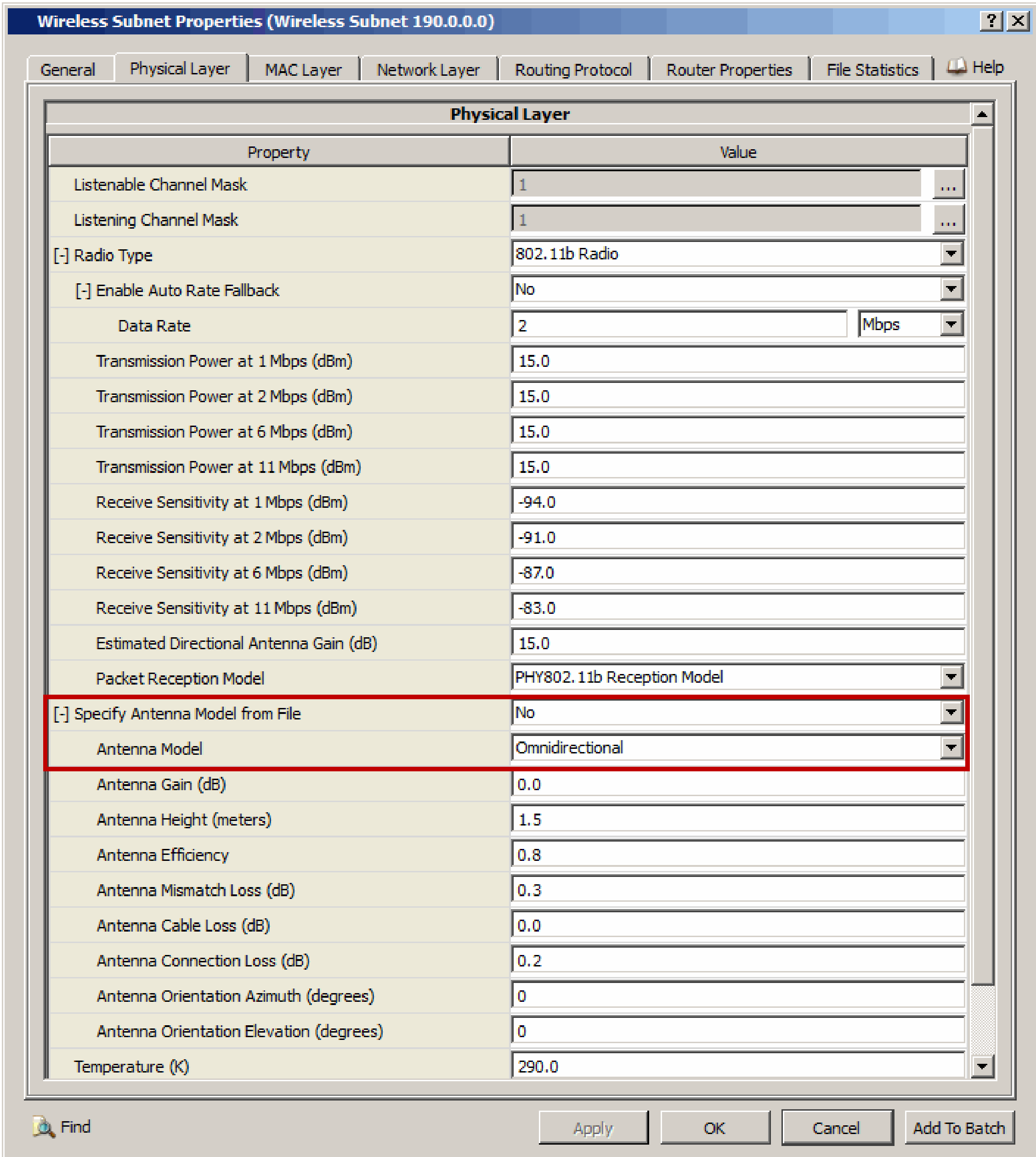


FIGURE 4-24. Specifying a Standard Antenna Model

TABLE 4-45. Command Line Equivalent of Standard Antenna Model Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Antenna Model	Subnet, Interface	`ANTENNA-MODEL`

3. You can specify a custom antenna model if you have specified an Antenna Models file or imported or created custom antenna models using the Antenna Model Editor (refer to *Wireless Model Library*).

   To specify a custom antenna model, set **Specify Antenna Model From File** to *Yes* and select a value for the parameter **Antenna Model** from the list. The names of all imported or created antenna models appear in the list.

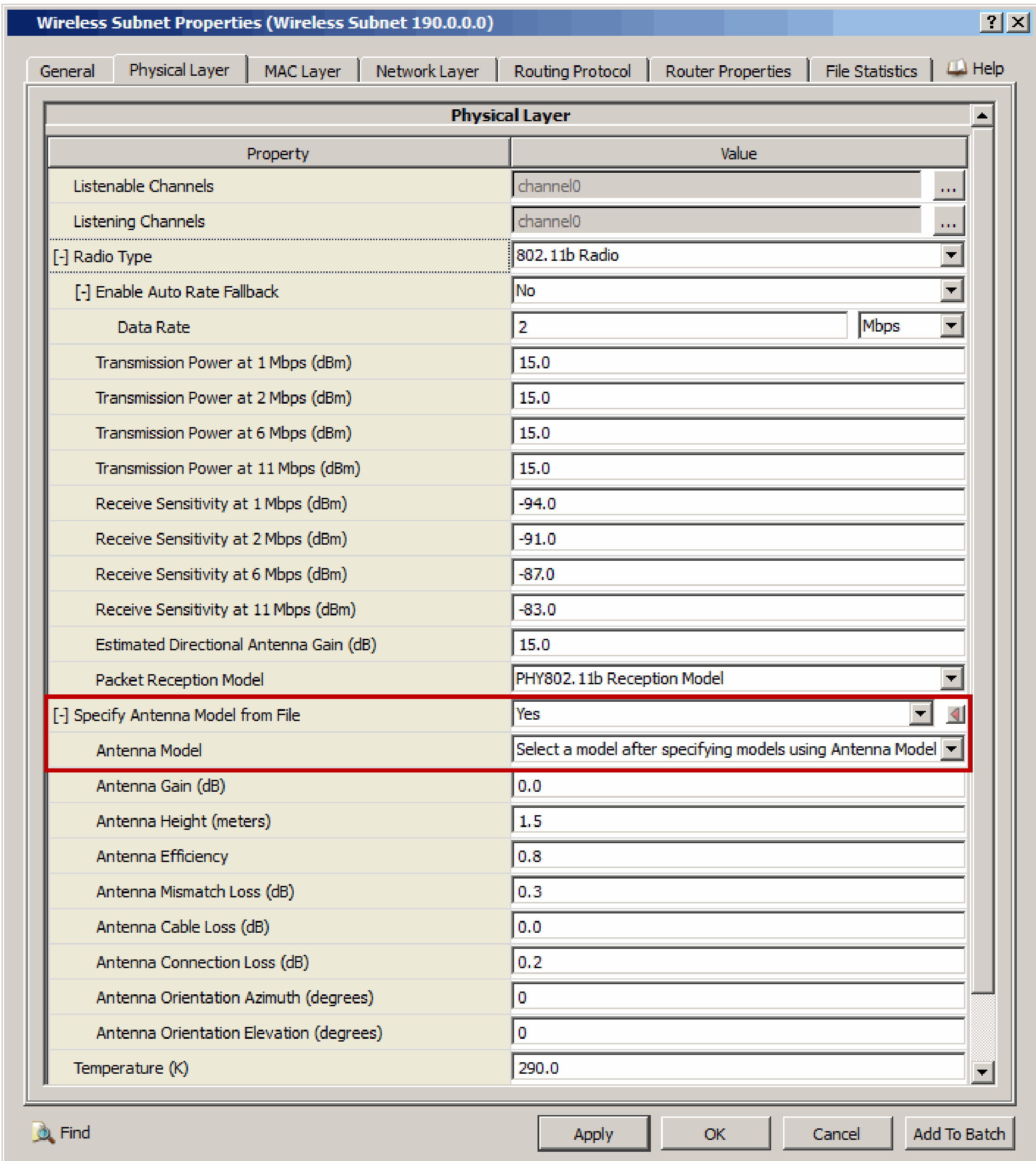


FIGURE 4-25. Specifying a Custom Antenna Model

TABLE 4-46. Command Line Equivalent of Custom Antenna Model Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Antenna Model	Subnet, Interface	ANTENNA-MODEL

4.2.8.1.4 Thermal Noise Settings

The noise power (in watts) for the physical radio model is computed as the product `T*k*B*f`, where, `T` is the temperature in Kelvin, `k` is the Boltzmann constant (= 1.381 x 10^{-23} Joules/Kelvin), `B` is the bandwidth in Hz, and `f` is a constant called the noise factor. Parameters `T` and `f` can be specified by the user.

4.2.8.1.4.1 Command Line Configuration

To set the thermal noise parameters for the command line interface, include the parameters listed in Table 4-47 in the scenario configuration (.config) file.

TABLE 4-47. Thermal Noise Parameters

Parameter	Value	Description
PHY-TEMPERATURE *Dependency:* Scenario is a wireless scenario *Optional* *Scope:* All	Real *Range:* ≥ 0.0 *Default:* 290.0 *Unit:* °K	Ambient temperature used in the calculation of the thermal noise level.
PHY-NOISE-FACTOR *Dependency:* Scenario is a wireless scenario *Optional* *Scope:* All	Real *Range:* ≥ 0.0 *Default:* 10.0	Noise factor used in the calculation of the thermal noise level.

4.2.8.1.4.2 GUI Configuration

To set the thermal noise parameters in the GUI, perform the following steps:

1. Go to one of the following locations:
   - To set properties for a specific subnet, go to **Wireless Subnet Properties Editor > Physical Layer**.
   - To set properties for a specific interface of a node, go to one of the following locations:
     - **Interface Properties Editor > Interfaces > Interface # > Physical Layer.**
     - **Default Device Properties Editor > Interfaces > Interface # > Physical Layer**.

   In this section, we show how to configure the thermal noise parameters in the Wireless Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. Set the thermal noise parameters listed in Table 4-48.

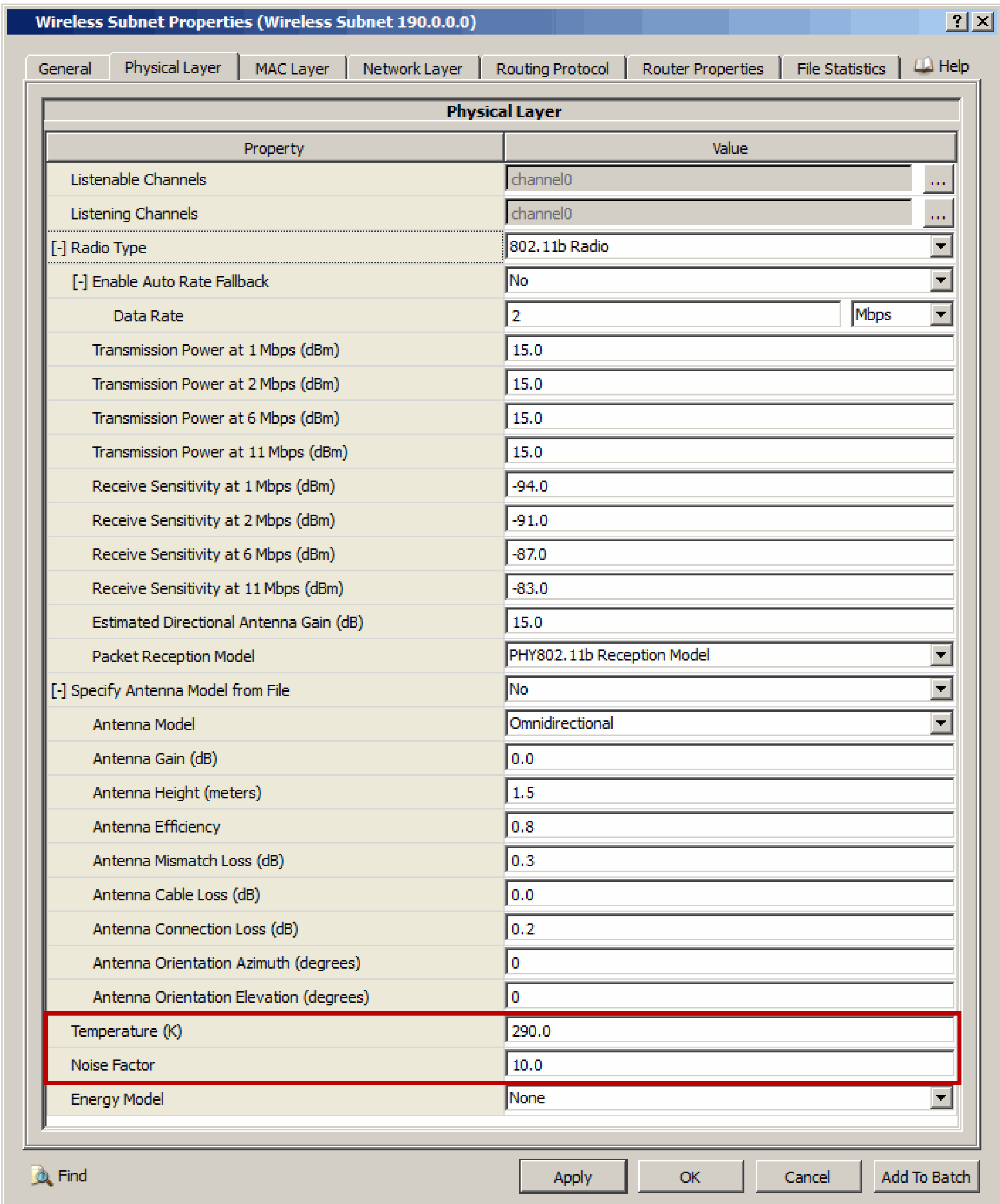


FIGURE 4-26. Setting Thermal Noise Parameters

TABLE 4-48. Command Line Equivalent of Thermal Noise Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Temperature	Subnet, Interface	`PHY-TEMPERATURE`
Noise Factor	Subnet, Interface	`PHY-NOISE-FACTOR`

4.2.8.1.5 Radio Energy Models

A radio energy model computes the energy consumed in transmitter and receiver circuitry (baseband circuits and circuits of power amplifier) and power amplifier of the transmitter in the various power state functions of the radio (primarily transmit, receive, idle, and sleep modes).

4.2.8.1.5.1 Command Line Configuration

To configure the radio energy model for the command line interface, include the parameters listed in Table 4-49 in the scenario configuration (.config) file.

TABLE 4-49. Radio Energy Model Parameters

Parameter	Value	Description
`ENERGY-MODEL` *Dependency:* Scenario is a wireless scenario *Optional* *Scope:* All	List: • `NONE` • `GENERIC` • `MICA-MOTES` • `MICAZ` • `USER-DEFINED` *Default:* `NONE`	Energy model. If this parameter is set to `NONE`, no energy model is used at the interface. See Table 4-50 for a description of the energy models.

Table 4-50 describes the different radio energy models. See the corresponding model library for a detailed description of each model and its parameters.

TABLE 4-50. Radio Energy Models

Command Line Name	GUI Name	Description	Model Library
`GENERIC`	Generic	This is a generic radio energy model that computes power consumption of the radio in different power modes and for variable transmission power.	Wireless
`MICA-MOTES`	Mica Motes	This is a radio-specific energy model which is pre-configured with the specification of power consumption of Mica motes (embedded sensor nodes).	Wireless
`MICAZ`	MicaZ	This is a radio-specific energy model which is pre-configured with the specification of power consumption of MicaZ motes (embedded sensor nodes).	Wireless
`USER-DEFINED`	User Specified	This radio energy model allows the user to specify the energy consumption of the radio in different power modes.	Wireless

4.2.8.1.5.2 GUI Configuration

To configure the radio energy model in the GUI, perform the following steps:

1. Go to one of the following locations:
    - To set properties for a specific subnet, go to **Wireless Subnet Properties Editor > Physical Layer**.
    - To set properties for a specific interface of a node, go to one of the following locations:
        - **Interface Properties Editor > Interfaces > Interface # > Physical Layer**
        - **Default Device Properties Editor > Interfaces > Interface # > Physical Layer**.

In this section, we show how to configure the radio energy model in the Wireless Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. Select the radio energy model by setting **Energy Model** to the desired value. The available radio energy models are described in Table 4-50.

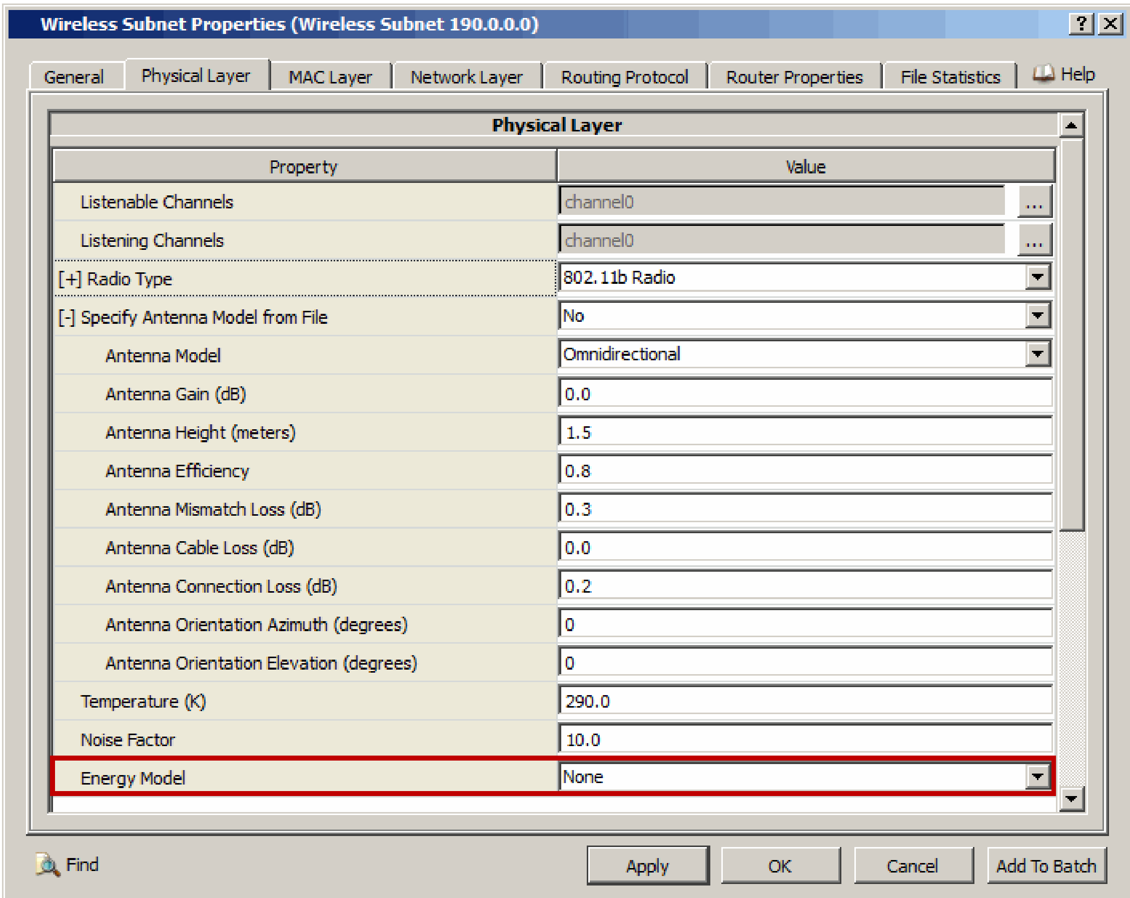


FIGURE 4-27. Specifying Radio Energy Model

TABLE 4-51. Command Line Equivalent of Energy Model Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Energy Model	Subnet, Interface	`ENERGY-MODEL`

3. Set the dependent parameters for the selected radio energy model. See the model library referenced in Table 4-50 for details.

4.2.8.2 MAC Layer

The Medium Access Control (MAC) protocol controls access to a channel to enable multiple devices to share the same channel.

4.2.8.2.1 Command Line Configuration

To configure the MAC protocol for the command line interface, include the parameters listed in Table 4-52 in the scenario configuration (.config) file.

TABLE 4-52. MAC Layer Parameters

Parameter	Value	Description
`MAC-PROTOCOL` *Dependency:* Interface is created by the `SUBNET` keyword *Required* *Scope:* All	List: • `ALOHA` • `ANE` • `CELLULAR-MAC` • `CSMA` • `GENERICMAC` • `GSM` • `MAC-LTE` • `MAC-WORMHOLE` • `MAC802.3` • `MAC802.15.4` • `MAC802.16` • `MACA` • `MACDOT11` • `MACDOT11e` • `SATCOM` • `SATELLITE-BENTPIPE` • `SWITCHED-ETHERNET` • `TDMA` • `USAP`	MAC protocol running at the subnet interface. See Table 4-53 for a description of MAC protocols.
`LINK-MAC-PROTOCOL` *Dependency:* Interface is created by the `LINK` or `ATM-LINK` keyword *Optional* *Scope:* All	List: • `MAC802.3` • `ABSTRACT` *Default:* `ABSTRACT`	MAC protocol running at the link interface. **Note:** `LINK-MAC-PROTOCOL` can be set to `MAC802.3` only for a wired link, i.e., if `LINK-PHY-TYPE` is `WIRED` (see Section 4.2.5.1.2). See Table 4-53 for a description of MAC protocols.
`MAC-PROPAGATION-DELAY` *Optional* *Scope:* All	Time *Range:* ≥ `0S` *Default:* `1US`	Average propagation delay in a wireless subnet. This is usually used by wireless MAC protocols as an estimate of the actual propagation delay. For example, a node that transmits a RTS waits for at least 2 * `MAC-PROPAGATION-DELAY` to receive the CTS.

TABLE 4-52. MAC Layer Parameters (Continued)

Parameter	Value	Description
PROMISCUOUS-MODE *Optional* *Scope:* All	List: • YES • NO *Default:* NO	Indication whether the MAC Layer should pass received packets not addressed to the node to the higher layers.
LLC-ENABLED *Optional* *Scope:* All	List: • YES • NO *Default:* NO	Indication whether the Logical Link Control (LLC) protocol is enabled at the node. Refer to *Developer Model Library* for details.

Table 4-53 describes the different MAC protocol models for wireless subnets in EXata. See the corresponding model library for a detailed description of each protocol and its parameters.

TABLE 4-53. MAC Protocols for Wireless Subnets

Command Line Name	GUI Name	Description	Model Library
ANE	Abstract Network Equation (ANE) Satellite	Models the Abstract Satellite Equation MAC protocol for satellites.	Wireless
ALOHA	Aloha	Models the Aloha MAC protocol.	Wireless
CELLULAR-MAC	Cellular MAC	Indicates that a cellular system MAC protocol is to be used. When this option is selected, the MAC protocol for the cellular system should be specified by using the parameter CELLULAR-MAC-PROTOCOL.	Cellular
CSMA	CSMA	Models the Carrier Sense Multiple Access (CSMA) MAC protocol.	Wireless
GENERICMAC	Generic MAC	Models an abstract wireless MAC protocol.	Wireless
GSM	GSM	Models the GSM MAC Layer.	Cellular
MAC-LTE	LTE	Models the LTE Layer 2.	LTE
MAC-WORMHOLE	Wormhole	Models the MAC protocol used in the Worm Hole adversary model.	Cyber
MAC802.15.4	802.15.4	Models the IEEE 802.15.4 MAC (ZigBee MAC) specification.	Sensor Networks
MAC802.16	802.16	Models the IEEE 802.16 MAC (WiMAX MAC) specification.	Advanced Wireless
MACA	MACA	Models the Multiple Access with Collision Avoidance (MACA) MAC protocol.	Wireless
MACDOT11	802.11	Models the IEEE 802.11 MAC specification. In addition, models the IEEE 802.11 MAC specification if 802.11 MAC is configured in infrastructure mode and the parameter MAC-DOT11s-MESH-POINT is set to YES for some node or interface in the subnet.	Wireless

TABLE 4-53. MAC Protocols for Wireless Subnets (Continued)

Command Line Name	GUI Name	Description	Model Library
MACDOT11e	802.11e	Models the IEEE 802.11e MAC specification. This is a QoS enhancement to the IEEE 802.11 MAC. In addition, models the IEEE 802.11n MAC specification if PHY-MODEL is set to PHY-802.11n.	Wireless
SATCOM	Abstract Satellite (SATCOM) MAC	Models an abstract MAC protocol for satellites.	Developer
SATELLITE-BENTPIPE	Satellite-RSV MAC	Models the bentpipe MAC protocol for satellites. This MAC protocol is used with the Satellite-RSV PHY model.	Wireless
TDMA	TDMA	Models the Time Division Multiple Access (TDMA) MAC protocol.	Wireless
USAP	USAP	Models the Unifying Slot Assignment Protocol (USAP).	Wireless

Table 4-54 describes the different MAC protocol models for wired subnets in EXata.

TABLE 4-54. MAC Protocols for Wired Subnets

Command Line Name	GUI Name	Description	Model Library
MAC802.3	802.3	Models the IEEE 802.3 MAC specification.	Developer
SWITCHED-ETHERNET	Switched Ethernet	Models an abstract switch connecting a subnet. This model does not have detailed models of switch ports, etc., and is limited to one subnet.	Multimedia and Enterprise

Table 4-55 describes the different MAC protocol models for point-to-point links in EXata.

TABLE 4-55. MAC Protocols for Point-to-point Links

Command Line Name	GUI Name	Description	Model Library
ABSTRACT	Abstract Link MAC	Models the abstract MAC protocol for point-to-point links.	Developer
MAC802.3	802.3	Models the IEEE 802.3 MAC specification.	Developer

4.2.8.2.2 GUI Configuration for Wireless Subnets

To configure the MAC Layer parameters for wireless subnets in the GUI, do the following:

1. Go to one of the following locations:
   - To set properties for a specific subnet, go to **Wireless Subnet Properties Editor > MAC Layer**.
   - To set properties for a specific interface of a node, go to one of the following locations:
     - **Interface Properties Editor > Interfaces > Interface # > MAC Layer**.
     - **Default Device Properties Editor > Interfaces > Interface # > MAC Layer**.

   In this section, we show how to configure the wireless subnet MAC properties in the Wireless Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. Set the MAC Layer parameters listed in Table 4-56. The available MAC protocols for wireless subnets are listed in Table 4-53.

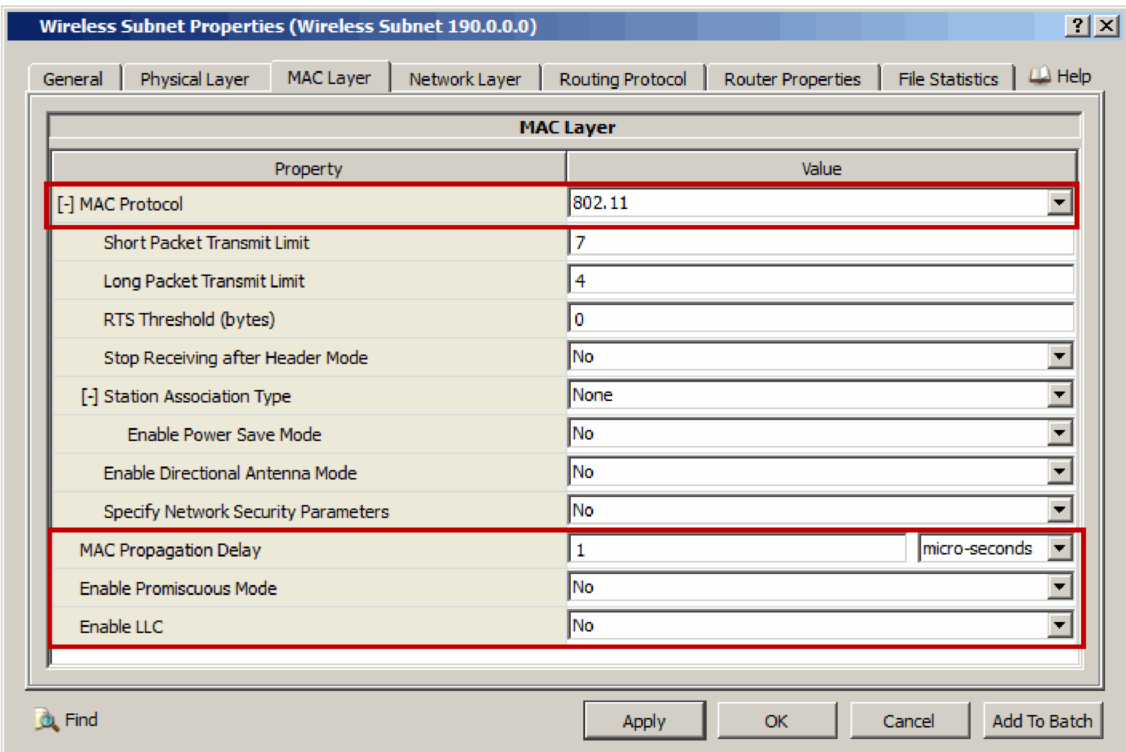


FIGURE 4-28. Specifying MAC Parameters for a Wireless Subnet

TABLE 4-56. Command Line Equivalent of Wireless MAC Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
MAC Protocol	Wireless Subnet, Interface	`MAC-PROTOCOL`
MAC Propagation Delay	Wireless Subnet, Interface	`MAC-PROPAGATION-PROTOCOL`
Enable Promiscuous Mode	Wireless Subnet, Interface	`PROMISCUOUS-MODE`
Enable LLC	Wireless Subnet, Interface	`LLC-ENABLED`

3. Set the dependent parameters for the selected MAC protocol. See the model library referenced in Table 4-53 for details.

4.2.8.2.3 GUI Configuration for Wired Subnets

To configure the MAC Layer parameters for wired subnets in the GUI, do the following:

1. Go to one of the following locations:

- To set properties for a specific subnet, go to **Wired Subnet Properties Editor > General**.
- To set properties for a specific interface of a node, go to one of the following locations:
  - **Interface Properties Editor > Interfaces > Interface # > MAC Layer**
  - **Default Device Properties Editor > Interfaces > Interface # > MAC Layer**.

In this section, we show how to configure the wireless subnet MAC properties in the Wired Subnet Properties Editor. Parameters can be set in the other properties editors in a similar way.

2. Set the MAC Layer parameters listed in Table 4-57. The available MAC protocols are listed in Table 4-54.

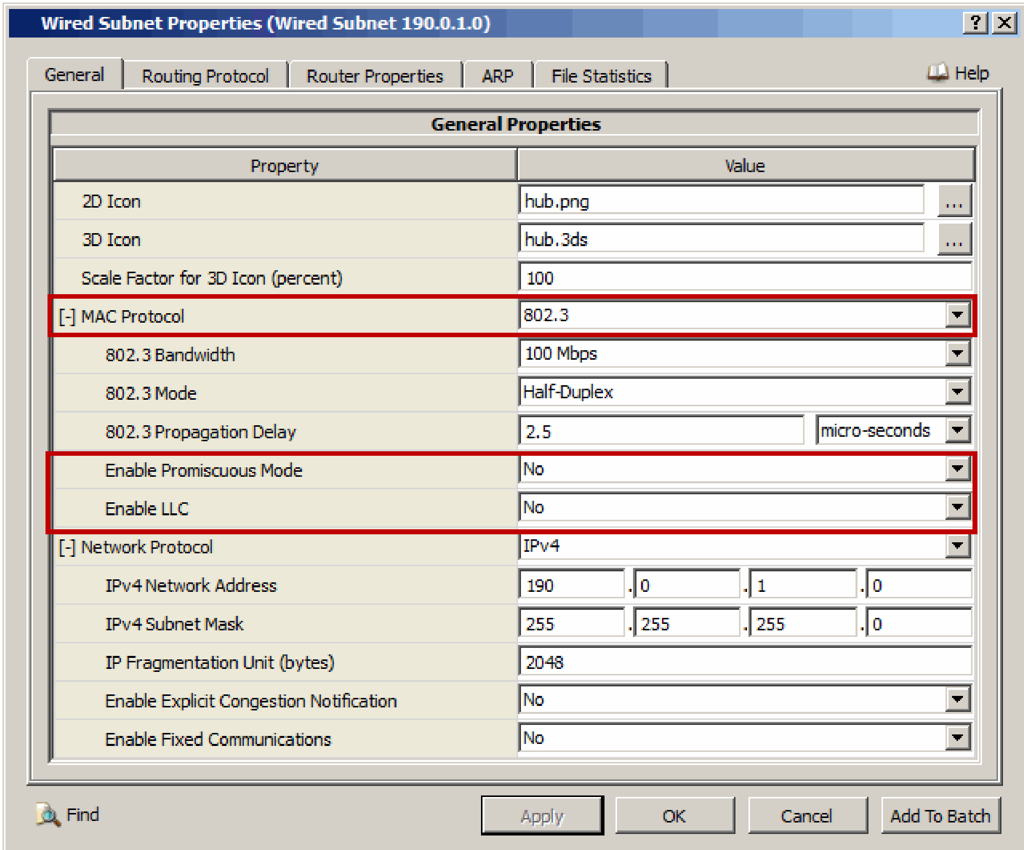


FIGURE 4-29. Specifying MAC Parameters for a Wired Subnet

TABLE 4-57. Command Line Equivalent of Wired MAC Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
MAC Protocol	Wired Subnet, Interface	`MAC-PROTOCOL`
Enable Promiscuous Mode	Wired Subnet, Interface	`PROMISCUOUS-MODE`
Enable LLC	Wired Subnet, Interface	`LLC-ENABLED`

3. Set the dependent parameters for the selected MAC protocol. See the model library referenced in Table 4-54 for details.

4.2.8.2.4 GUI Configuration for Point-to-point Links

To configure the MAC Layer parameters for point-to-point links in the GUI, do the following:

1. Go to **Point-to-point Link Properties Editor > General**.
2. Set the MAC Layer parameters listed in Table 4-58. The available MAC protocols are listed in Table 4-55.

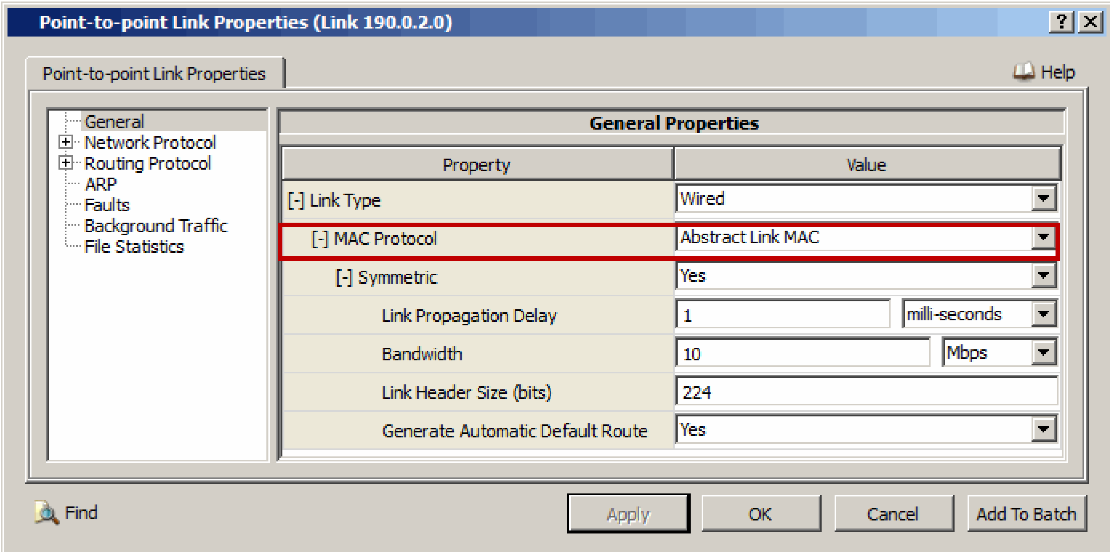


FIGURE 4-30. Specifying MAC Parameters for a Point-to-point Link

TABLE 4-58. Command Line Equivalent of Point-to-point Link MAC Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
MAC Protocol	Point-to-point Link	`LINK-MAC-PROTOCOL`

3. Set the dependent parameters for the selected MAC protocol. See the model library referenced in Table 4-55 for details.

4.2.8.3 Network Layer

The Network Layer provides the functionality of transferring variable length data sequences from a source to a destination via one or more networks while maintaining the quality of service requested by the Transport Layer.

4.2.8.3.1 General Network Layer Parameters

General Network Layer parameters include specification of the network and related protocols, parameter to enable forwarding of packets, and parameters to assign IP addresses and subnet masks to interfaces.

4.2.8.3.1.1 Command Line Configuration

To configure the general Network Layer parameters for the command line interface, include the parameters listed in Table 4-59 in the scenario configuration (.config) file.

TABLE 4-59. General Network Layer Parameters

Parameter	Value	Description
NETWORK-PROTOCOL *Required* *Scope:* All *Instances:* interface-number	List: • CELLULAR-LAYER3 • DUAL-IP • GSM-LAYER3 • IP • IPv6	Network protocol in use at the node. See Table 4-60 for a description of network protocols. **Note:** This parameter can have instances (which correspond to interfaces) if it is qualified by a node ID. Interfaces of a node are numbered in the order in which they are created by LINK and SUBNET statements in the scenario configuration (.config) file.
IP-ENABLE-LOOPBACK *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* YES	Indication whether the loopback function is enabled for the node.
IP-LOOPBACK-ADDRESS *Dependency:* IP-ENABLE-LOOPBACK = YES *Required* *Scope:* Global, Node	IPv4 Address	IPv4 loopback address of the node.
IP-FRAGMENTATION-UNIT *Optional* *Scope:* All	Integer *Range:* [256, 65535] *Default:* 2048 *Unit:* bytes	Maximum size of an IP fragment.
ECN *Optional* *Scope:* All	List: • YES • NO *Default:* NO	Indication whether Explicit Congestion Notification (ECN) is enabled. **Note:** ECN is effective only in combination with an active queue management policy, such as Random Early Detection (RED).
ICMP *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* NO	Indication whether the Internet Control Message protocol (ICMP) is in use at a node. See *Developer Model Library* for details of configuring ICMP.
IP-ADDRESS *Optional* (See note below table)	IPv4 Address	IPv4 address to be assigned to the interface. **Note:** This parameter must be specified in the format described below the table.

TABLE 4-59. General Network Layer Parameters (Continued)

Parameter	Value	Description
`IP-SUBNET-MASK` *Optional* (See note below table)	IPv4 Address	IPv4 subnet mask to be assigned to the interface. **Note:** This parameter must be specified in the format described below the table.
`IPV6-ADDRESS` *Optional* (See note below table)	IPv6 Address	IPv6 address to be assigned to the interface. **Note:** This parameter must be specified in the format described below the table.
`IPV6-PREFIX-LEN` *Optional* (See note below table)	Integer *Range:* `[0, 128]`	Length of the prefix of the IPv6 interface address. **Note:** This parameter must be specified in the format described below the table.
`UNNUMBERED` *Optional* *Scope:* Global, Node *Instances:* interface-number	List: • `YES` • `NO` *Default:* `NO`	Configures the interface as an unnumbered interface. **Note:** The 0^{th} interface of a node can not be configured as an unnumbered interface. **Note:** If any of the interfaces of a node is configured as an unnumbered interface, then the Address Resolution Protocol (ARP) and Logical Link Control (LLC) protocol should also be configured for the node. See *Developer Model Library* for details of the ARP and LLC models.

Note: Parameters `IP-ADDRESS`, `IP-SUBNET-MASK`, `IP-ADDRESS-IPv6`, and `IPV6-PREFIX-LEN` must be specified using the following format:

```
<Node ID> <Parameter Name> [<Index>] <Parameter Value>
```

where

`<Node ID>`	Node identifier to which this parameter declaration is applicable, enclosed in square brackets.
`<Parameter Name>`	Name of the parameter (`IP-ADDRESS`, `IP-SUBNET-MASK`, `IP-ADDRESS-IPv6`, or `IPV6-PREFIX-LEN`).
`<Index>`	Interface index 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 interfaces. The interface index is optional. If the interface index is not included, then the parameter declaration is applicable to the 0^{th} interface.
`<Parameter Value>`	Value of the parameter.

Table 4-60 describes the different network protocol models in EXata. See the corresponding model library for a detailed description of each protocol and its parameters.

TABLE 4-60. Network Protocols

Command Line Name	GUI Name	Description	Model Library
`CELLULAR-LAYER3`	Cellular Layer 3	Indicates that a cellular system Network Layer model is to be used. When this option is selected, the Network Layer model for the cellular system should be specified by using the parameter `CELLULAR-LAYER3-PROTOCOL`.	Cellular
`GSM-LAYER3`	GSM Layer 3	Models the GSM Network Layer.	Cellular
`DUAL-IP`	Dual-IP	Indicates that both IPv4 and IPv6 are used at the node. When this option is selected, the Network Layer can handle both IPv4 and IPv6 packets.	Developer
`IP`	IPv4	Models the Internet Protocol version 4 (IPv4) specified in RFC 791.	Developer
`IPv6`	IPv6	Models the Internet Protocol version 6 (IPv6) specified in RFC 2460.	Developer

4.2.8.3.1.2 GUI Configuration

To configure the general Network Layer parameters in the GUI, perform the following steps:

1. Go to one of the following locations:
   - To set properties for a specific wireless subnet, go to **Wireless Subnet Properties Editor > Network Layer > General**.
   - To set properties for a specific wired subnet, go to **Wired Subnet Properties Editor > General**.
   - To set properties for a specific point-to-point link, go to **Point-to-point Link Properties Editor > Point-to-point Link Properties > Network Protocol**.
   - To set properties for a specific node, go to **Default Device Properties Editor > Node Configuration > Network Layer**.
   - To set properties for a specific interface of a node, go to one of the following locations:
     - **Interface Properties Editor > Interfaces > Interface # > Network Layer**
     - **Default Device Properties Editor > Interfaces > Interface # > Network Layer**.

2. Set the Network Layer parameters listed in Table 4-61. The available network protocols are listed in Table 4-60.

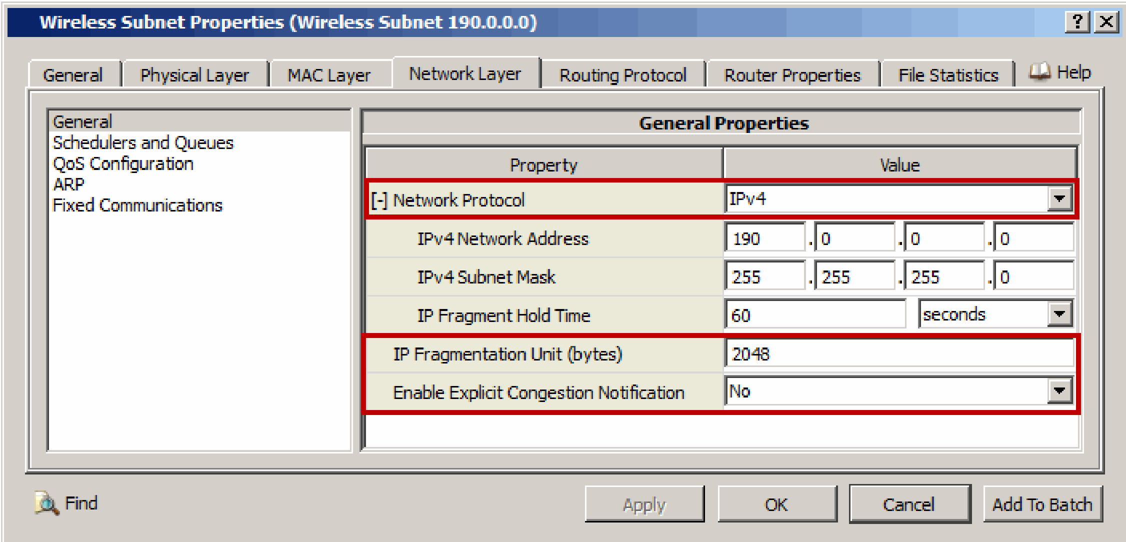


FIGURE 4-31. General Network Layer Parameters for a Wireless Subnet

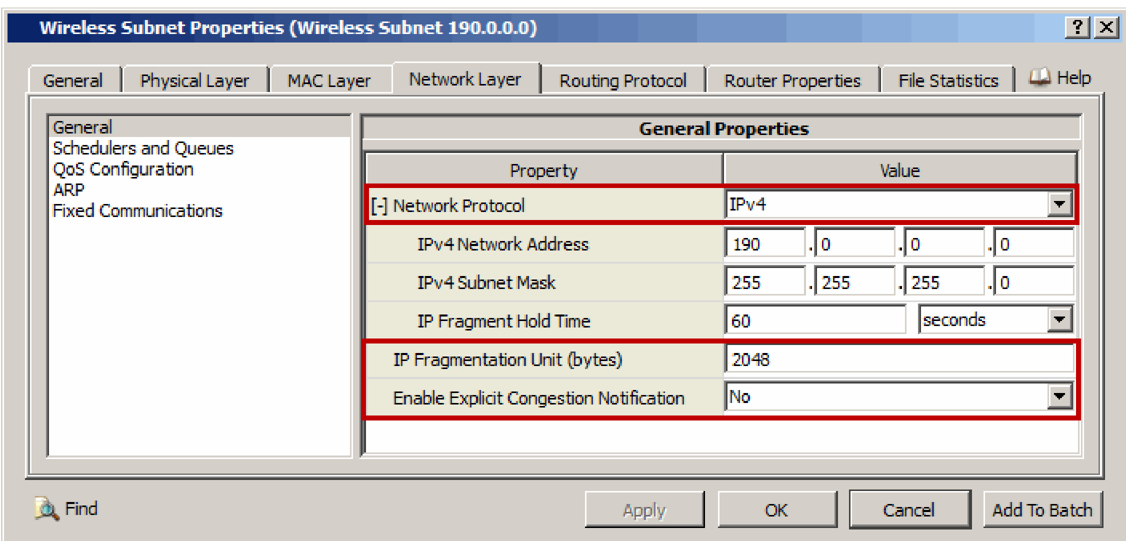


FIGURE 4-32. General Network Layer Parameters for a Wired Subnet

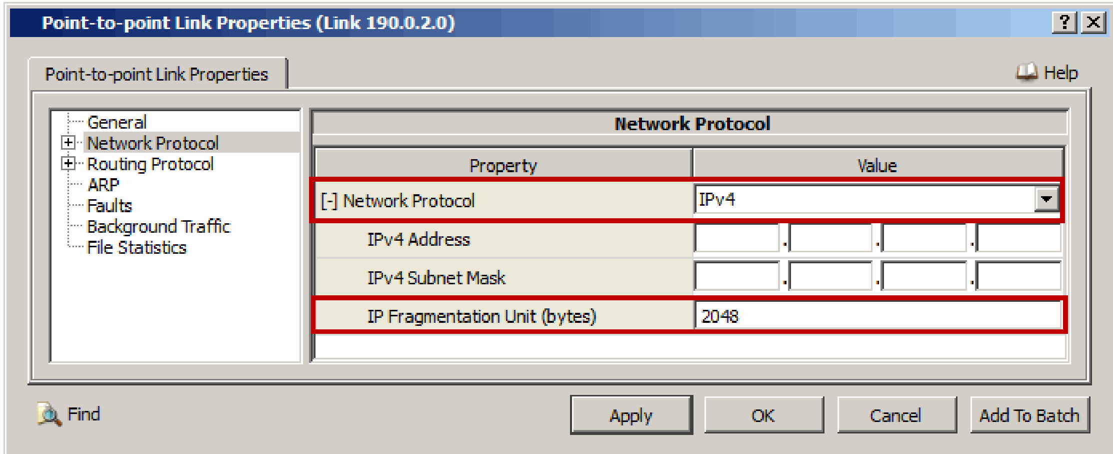


FIGURE 4-33. General Network Layer Parameters for a Point-to-Point Link

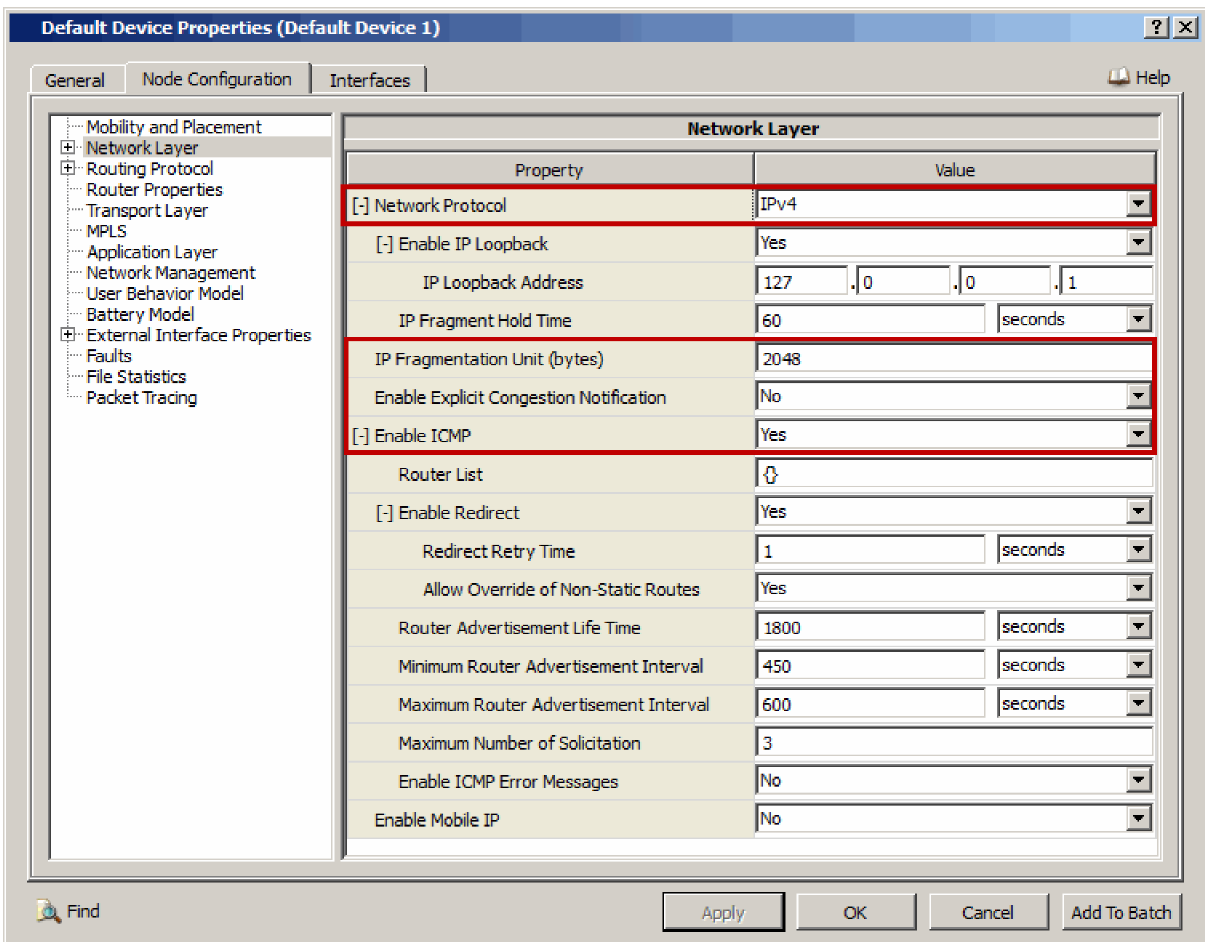


FIGURE 4-34. General Network Layer Parameters for a Node

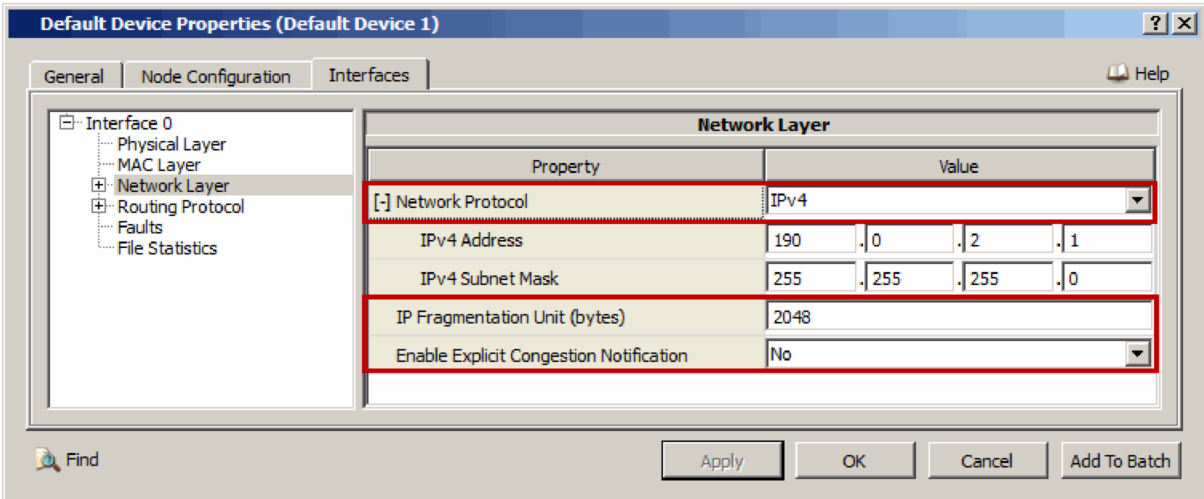


FIGURE 4-35. General Network Layer Parameters for an Interface

TABLE 4-61. Command Line Equivalent of General Network Layer Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Network Protocol	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`NETWORK-PROTOCOL`
IP Fragmentation Unit	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`IP-FRAGMENTATION-UNIT`
Enable Explicit Congestion Notification	Node, Wireless Subnet, Wired Subnet, Interface	`ECN`
Enable ICMP	Node	`ICMP`

3. When a subnet or point-point link is created, the GUI assigns default IP address(es) and IPv4 subnet mask to the subnet. To overwrite the default IP address(es) and subnet mask assigned to a subnet or link, do the following:

- If **Network Protocol** is set to *IPv4*, then set **IPv4 Network Address** and **IPv4 Subnet Mask**.
- If **Network Protocol** is set to *IPv6*, then set **IPv6 Network Address** and **IPv6 Prefix Length**.
- If **Network Protocol** is set to *Dual-IP*, then set **IPv4 Network Address**, **IPv4 Subnet Mask**, **IPv6 Network Address**, and **IPv6 Prefix Length**.

Figure 4-36 shows the subnet address parameters in the Wireless Subnet Properties Editor when **Network Protocol** has been set to *Dual-IP*.

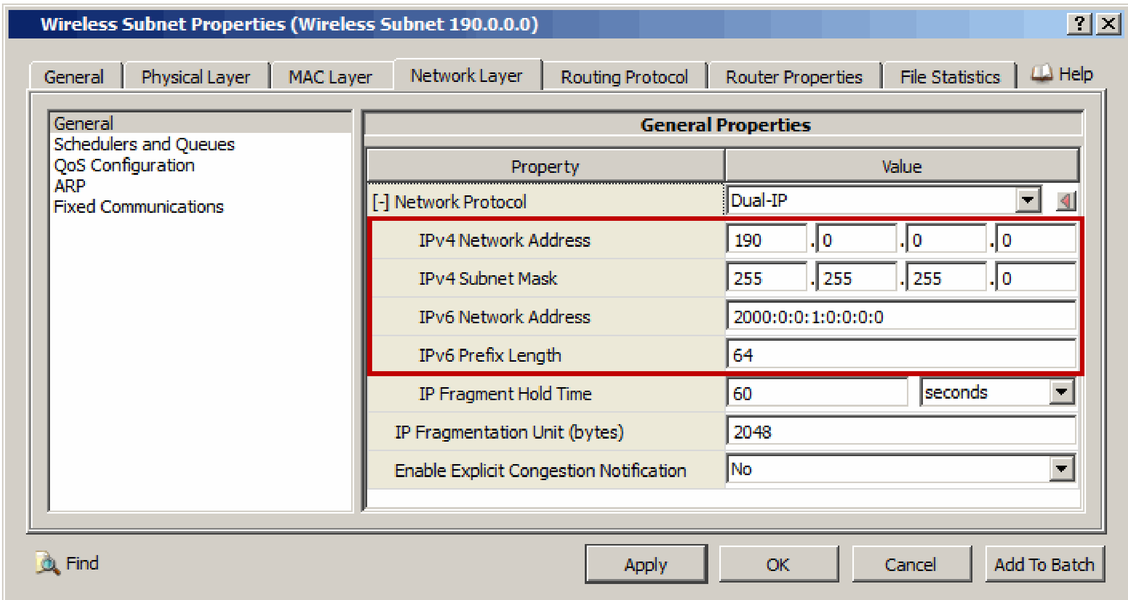


FIGURE 4-36. Assigning IP Address(es) and Subnet Mask to a Wireless Subnet

Note: These parameters correspond to the argument(s) of the `SUBNET` and `LINK` keywords in Command Line Interface (see Section 4.2.5).

4. Architect assigns default IP address(es) and IPv4 subnet mask each interface based on the IP address(es) and IPv4 subnet mask of the subnet of which the interface is a part. To overwrite the default IP address(es) and subnet mask assigned to an interface, set the appropriate parameters listed in Table 4-62.

Figure 4-37 shows the interface address parameters in the Interface Properties Editor when **Network Protocol** has been set to *Dual-IP*.

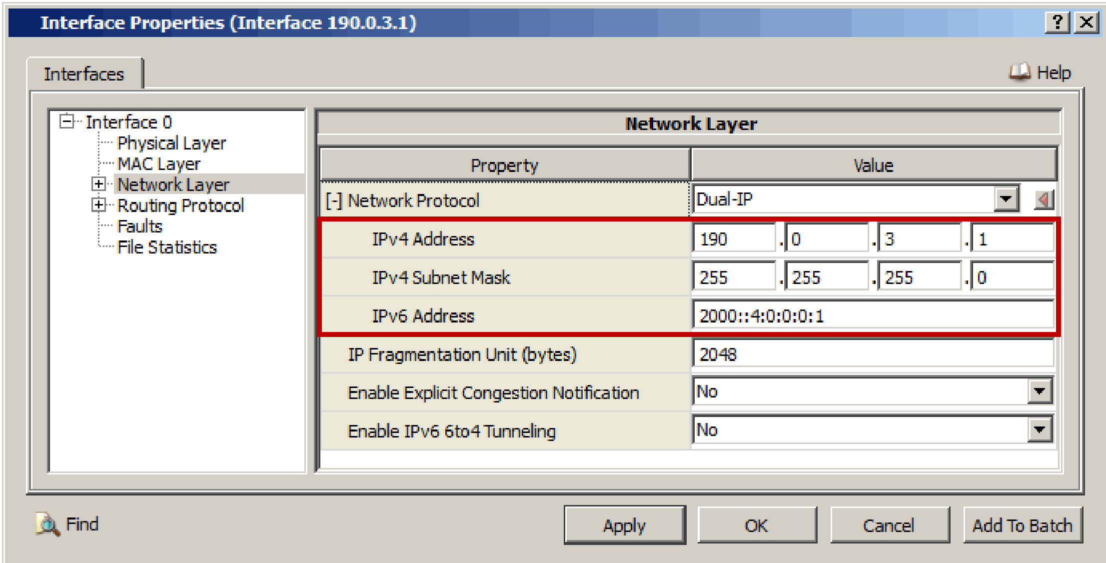


FIGURE 4-37. Assigning IP Address(es) and Subnet Mask to an Interface

TABLE 4-62. Command Line Equivalent of Interface Address Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
IPv4 Address	Interface	`IP-ADDRESS`
IPv4 Mask	Interface	`IP-SUBNET-MASK`
IPv6 Address	Interface	`IP-ADDRESS-IPv6`

5. To enable the loopback function for an IPv4 or dual IP node, set **Enable IP Loopback** to *Yes* and set the dependent parameters listed in Table 4-63.

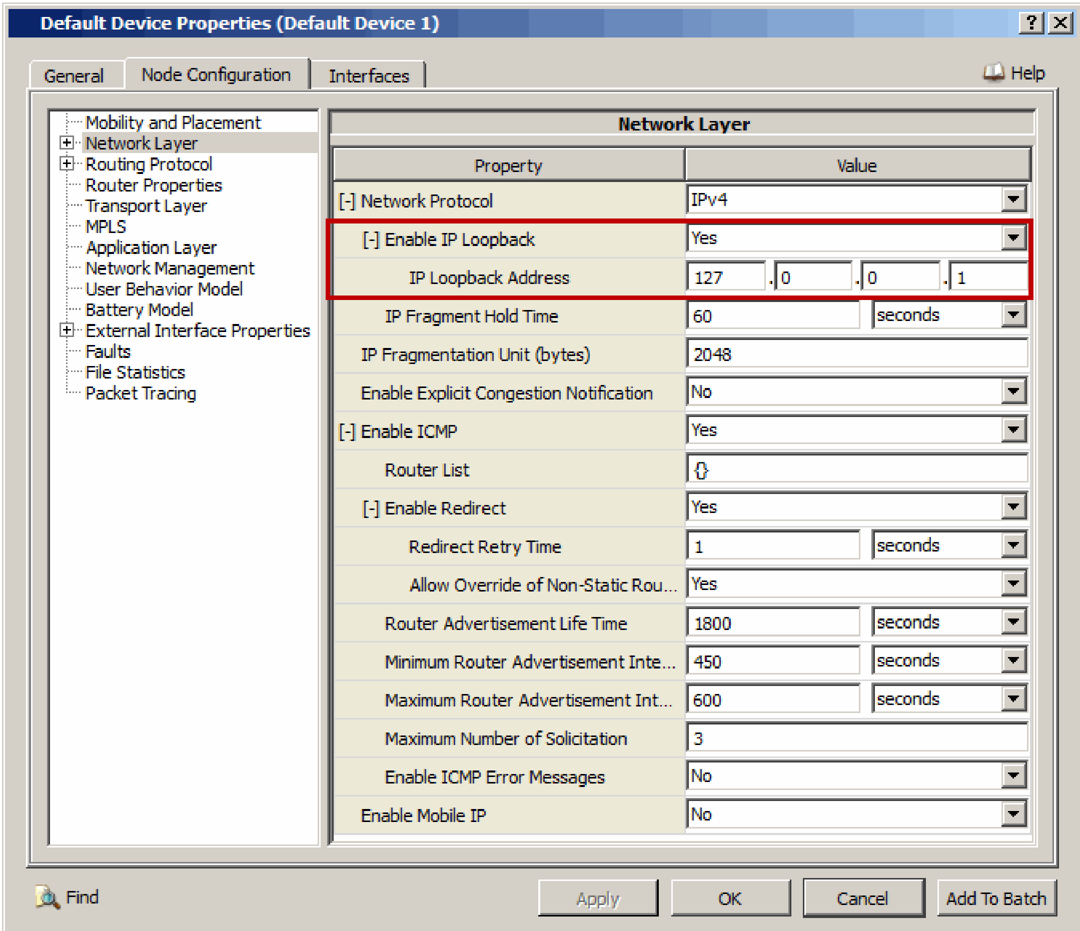


FIGURE 4-38. Enabling IP Loopback for a Node

TABLE 4-63. Command Line Equivalent of IP Loopback Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Enable IP Loopback	Node	`IP-ENABLE-LOOPBACK`
IP Loopback Address	Node	`IP-LOOPBACK-ADDRESS`

6. To configure an interface as an unnumbered interface in the GUI, set **Configure as Unnumbered Interface** to *Yes*.

Note: The 0^{th} interface of a node can not be configured as an unnumbered interface.

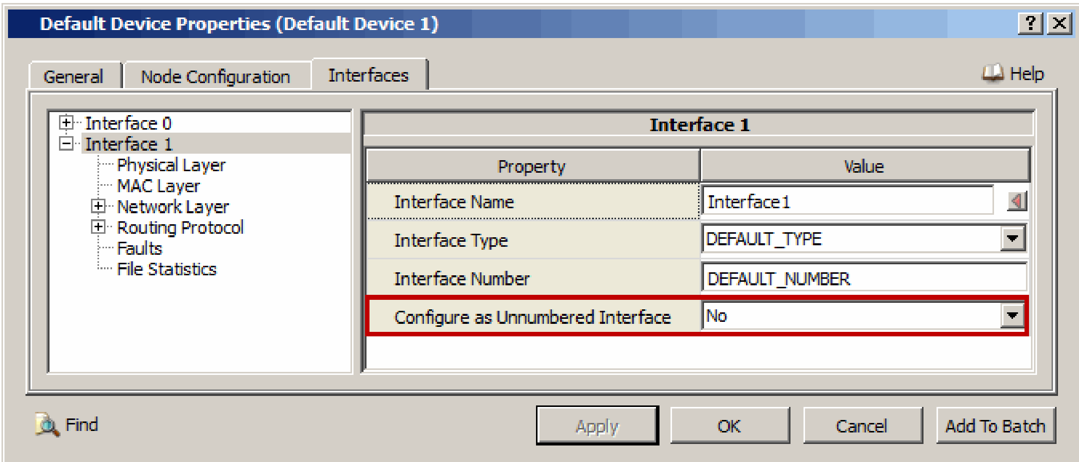


FIGURE 4-39. Configuring an Unnumbered Interface

TABLE 4-64. Command Line Equivalent of Unnumbered Interface Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Configure as Unnumbered Interface	Interface	UNNUMBERED

4.2.8.3.2 Unicast Routing

A routing protocol determines the path of a packet from the source to the destination. To forward a packet, the network protocol needs to know the next node in the path as well as the outgoing interface on which to send the packet. A routing protocol computes such routing information.

In general, routing protocols can be divided into two categories: proactive routing protocols and on-demand routing protocols. A proactive routing protocol discovers the network topology and computes the routing information regardless of whether the network protocol has a packet which needs that information. An on-demand routing protocol tries to discover a path to a destination only when the network protocol receives a packet addressed to that destination.

4.2.8.3.2.1 Command Line Configuration

To configure the unicast routing parameters for the command line interface, include the parameters listed in Table 4-65 in the scenario configuration (.config) file.

TABLE 4-65. Unicast Routing Parameters

Parameter	Value	Description
ROUTING-PROTOCOL *Optional* *Scope:* All	List: • NONE • ANODR • AODV • BELLMANFORD • DSR • DYMO • EIGRP • FISHEYE • FSRL • IARP • IERP • IGRP • LAR1 • OLSR-INRIA • OLSRv2-NIIGATA • OSPFv2 • OSPFv3 • RIP • STAR • ZRP *Default:* BELLMANFORD	Name of the routing protocol used at the interface. **Note:** This parameter must be used to specify the routing protocol at the interface if the node is an IPv4 node or a dual IP node. It can also be used if the node is an IPv6 node. If the node is an IPv6 node and both ROUTING-PROTOCOL and ROUTING-PROTOCOL-IPv6 are specified, then the routing protocol specified by ROUTING-PROTOCOL-IPv6 is used. If the node is an IPv4 node or a dual IP node, then this parameter specifies the IPv4 routing protocol at the interface. If this parameter is set to NONE, no IPv4 routing protocol is used at the interface. If the node is an IPv6 node, then this parameter specifies the IPv6 routing protocol at the interface. If the node is an IPv6 node and this parameter is set to NONE, no IPv6 routing protocol is used at the interface unless one is specified by the parameter ROUTING-PROTOCOL-IPv6.

TABLE 4-65. Unicast Routing Parameters (Continued)

Parameter	Value	Description
`ROUTING-PROTOCOL-IPv6` *Optional* *Scope:* All	List: • `AODV` • `DYMO` • `NONE` • `OLSR-INRIA` • `OLSRv2-NIIGATA` • `OSPFv3` • `RIPng` *Default:* `NONE`	Name of the IPv6 routing protocol used at the interface. **Note:** This parameter must be used to specify the routing protocol at the interface if the node is a dual IP node. It can also be used if the node is an IPv6 node. If the node is an IPv6 node and both `ROUTING-PROTOCOL` and `ROUTING-PROTOCOL-IPv6` are specified, then the routing protocol specified by `ROUTING-PROTOCOL-IPv6` is used. If the node is an IPv6 node or a dual IP node, then this parameter specifies the IPv6 routing protocol at the interface. If the node is a dual IP node and this parameter is set to `NONE`, no IPv6 routing protocol is used at the interface. If the node is an IPv6 node and this parameter is set to `NONE`, no IPv6 routing protocol is used at the interface unless one is specified by the parameter `ROUTING-PROTOCOL`.
`IP-FORWARDING` *Optional* *Scope:* Global, Node	List: • `YES` • `NO` *Default:* `YES`	Indication whether the node forwards IP packets.
`STATIC-ROUTE` *Optional* *Scope:* Global, Node	List: • `YES` • `NO` *Default:* `NO`	Indication whether the node uses static routes. See Section 4.2.8.3.2.1.1 for a description of static routes.
`DEFAULT-ROUTE` *Optional* *Scope:* Global, Node	List: • `YES` • `NO` *Default:* `NO`	Indication whether the node uses default routes. See Section 4.2.8.3.2.1.1 for a description of default routes.
`DEFAULT-GATEWAY` *Optional* *Scope:* Global, Node	Integer (> 0) or IPv4 Address	Node ID or IPv4 address of the default gateway for the node. If the node receives a packet for which it has no routing information, the node sends the packet to the default gateway, if one is specified; otherwise, the node drops the packet.

Table 4-66 describes the different unicast routing protocols in EXata. The table also specifies whether a routing protocol is a proactive or on-demand protocol, and if it is supported in IPv4 networks, IPv6 networks, or both. See the corresponding model library for a detailed description of each protocol and its parameters.

TABLE 4-66. Unicast Routing Protocols

Command Line Name	GUI Name	Description	Type	IP Version(s)	Model Library
`ANODR`	ANODR	ANonymous On-Demand Routing (ANODR) protocol. This is a secure routing protocol.	On-demand	IPv4	Cyber
`AODV`	AODV	Ad-hoc On-demand Distance Vector (AODV) routing protocol.	On-demand	IPv4, IPv6	Wireless
`BELLMANFORD`	Bellman Ford	Bellman-Ford routing protocol.	Proactive	IPv4	Developer
`DSR`	DSR	Dynamic Source Routing (DSR) protocol.	On-demand	IPv4	Wireless
`DYMO`	DYMO	DYnamic MANET On-demand (DYMO) routing protocol.	On-demand	IPv4, IPv6	Wireless
`EIGRP`	EIGRP	Enhanced Interior Gateway Routing Protocol (EIGRP). This is a distance vector routing protocol designed for fast convergence.	Proactive	IPv4	Multimedia and Enterprise
`FISHEYE`	Fisheye	Fisheye Routing Protocol. This is a link state-based routing protocol.	Proactive	IPv4	Wireless
`FSRL`	LANMAR	Landmark Ad-hoc Routing (LANMAR) protocol. This protocol uses Fisheye as the local scope routing protocol.	Proactive	IPv4	Wireless
`IARP`	IARP	IntrA-zone Routing Protocol (IARP) This is a vector-based proactive routing protocol and is a component of ZRP.	Proactive	IPv4	Wireless
`IERP`	IERP	Inter-zone Routing Protocol (IERP) This is an on-demand routing protocol and is a component of ZRP.	On-demand	IPv4	Wireless
`IGRP`	IGRP	Interior Gateway Routing Protocol (IGRP). This is a distance vector Interior Gateway protocol (IGP).	Proactive	IPv4	Multimedia and Enterprise
`LAR1`	LAR1	Location-Aided Routing (LAR) protocol, version 1. This protocol utilizes location information to improve scalability of routing.	On-demand	IPv4	Wireless

TABLE 4-66. Unicast Routing Protocols (Continued)

Command Line Name	GUI Name	Description	Type	IP Version(s)	Model Library
OLSR-INRIA	OLSR INRIA	Optimized Link State Routing (OLSR) protocol. This is a link state-based routing protocol.	Proactive	IPv4, IPv6	Wireless
OLSRv2-NIIGATA	OLSRv2 NIIGATA	Optimized Link State Routing, version 2 (OLSRv2) protocol. This is a successor of the OLSR protocol.	Proactive	IPv4, IPv6	Wireless
OSPFv2	OSPFv2	Open Shortest Path First (OSPF) routing protocol, version 2. This is a link state-based routing protocol for IPv4 networks.	Proactive	IPv4	Multimedia and Enterprise
OSPFv3	OSPFv3	Open Shortest Path First (OSPF) routing protocol, version 3. This is a link state-based routing protocol for IPv6 networks.	Proactive	IPv6	Multimedia and Enterprise
RIP	RIP	Routing Information Protocol (RIP) routing protocol.	Proactive	IPv4	Developer
RIPng	RIPng	Routing Information Protocol, next generation (RIPng) routing protocol. This protocol can be used for IPv6 networks.	Proactive	IPv6	Developer
STAR	STAR	Source Tree Adaptive Routing (STAR) protocol.	Proactive	IPv4	Wireless
ZRP	ZRP	Zone Routing Protocol.	Hybrid (Proactive and On-demand)	IPv4	Wireless

4.2.8.3.2.1.1 Static and Default Routes

Static and default routes are used to pre-configure the IP forwarding table with permanent routes. Static and default routes work in the same manner but have different priorities. Static routes have a higher priority than routes added to the IP forwarding table by dynamic routing protocols, which have a higher priority than default routes. Static routes can be used to bypass dynamic routing protocols for specific destinations. Default routes are usually serve as fallback routes to be used when dynamic routing protocols fail to discover a route. Default routes are also used to direct all packets at a host to its default gateway.

If a node is configured to use an on-demand routing protocol (see Table 4-66), then only routes discovered by the routing protocol are used even if static and/or default routes are enabled at the node.

If a node is configured to use a proactive routing protocol and static routes are enabled at the node, then the static routes take precedence over routes discovered by the routing protocol.

If static routes are disabled at a node and the node is not configured to use any routing protocol, then default routes are used default routes are enabled at the node.

Refer to *Developer Model Library* for a description of static and default routes.

4.2.8.3.2.2 GUI Configuration

To configure the unicast routing parameters in the GUI, do the following:

1. Go to one of the following locations:
   - To set properties for a specific wireless subnet, go to **Wireless Subnet Properties Editor > Routing Protocol > General**.
   - To set properties for a specific wired subnet, go to **Wired Subnet Properties Editor > Routing Protocol > General**.
   - To set properties for a specific point-to-point link, go to **Point-to-point Link Properties Editor > Point-to-point Link Properties > Routing Protocol**.
   - To set properties for a specific node, go to **Default Device Properties Editor > Node Configuration > Routing Protocol**.
   - To set properties for a specific interface of a node, go to one of the following locations:
     - **Interface Properties Editor > Interfaces > Interface # > Routing Protocol**
     - **Default Device Properties Editor > Interfaces > Interface # > Routing Protocol**.

   In this section, we show how to configure unicast routing parameters in the **Default Device Properties Editor** when **Network Protocol** is set as *Dual IP*. Parameters can be set in the other properties editors in a similar way.

2. Set the unicast routing parameters listed in Table 4-67. The available unicast routing protocols are listed in Table 4-66.

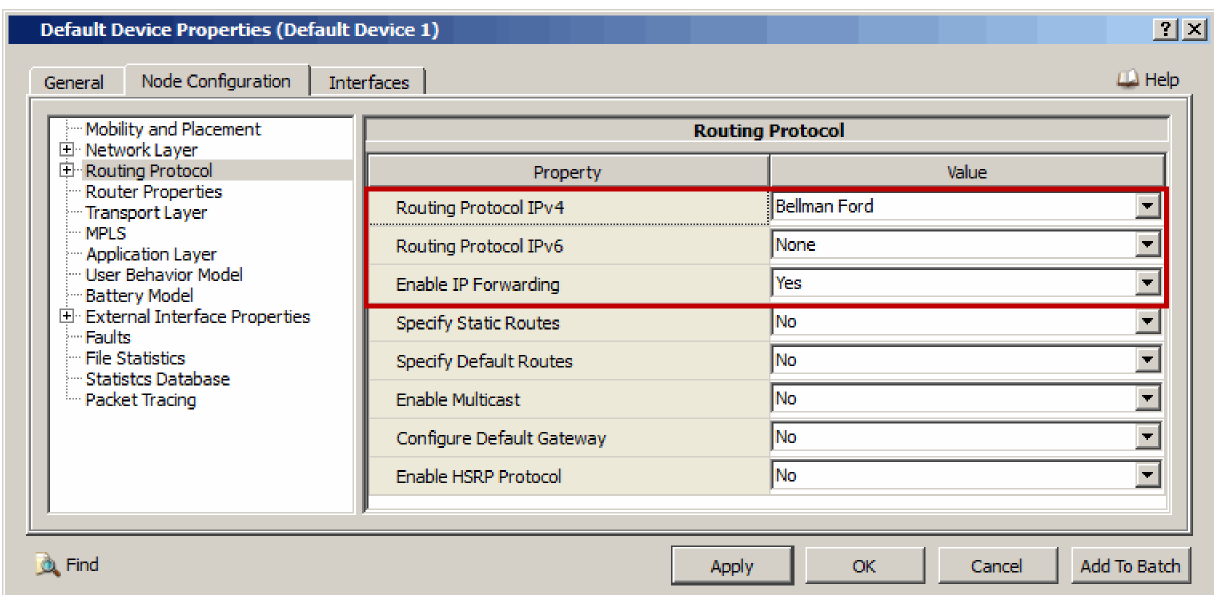


FIGURE 4-40. Setting Unicast Routing Parameters for a Dual IP Node

TABLE 4-67. Command Line Equivalent of Unicast Routing Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Routing Protocol IPv4	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`ROUTING-PROTOCOL`
Routing Protocol IPv6	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`ROUTING-PROTOCOL-IPv6`
Enable IP Forwarding	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`IP-FORWARDING`

3. Set the dependent parameters for the selected routing protocol. See the model library referenced in Table 4-66 for details.
4. To enable static and/or default routes, set the parameters listed in Table 4-68, and set the dependent parameters. Static and default routes are described in Section 4.2.8.3.2.1.

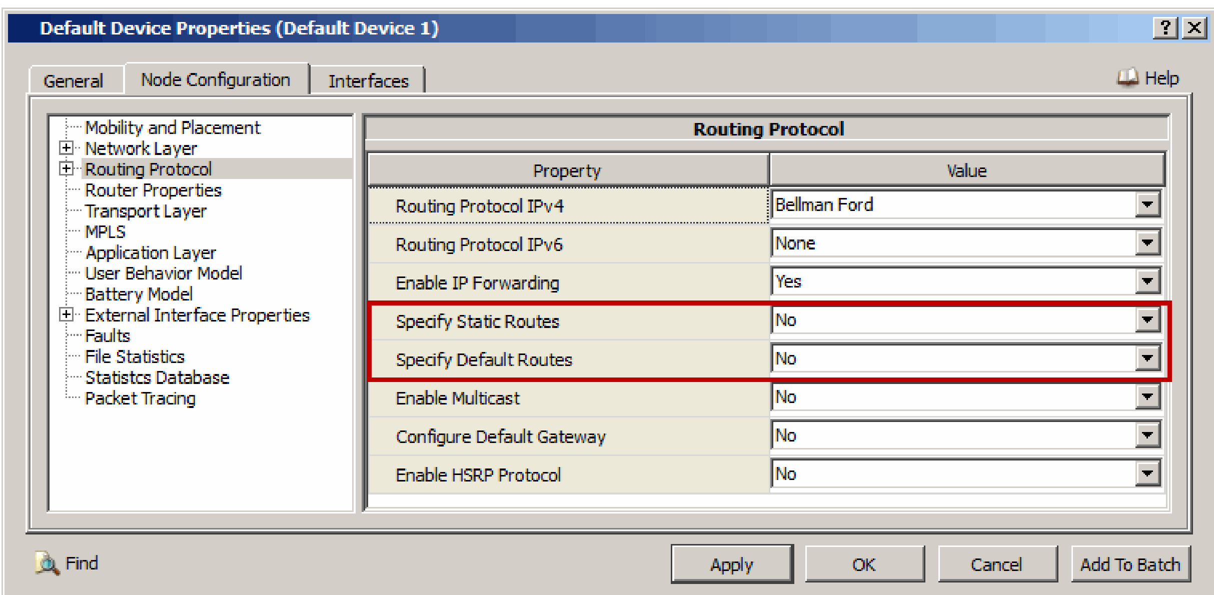


FIGURE 4-41. Enabling Static and Default Routes for a Node

TABLE 4-68. Command Line Equivalent of Static and Default Route Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Specify Static Routes	Node, Point-to-point Link	`STATIC-ROUTE`
Specify Default Routes	Node, Point-to-point Link	`DEFAULT-ROUTE`

5. To configure a default gateway for a node, set **Configure Default Gateway** to *Yes* and set the dependent parameters listed in Table 4-69.

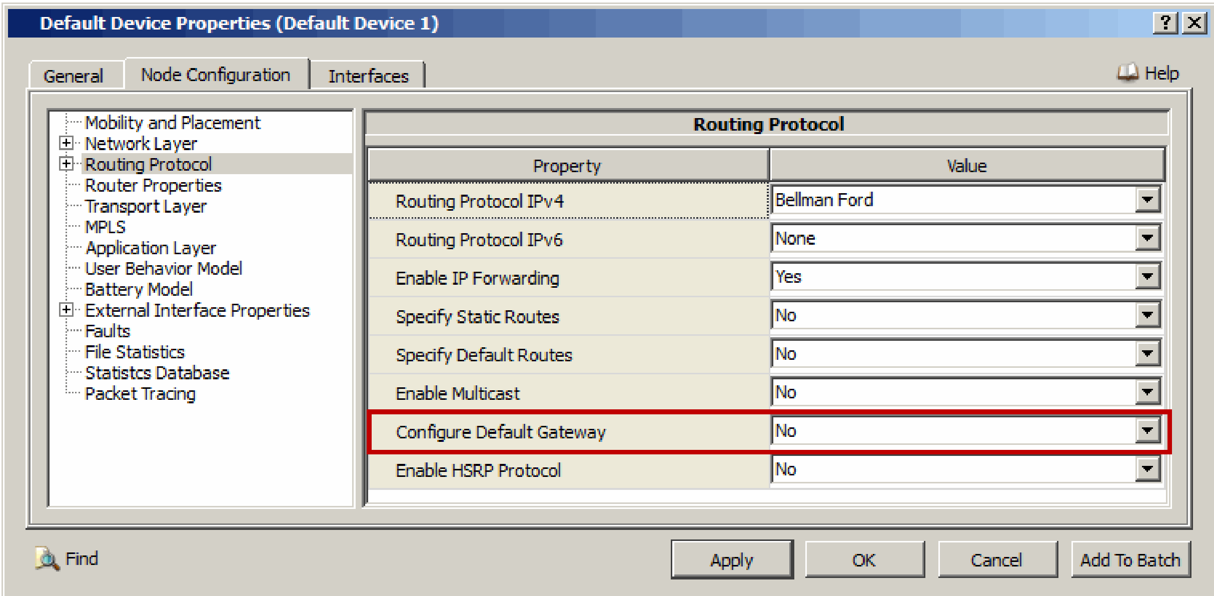


FIGURE 4-42. Enabling Default Gateway for a Node

TABLE 4-69. Command Line Equivalent of Default Gateway Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Default Gateway	Node	`DEFAULT-GATEWAY`

4.2.8.3.3 Multicast Routing

Multicast routing protocols are used to forward packets to groups of nodes. Group formation and management is an integral part of multicasting. Very often (but not always) a group management protocol, such as the Internet Group Management Protocol (IGMP), is used to form and manage groups. Thus, for a multicast scenario, IGMP may need to be configured in addition to the multicast routing protocol.

4.2.8.3.3.1 Command Line Configuration

To configure the multicast routing parameters for the command line interface, include the parameters listed in Table 4-70 in the scenario configuration (.config) file.

TABLE 4-70. Multicast Routing Parameters

Parameter	Value	Description
`MULTICAST-PROTOCOL` *Optional* *Scope:* All	List: • `NONE` • `DVMRP` • `MOSPF` • `ODMRP` • `PIM`	Name of the multicast routing protocol used at the interface. If this parameter is not specified, no multicast routing protocol is used at the interface.
`MULTICAST-STATIC-ROUTE` *Optional* *Scope:* Global, Node	List: • `YES` • `NO` *Default:* `NO`	Indication whether the node uses static multicast routes. See Section 4.2.8.3.3.1.2 for a description of static multicast routes.
`GROUP-MANAGEMENT-PROTOCOL` *Optional* *Scope:* Global, Node	List: • `IGMP`	Group management protocol to be used. See *Developer Model Library* for details of the IGMP model. **Note:** If this parameter is not specified, then no group management protocol is used.
`MULTICAST-GROUP-FILE` *Dependency:* `MULTICAST-PROTOCOL` ≠ `NONE` for some interface *Required* *Scope:* Global	Filename	Name of the Multicast Group file. For IGMP version 2, this file specifies when receiver nodes and interfaces join and leave multicast groups. For IGMP version 3, this file specifies when receiver nodes and interfaces join multicast groups and the source nodes from which traffic should be forwarded to the receiver nodes. The format of this file is described in Section 4.2.8.3.3.1.1.

Table 4-71 describes the different multicast routing protocols in EXata. The table also specifies whether a routing protocol is a proactive or on-demand protocol. See the corresponding model library for a detailed description of each protocol and its parameters.

Note: All multicast protocols listed in Table 4-71 are supported only for IPv4 networks.

TABLE 4-71. Multicast Routing Protocols

Command Line Name	GUI Name	Description	Type	Model LIbrary
`DVMRP`	DVMP	Distance Vector Multiple Routing Protocol (DVMRP) DVMRP is a multicast routing protocol designed for wired networks. It is a tree-based multicast scheme that uses reverse path multicasting.	Proactive	Multimedia and Enterprise
`MOSPF`	MOSPF	Multicast Open Path Shortest First (MOSPF) protocol. This is a multicast extension of OSPFv2. MOSPF is a pruned tree-based, multicast scheme that takes advantage of commonality of paths from source to destinations.	Proactive	Multimedia and Enterprise
`ODMRP`	ODMRP	On-Demand Multicast Routing Protocol (ODMRP). This is a mesh-based, wireless ad-hoc routing protocol for single subnets. It applies a soft on-demand procedures to build routes and uses soft state to maintain multicast group membership.	On-demand	Wireless
`PIM`	PIM	Protocol Independent Multicast (PIM) routing protocol. PIM relies on an underlying topology-gathering protocol to populate a routing table with routes. The routing table provides the next hop router along a multicast-capable path to each destination subnet. Both sparse mode and dense mode versions of the protocol are supported.	Proactive	Multimedia and Enterprise

4.2.8.3.3.1.1 Format of the Multicast Group File

Entries in the Multicast Group file depend on the IGMP version running at the receiver node or interface.

If IGMP version 2 is running on a receiver node or interface, then the entry in the Multicast Group file specifies when that node or interface joins or leaves a multicast group. The format for this entry is:

```
<Identifier> <Group-Address> <Join-Time> <Leave-Time>
```

where

`<Identifier>` Node ID or interface IP address of the receiver.

Note: If a node ID is specified, all interfaces of the node join and leave the multicast group at the specified times.

`<Group-Address>` IP address of the multicast group to join.

`<Join-Time>` Time when the node or interface joins the group.

`<Leave-Time>` Time when node or interface leaves the group.

If IGMP version 3 is running on a receiver node or interface, then the entry in the Multicast Group file specifies when that node or interface joins a multicast group and the source nodes and interfaces from which traffic should be forwarded to that receiver node or interface. The format for this subscription is:

```
<Identifier> <Group-Address> <Join-Time> <Filter> <Source-List>
```

where

`<Identifier>`	Node ID or interface IP address of the receiver. **Note:** If a node ID is specified, all interfaces of the node join the multicast group at the specified time.
`<Group-Address>`	IP address of the multicast group to join.
`<Join-Time>`	Time when the node or interface joins the group.
`<Filter>`	Filter to be used for forwarding traffic to the receiver. This can be `INCLUDE` or `EXCLUDE`. `INCLUDE` : Traffic from all sources in `<Source List>` sent to `<Group-Address>` should be forwarded to `<Identifier>`. `EXCLUDE` : Traffic from all sources except for those in `<Source-List>` sent to `<Group-Address>` should be forwarded to `<Identifier>`. **Note:** The EXata multicast protocol models do not support the `EXCLUDE` filter.
`<Source-List>`	List of node IDs and interface IP addresses of traffic sources to which the filter applies. Node IDs and IP addresses in this list are separated by spaces. **Note:** `<Source-List>` can be empty. If the filter is `INCLUDE` and the source list is empty, then `<Identifier>` stops receiving traffic from all sources sent to `<Group-Address>` at `<Join-Time>`.

Note: A subscription by a node or interface to traffic from a list of sources sent to a multicast address remains in effect until a later subscription (if any) by the same node or interface to the same multicast group with a different list of sources. The list of sources in the later subscription replaces all sources in the previous subscription.

Example:

The following lines show a sample Multicast Group file:

```
5          225.0.0.0 1M  13M
190.0.0.1  255.0.0.0 1M  13M
5          225.0.0.0 17M 30M
190.0.2.1  255.0.0.0 17M 30M
1          232.0.0.1 2M  INCLUDE 7 190.0.1.2
1          232.0.0.1 10M INCLUDE 8 190.0.1.2
1          232.0.0.1 12M INCLUDE
```

In the above example:

- Node 5 and interface 190.0.0.1 are running IGMPv2. They join multicast group 225.0.0.0 at time 1 minute and leave at time 13 minutes. They rejoin the same group at time 17 minutes and leave at time 30 minutes.
- Node 1 is running IGMPv3. It joins multicast group 232.0.0.1 at time 2 minutes and subscribes to traffic only from node 7 and interface 190.0.1.2. At time 10 minutes, it drops node 7 from its subscription and adds node 8 (it continues to receive traffic from 190.0.1.2). At time 12 minutes, it stops receiving traffic from all sources that send traffic to group 232.0.0.1. At time 15 minutes, it starts receiving traffic from all sources that send traffic to group 232.0.0.1.

4.2.8.3.3.1.2 Static Multicast Routes

Static multicast routes are used to configure permanent multicast routes in the IP multicast forwarding table. These are similar to static and default routes described in Section 4.2.8.3.2.

The IP multicast forwarding capability implemented in EXata does not support multiple routes for the same multicast address discovered by different multicast routing protocols. Thus, if both static multicast routes and multicast routing protocols are configured, the multicast routing protocol may overwrite the static multicast routes.

Refer to *Developer Model Library* for a description of static multicast routes.

4.2.8.3.3.2 GUI Configuration

To configure the unicast routing parameters in the GUI, perform the following steps:

1. Go to one of the following locations:

- To set properties for a specific wireless subnet, go to **Wireless Subnet Properties Editor > Routing Protocol > General**.
- To set properties for a specific wired subnet, go to **Wired Subnet Properties Editor > Routing Protocol > General**.
- To set properties for a specific point-to-point link, go to **Point-to-point Link Properties Editor > Point-to-point Link Properties > Routing Protocol**.
- To set properties for a specific node, go to **Default Device Properties Editor > Node Configuration > Routing Protocol**.
- To set properties for a specific interface of a node, go to one of the following locations:
  - **Interface Properties Editor > Interfaces > Interface # > Routing Protocol**
  - **Default Device Properties Editor > Interfaces > Interface # > Routing Protocol**.

In this section, we show how to configure multicast routing parameters for a node in the **Default Device Properties Editor**. Parameters can be set in the other properties editors in a similar way.

2. To configure multicast routing, set **Enable Multicast** to *Yes* and set the dependent parameters listed in Table 4-72. The available multicast routing protocols are listed in Table 4-71.

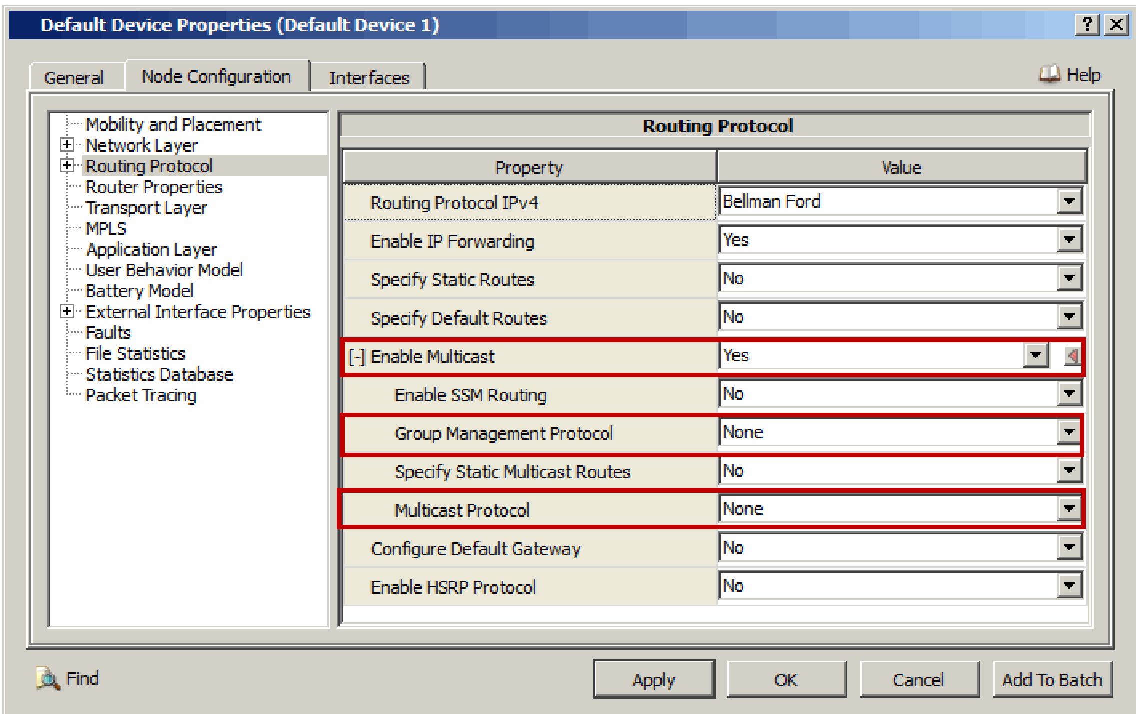


FIGURE 4-43. Setting Multicast Routing Parameters for a Dual IP Node

TABLE 4-72. Command Line Equivalent of Unicast Routing Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Group Management Protocol	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`GROUP-MANAGEMENT-PROTOCOL`
Multicast Protocol	Node, Wireless Subnet, Wired Subnet, Point-to-point Link, Interface	`MULTICAST-PROTOCOL`

3. Set the dependent parameters for the selected multicast protocol. See the model library referenced in Table 4-67 for details.

4. To enable static multicast routes, set **Specify Static Multicast Routes** to *Yes* and set the dependent parameters listed in Table 4-73. Static multicast routes are described in Section 4.2.8.3.3.1.

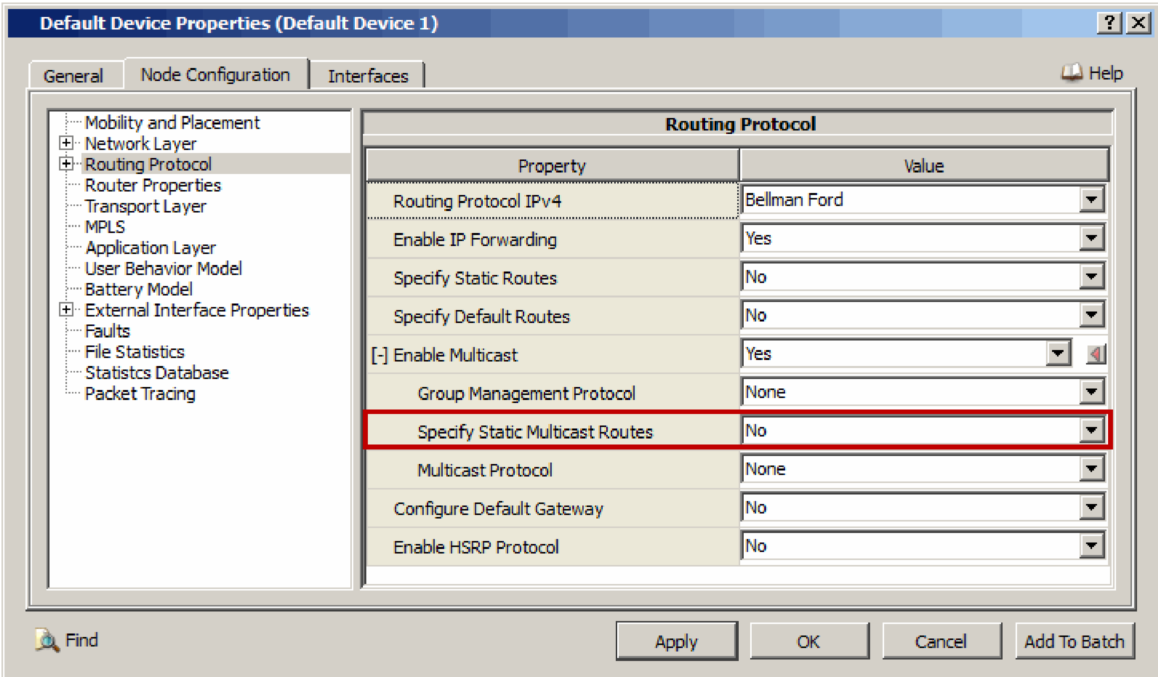


FIGURE 4-44. Enabling Static Multicast Routes for a Node

TABLE 4-73. Command Line Equivalent of Static Multicast Route Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Specify Static Multicast Routes	Node, Point-to-point Link	`MULTICAST-STATIC-ROUTE`

5. Using the **Multicast Group Editor**, specify the name of the Multicast Group file or create multicast groups, as described in Section 4.2.8.3.3.2.1.

4.2.8.3.3.2.1 Multicast Group Editor

To launch the Multicast Group Editor, go to the **Tools** menu and select **Multicast Group Editor**.

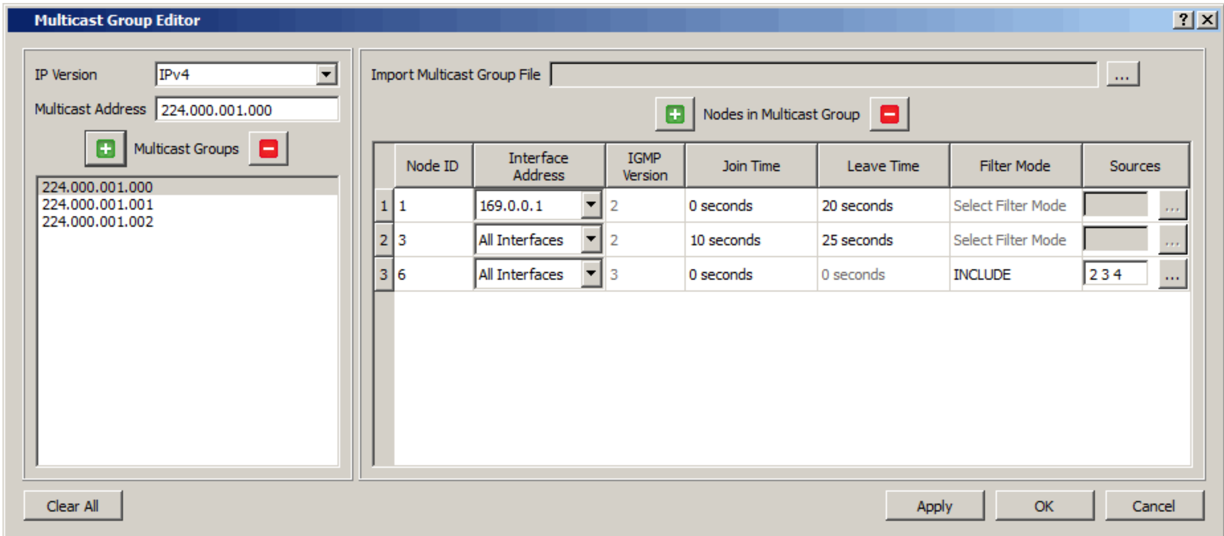


FIGURE 4-45. Multicast Group Editor

To import multicast groups from a file, click the ... button next to **Import Multicast Group File** and select a Multicast Group file from the file browser. All multicast groups in the selected file are displayed in the left panel.

To create a new multicast group, do the following:

1. Set the IP version (*IPv4* or *IPv6*) of the multicast group to be created in the **IP Type** field.
2. Click the button next to **Multicast Groups** in the left panel. This creates an empty multicast group of the selected IP type. A default address is assigned to the multicast group.
3. To change the address of a group, select the group and specify a new address in the **Multicast Address** field.
4. To specify the subscription of nodes and interfaces to a group, select the group in the left panel and do the following:

   a. Click the button next to **Nodes in Multicast Group** in the right panel. This creates a new row in the right panel. Each row in the right panel has the following entries: **Node ID**, **Interface Address**, **IGMP Version**, **Join Time**, **Leave Time**, **Filter Mode**, and **Sources**. The **IGMP Version** field can not be modified and is set to the version of IGMP version configured at that node. (IGMP version must be configured for nodes using the Default Device Properties Editor before using the Multicast Group Editor.) If **IGMP Version** is 2, then the **Filter Mode** and **Sources** fields are not applicable and can not be modified. If IGMP Version is 3, then the **Leave Time** field is not applicable and can not be modified.

   b. Select a node from the pull-down list in the **Node ID** column and select a specific interface or *All Interfaces* from the pull-down list in the **Interface Address** column. Set the join time in the **Join Time** column.

   c. If **IGMP Version** is 2, then set the leave time in the **Leave Time** column.

d. If **IGMP Version** is 3, then select a filter mode (*INCLUDE* or *EXCLUDE*) from the pull-down list in the **Filter Mode** column. In the **Sources** column, specify the node IDs or interface addresses of the sources whose traffic should be included or excluded. (See Section 4.2.8.3.3.1.1 for a description of the filters.)

Note: You can also click the ... button to open an editor in which to specify node IDs and interface addresses of sources. Node IDs and interface addresses can be separated by spaces or commas. You can also specify a range of node IDs by using the keyword *thru*.

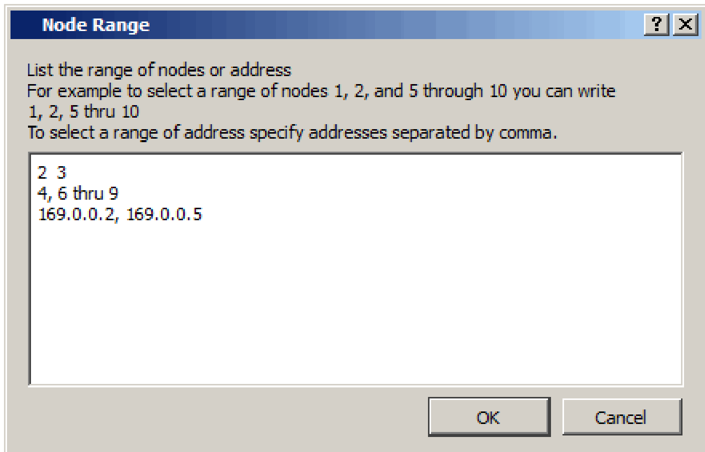


FIGURE 4-46. Specifying Traffic Sources

e. To delete a row, select the row and click the button in the right panel.

To delete a group, select the group and click the button in the left panel.

Note: Adding a node or interface to a multicast group does not automatically enable mutlicasting at that node or interface. To enable multicast at a node or interface, see Section 4.2.8.3.2.2.

4.2.8.3.4 Schedulers and Queues

Since data links have limited capacity, the Network Layer may need to temporarily buffer data packets in queues before handing them to the data link layer. The network protocol usually maintains separate queues for different outgoing interfaces. In addition, for each outgoing interface, it usually maintains multiple queues which have different priorities. Packets are assigned to appropriate queues according to their priorities.

When multiple queues are implemented at an interface, a scheme is needed to determine the order in which packets from these queues are transmitted. This scheme is implemented by a scheduler.

4.2.8.3.4.1 Command Line Configuration

To configure queues and schedulers for the command line interface, include the parameters listed in Table 4-74 in the scenario configuration (.config) file.

TABLE 4-74. Scheduler and Queue Parameters

Parameter	Value	Description
`IP-QUEUE-PRIORITY-INPUT-QUEUE-SIZE` *Optional* *Scope:* All	Integer *Range:* `> 0` *Default:* `150000` *Unit:* bytes	Size of each input priority queue.
`IP-QUEUE-SCHEDULER` *Required* *Scope:* All	List: • `CBQ` • `DIFFSERV-ENABLED` • `ROUND-ROBIN` • `SELF-CLOCKED-FAIR` • `STRICT-PRIORITY` • `WEIGHTED-FAIR` • `WEIGHTED-ROUND-ROBIN`	Type of scheduler at the interface.
`IP-QUEUE-NUM-PRIORITIES` *Required* *Scope:* All	Integer *Range:* `[1, 256]`	Number of priority queues at an interface.
`IP-QUEUE-PRIORITY-QUEUE-SIZE` *Optional* *Scope:* All *Instances:* queue index	Integer *Range:* `> 0` *Default:* `150000` *Unit:* bytes	Size of each output priority queue.
`IP-QUEUE-TYPE` *Required* *Scope:* All *Instances:* queue index	List: • `FIFO` • `RED` • `RIO` • `WRED`	Type of the priority queue.

Table 4-75 describes the different scheduler models in EXata. See the corresponding model library for a detailed description of each model and its parameters.

TABLE 4-75. Scheduler Models

Command Line Name	GUI Name	Description	Model Library
`CBQ`	Class Based Queuing	Class-based queuing algorithm. This algorithm is usually used by DiffServ. Queues are divided into classes. The network protocol allocates bandwidth for each queue. Scheduling is based on the bandwidth available to each class.	Developer
`DIFFSERV-ENABLED`	DiffServ	Differentiated services (DiffServ) quality of service protocol. When this option is selected, DiffServ queues and schedulers are configured. The DiffServ scheduler for IP is a combination of two schedulers: the inner and outer schedulers. Generally, the weighted fair or weighted round-robin scheduler is chosen as the inner scheduler and the strict priority scheduler is chosen as the outer scheduler.	Multimedia and Enterprise
`ROUND-ROBIN`	Round Robin	Round-robin scheduler. Queues are scheduled in a round-robin fashion.	Developer
`SELF-CLOCKED-FAIR`	Self Clocked Fair	Self-clocked fair scheduler. Scheduling is based on the Self-Clocked Fair Queuing (SFFQ) algorithm.	Developer
`STRICT-PRIORITY`	Strict Priority	Strict priority scheduler. Packets are scheduled strictly based on their priority. A packet is scheduled only when all higher priority queues are empty.	Developer
`WEIGHTED-FAIR`	Weighted Fair	Weighted fair scheduler. Scheduling is based on the Weighted Fair Queuing (WFQ) algorithm.	Developer
`WEIGHTED-ROUND-ROBIN`	Weighted Round Robin	Weighted Round-Robin (WRR) scheduler. This is a variant of the round-robin scheduler. The round-robin scheduler services one packet from each queue in turn. The WRR scheduler services multiple packets from each queue in turn, where the number of packets serviced depends on the queue's weight.	Developer

Table 4-76 describes the different queue models in EXata. See the corresponding model library for a detailed description of each model and its parameters.

TABLE 4-76. Queue Models

Command Line Name	GUI Name	Description	Model Library
`FIFO`	FIFO	First In First Out (FIFO) queue. This is the basic queue type and is also called the Drop Tail queue. Packets are enqueued as long as there is buffer space available. If the queue is full when a packet arrives, the packet is dropped.	Developer
`RED`	RED	Random Early Drop (RED) queue. This queue is similar to FIFO, except that when the queue length exceeds a certain threshold, arriving packets are randomly dropped with a probability that depends on the queue length.	Developer
`RIO`	RIO	RED with In/Out bit (RIO) queue. RIO is a multiple average multiple threshold variant of RED that operates two-color and three-color modes. Twin and three RED algorithms are used in two-color and three-color modes, respectively.	Developer
`WRED`	WRED	Weighted Random Early Drop (WRED) queue. WRED is a variant of RED and uses three RED algorithms for three drop precedence levels.	Developer

4.2.8.3.4.2 GUI Configuration

To configure scheduler and queue models in the GUI, perform the following steps:

1. Go to one of the following locations:

- To set properties for a specific wireless subnet, go to **Wireless Subnet Properties Editor > Network Layer > Schedulers and Queues**.
- To set properties for a specific node, go to **Default Device Properties Editor > Node Configuration > Network Layer > Schedulers and Queues**.
- To set properties for a specific interface of a node, go to one of the following locations:
  - **Interface Properties Editor > Interfaces > Interface # > Network Layer > Schedulers and Queues**
  - **Default Device Properties Editor > Interfaces > Interface # > Network Layer > Schedulers and Queues**.

In this section, we show how to configure scheduler and queue parameters for a node in the **Default Device Properties Editor**. Parameters can be set in the other properties editors in a similar way.

2. Set the queue and scheduler parameters listed in Table 4-77. The available scheduler models are listed in Table 4-75.

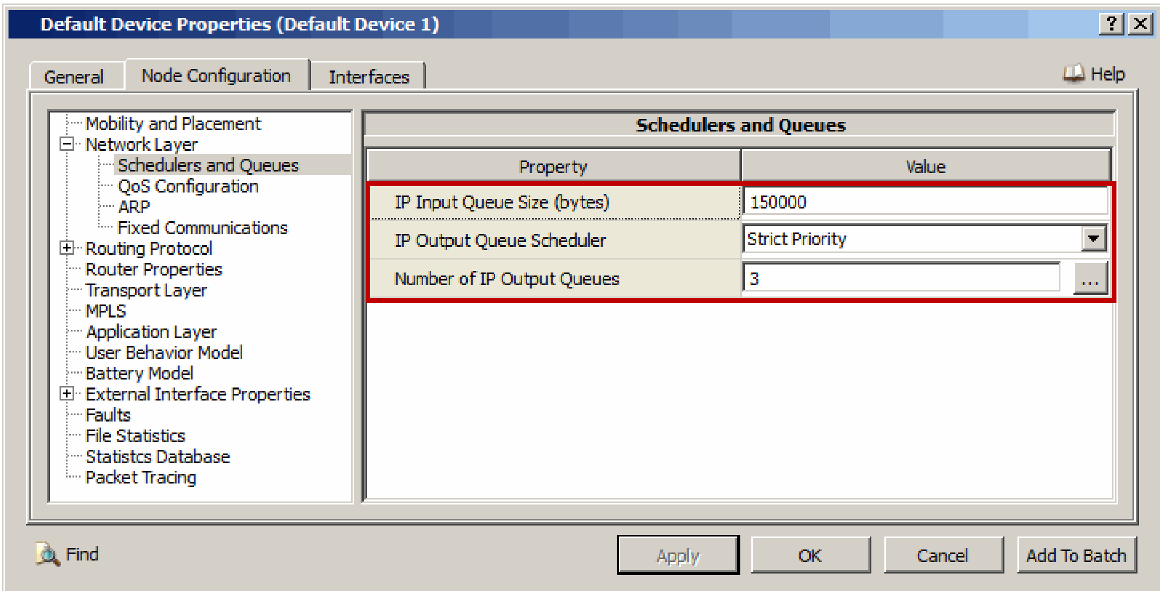


FIGURE 4-47. Setting Scheduler Parameters

TABLE 4-77. Command Line Equivalent of Scheduler Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
IP Input Queue Size	Node, Subnet, Interface	`IP-QUEUE-PRIORITY-INPUT-QUEUE-SIZE`
IP Output Queue Scheduler	Node, Subnet, Interface	`IP-QUEUE-SCHEDULER`
Number of IP Output Queue	Node, Subnet, Interface	`IP-QUEUE-NUM-PRIORITIES`

3. To configure the properties of IP output queues, do the following:

   a. Click the **Open Array Editor** [...] button in the **Value** column. This opens the Array Editor (Figure 4-48).

   b. In the left panel of the Array Editor, select the index of the queue to be configured. In the right panel, set the parameters listed in Table 4-78 for each IP output queue index.

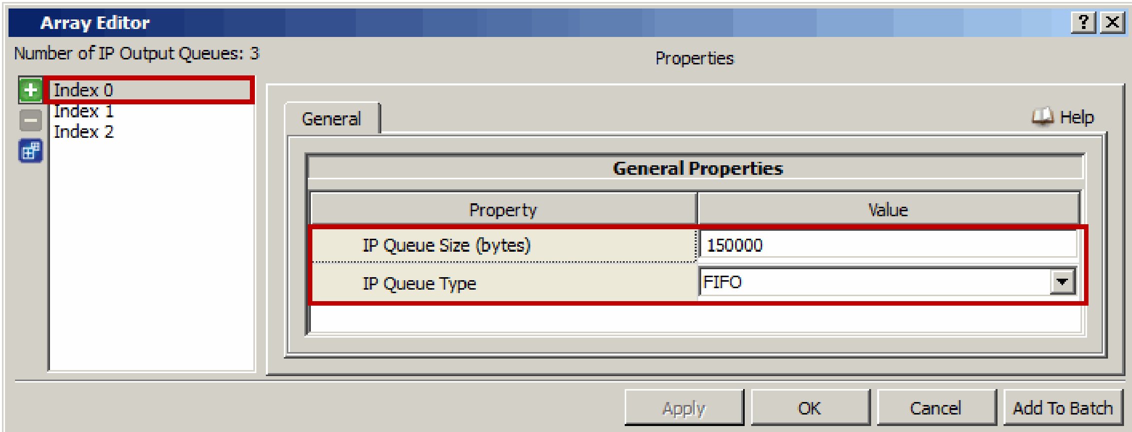


FIGURE 4-48. Setting Queue Parameters

TABLE 4-78. Command Line Equivalent of Queue Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
IP Queue Size	Node, Subnet, Interface	`IP-QUEUE-PRIORITY-QUEUE-SIZE`
IP Queue Type	Node, Subnet, Interface	`IP-QUEUE-TYPE`

4. Set the dependent parameters for the selected scheduler and queue models. See the model library referenced in Table 4-75 and Table 4-76 for details.

4.2.8.4 Transport Layer

A transport protocol provides end-to-end data transport services. It serves the Application Layer and allows multiple Application Layer sessions to be multiplexed on the transport services. The two major transport protocols are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is a connection-based protocol and provides reliable data transmissions with congestion control and flow control. UDP is a simple, connection-less protocol. EXata also implements another transport protocol, the Traffic Extension to Resource Reservation Protocol (RSVP-TE), which is a special protocol for distributing the labels for MPLS networks.

TCP and UDP are always enabled. RSVP-TE can be enabled or disabled.

4.2.8.4.1 Command Line Configuration

To configure RSVP-TE for the command line interface, include the parameters listed in Table 4-79 in the scenario configuration (.config) file.

TABLE 4-79. RSVP-TE Protocol Parameters

Parameter	Value	Description
TRANSPORT-PROTOCOL-RSVP *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* YES	Indication whether RSVP-TE is enabled at the node. YES : RSVP-TE is enabled. NO : RSVP-TE is not enabled

Table 4-80 describes the different transport protocols in EXata. The table also specifies whether a transport protocol is supported in IPv4 networks, IPv6 networks, or both. See the corresponding model library for a detailed description of each protocol and its parameters.

TABLE 4-80. Transport Protocols

Protocol	Description	IP Version(s)	Model Library
RSVP-TE	Resource Reservation Protocol - Traffic Extension (RSVP-TE). This transport protocol is used to distribute MPLS labels.	IPv4	Developer
TCP	Transmission Control Protocol (TCP). This is a connection-based transport protocol that provides reliable, end-to-end data transmission service.	IPv4, IPv6	Developer
UDP	User Datagram Protocol (UDP). This is a connection-less transport protocol that provides best-effort data transmission service.	IPv4, IPv6	Developer

4.2.8.4.2 GUI Configuration

To configure RSVP-TE for a specific node, do the following:

1. Go to **Default Device Properties Editor > Node Configuration > Transport Layer**.
2. To enable or disable RSVP-TE, set the parameter **Enable RSVP**.

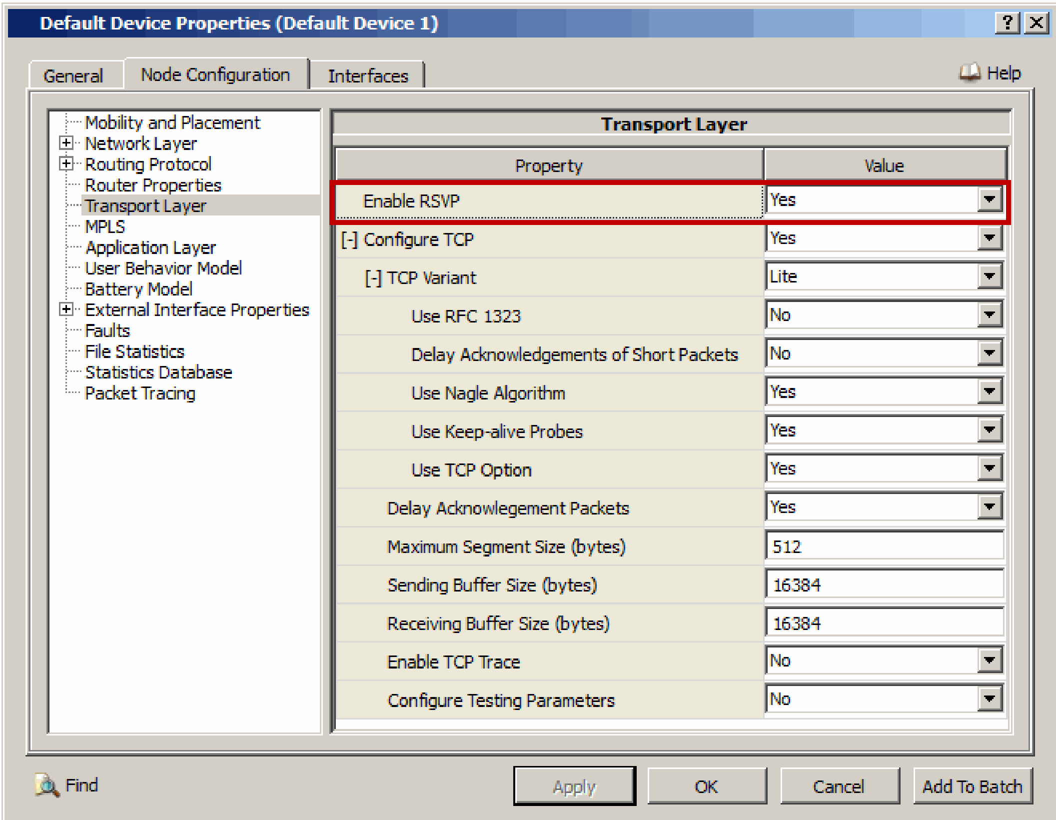


FIGURE 4-49. Enabling RSVP-TE for a Node

TABLE 4-81. Command Line Equivalent of RSVP-TE Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Enable RSVP	Node	`TRANSPORT-PROTOCOL-RSVP`

3. To configure TCP parameters, set **Configure TCP** to *Yes* and set the dependent parameters. Refer to *Developer Model Library* for details.

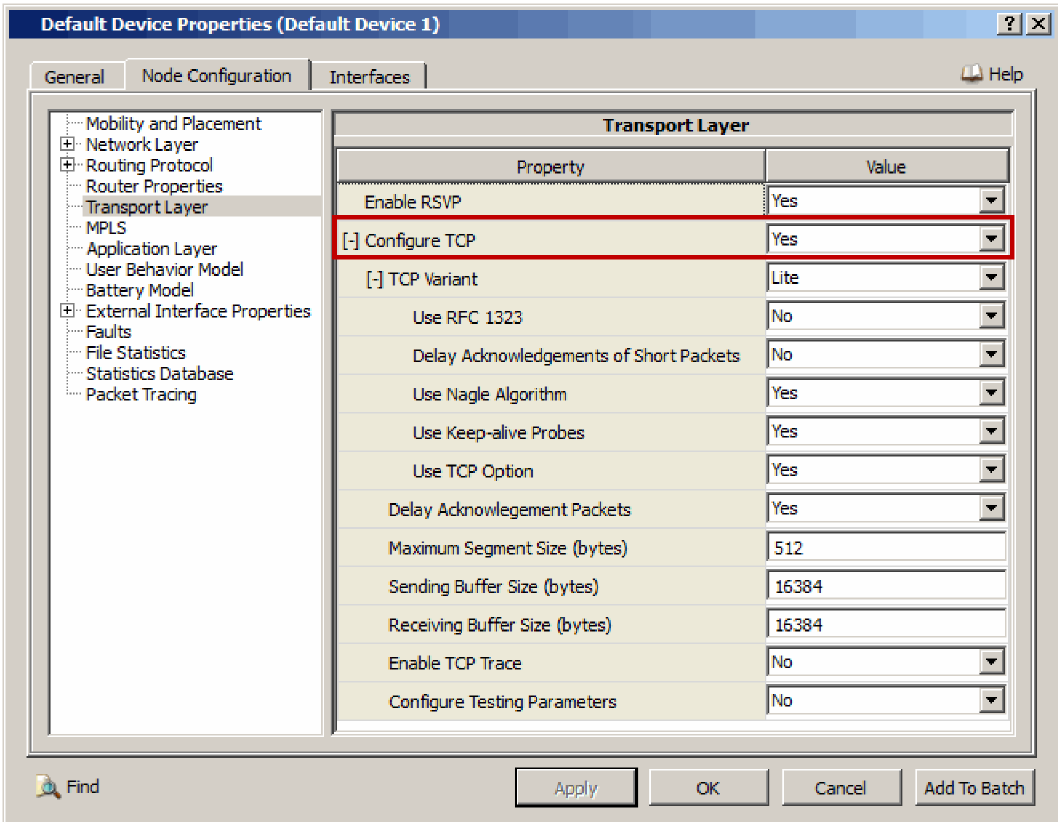


FIGURE 4-50. Configuring TCP Parameters

4.2.8.5 Application Layer

The Application Layer implements traffic generators which simulate user data traffic, such as FTP, Telnet, etc.

Note: In addition to traffic generators, some routing protocols are also implemented at the Application Layer. These routing protocols are described with the Network Layer routing protocols in Section 4.2.8.3.2 and Section 4.2.8.3.3.

4.2.8.5.1 Command Line Configuration

Traffic generators are configured in an application configuration file (which usually has the extension ".app"). The name of the application file is specified in the scenario configuration (.config) file. The Application Layer parameters are described in Table 4-82.

TABLE 4-82. Application Layer Parameters

Parameter	Value	Description
`APP-CONFIG-FILE` *Required* *Scope:* Global	Filename	Name of the application configuration file. The application file can have any valid file name and extension, but it is recommended that it have the extension .app. The syntax of the application file is described in Section 4.2.8.5.1.1.

Table 4-83 describes the different application protocols in EXata. The table also specifies whether a application protocol is supported in IPv4 networks, IPv6 networks, or both. See the corresponding model library for a detailed description of each protocol and its parameters.

TABLE 4-83. Traffic Generators

Application Protocol Abbreviation	Description	IP Version(s)	Model Library
`CBR`	Constant Bit Rate (CBR) traffic generator. This UDP-based client-server application sends data from a client to a server at a constant bit rate.	IPv4, IPv6	Developer
`CELLULAR-ABSTRACT-APP`	Abstract cellular application. This is an application to generate traffic for networks running abstract cellular models.	IPv4	Cellular
`FTP`	File Transfer Protocol (FTP). This tcplib application generates TCP traffic based on historical trace data.	IPv4, IPv6	Developer
`FTP/GENERIC`	Generic FTP. This model is similar to the FTP model but allows the user to have more control over the traffic properties. It uses FTP to transfer a user-specified amount of data.	IPv4, IPv6	Developer
`GSM`	Global System for Mobile communications (GSM). This is an application for generating traffic for GSM networks.	IPv4	Cellular
`HTTP`	Hypertext Transfer Protocol (HTTP). The HTTP application generates realistic web traffic between a client and one or more servers. The traffic is randomly generated based on historical data.	IPv4, IPv6	Developer
`LOOKUP`	Look-up traffic generator. This is an abstract model of unreliable query/ response traffic, such as DNS look-up, or pinging.	IPv4	Developer

TABLE 4-83. Traffic Generators (Continued)

Application Protocol Abbreviation	Description	IP Version(s)	Model Library
`MCBR`	Multicast Constant Bit rate (MCBR). This model is similar to CBR and generates multicast constant bit rate traffic.	IPv4	Developer
`PHONE-CALL`	Phone call application. This model simulates a phone call between two end users in a UMTS network.	IPv4, IPv6	UMTS
`SUPER-APPLICATION`	Super application. This model can simulate both TCP and UDP flows as well as two-way (request-response type) UDP sessions.	IPv4	Developer
`TELNET`	Telnet application. This model generates realistic Tenet-style TCP traffic between a client and a server based on historical data. It is part of the tcplib suit of applications.	IPv4, IPv6	Developer
`TRAFFIC-GEN`	Random distribution-based traffic generator. This is a flexible UDP traffic generator that supports a variety of data size and interval distributions and QoS parameters.	IPv4, IPv6	Developer
`TRAFFIC-TRACE`	Trace file-based traffic generator. This model generates traffic according to a user-specified file, and like Traffic-Gen, it supports QoS parameters.	IPv4	Developer
`VBR`	Variable Bit Rate (VBR) traffic generator. This model generates fixed-size data packets transmitted using UDP at exponentially distributed time intervals.	IPv4	Developer
`VOIP`	Voice over IP traffic generator. This model simulates IP telephony sessions.	IPv4	Multimedia and Enterprise
`ZIGBEEAPP`	ZigBee Application. This is similar to the CBR application but is used only in sensor networks.	IPv4, IPv6	Sensor Networks

4.2.8.5.1.1 Format of the Application Configuration File

The application configuration file specifies all applications (or traffic-generators) in the scenario.

The application file can have any valid filename and extension, but it is recommended that it have the extension ".app". See the file default.app in the folder EXATA_HOME/scenarios/default for an example of an application file.

Each line in the application configuration file specifies one application and has the following format:

```
<Application Name> <Application Parameters>
```

where

`<Application Name>`	Abbreviation for the application protocol. See Table 4-83.
`<Application Parameters>`	List of parameters for the application protocol. The list of parameters depends on the application. See the application protocol's description in the model libraries for a description of its parameters.

Notes: **1.** Each application specification should be on a single line by itself.

2. Comments can be entered any where in the application file (see Section 2.2.1).

Example:

The following lines show a segment of an application configuration file:

```
FTP/GENERIC 1 2 3 512 10 150S
FTP 1 2 10 150S
TELNET 3 4 10S 150S
CBR 19 17 10000 512   5S    70S 100S
CBR 11 29 10000 512 2.5S 82.49S 199S
```

For some application parameters that take numerical values, it is possible to specify a random distribution instead of a specific numerical value. See Section 2.2.8 for details of specifying random number distributions.

4.2.8.5.2 GUI Configuration

In the GUI, traffic-generators can be configured in the following ways:

- By specifying an application configuration file
- By configuring applications on the canvas. The following types of applications can be configured:
  - Client-server applications
  - Single host applications
  - Loopback applications

Specifying the Application Configuration File

To specify an application configuration file, do the following:

1. Go to **Scenario Properties Editor > Supplemental Files**.
2. Set **Application Configuration File** to the name of the application configuration file. See Section 4.2.8.5.1.1 for the format of the Application configuration file.

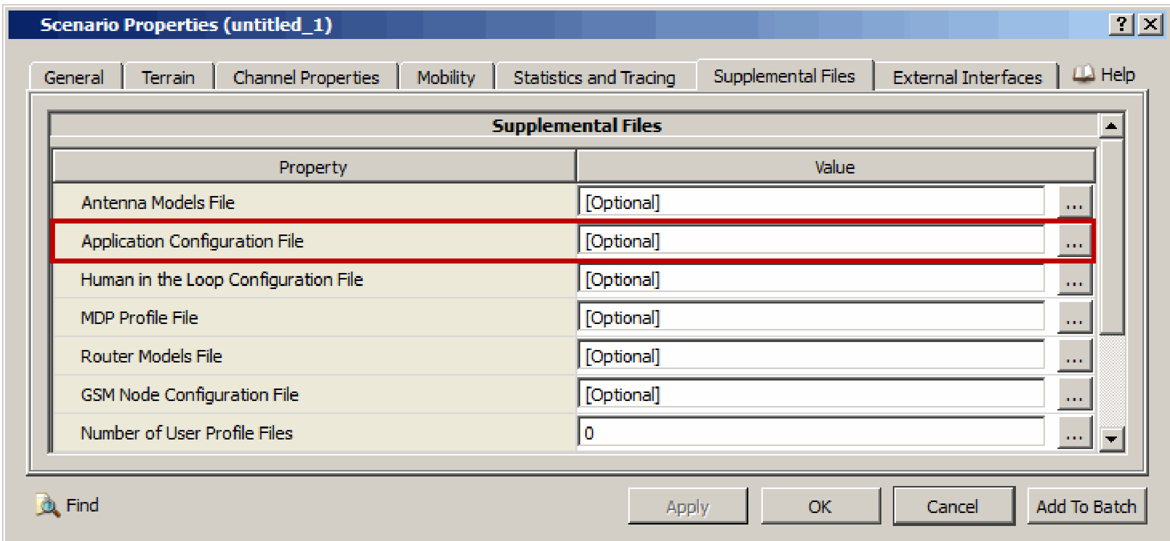


FIGURE 4-51. Specifying the Application Configuration File

TABLE 4-84. Command Line Equivalent of Application Configuration File Parameter

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Application Configuration File	Global	`APP-CONFIG-FILE`

Specifying Client-Server Applications on the Canvas

To set up a client-server application, do the following:

1. Click on the desired application button in the **Applications** toolbar.
2. Click on the source node and drag the mouse to the destination node. A solid green line is drawn from the source to the destination.

   Figure 4-52 shows a FTP session set up from node 1 to node 3.

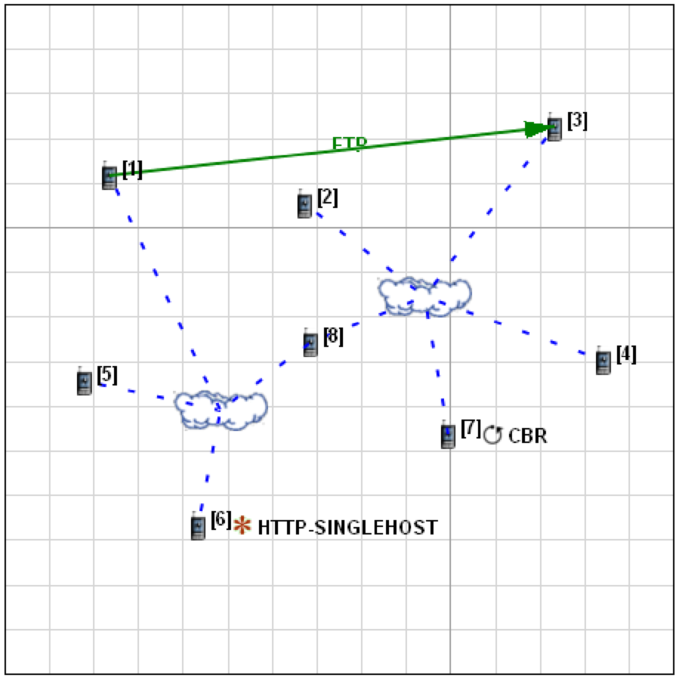


FIGURE 4-52. Types of Applications in a Scenario

3. Set the dependent parameters for the application. See the model library referenced in Table 4-83 for details.

Specifying Single Host Applications on the Canvas

Single host applications can be used to model multicast applications in which traffic is sent from a source node to a group of destination nodes. To configure a single host application, do the following:

1. Click on the desired application button in the **Single Host Applications** toolbar.
2. Click on the host node. A symbol appears next to the host node.

   Figure 4-52 shows a single host application (HTTP) configured at node 6.
3. Set the dependent parameters for the application. See the model library referenced in Table 4-83 for details.

Specifying Loopback Applications on the Canvas

A loopback application is a special type of client-server application in which the source and destination are the same. To set up a loopback application, do the following:

1. Click on the desired application button in the **Applications** toolbar.
2. Click on the node where you want to configure the application. A symbol appears next to the node.

   Figure 4-52 shows a loopback application (CBR) configured at node 7.
3. Set the dependent parameters for the application. See the model library referenced in Table 4-83 for details.

4.2.9 Collecting Statistics

The next set of parameters control statistics collection. Scenario statistics are printed to the statistics (.stat) file at the end of simulation. In addition, detailed statistics for some models can be collected in a statistics database.

4.2.9.1 Statistics File

The statistics file generated by running EXata is named according to the rules described in Section 2.1.1.2. The format of the statistics file is explained in Section 2.3.

This section describes the statistics parameters that determine which statistics are printed to the statistics file for each node.

4.2.9.1.1 Command Line Configuration

Table 4-85 lists the statistics parameters. For the sake of brevity, this table uses a different format than the other parameter tables and should be interpreted as follows:

- Each parameter is optional.
- The possible values of each parameter are: `YES` or `NO`. `YES` indicates that the statistic is enabled for the node (or interface or instance) and `NO` indicates that the statistic is disabled.
- The default value of each parameter is indicated in the second column of the table.
- Each parameter can be specified at the global and node level. In addition, if the parameter can be specified at the subnet and interface level, that is indicated in the third column in the table.
- If the parameter can have multiple instances, that is indicated in the fourth column of the table.
- The last column has the description of the parameter and any conditions that may apply to the parameter.

TABLE 4-85. Statistics Parameters

Parameter	Default Value	Interface/ Subnet Qualifier Allowed?	Instances Allowed?	Description
`ACCESS-LIST-STATISTICS`	`NO`	No	No	Statistics for the access list model.
`APPLICATION-STATISTICS`	`YES`	No	No	Statistics for traffic generators.
`ARP-STATISTICS`	`NO`	Yes	No	Statistics for the Address Resolution Protocol (ARP).
`BATTERY-MODEL-STATISTICS`	`NO`	No	No	Statistics for the battery model.
`CELLULAR-STATISTICS`	`YES`	No	No	Statistics for the abstract cellular models.
`DIFFSERV-EDGE-ROUTER-STATISTICS`	`NO`	No	No	Statistics for the DiffServ Edge router.
`ENERGY-MODEL-STATISTICS`	`YES`	Yes	No	Statistics for the energy model.
`EXTERIOR-GATEWAY-PROTOCOL-STATISTICS`	`YES`	No	No	Statistics for the Exterior Gateway Protocol.
`GSM-STATISTICS`	`NO`	No	No	Statistics for GSM models.
`HOST-STATISTICS`	`NO`	No	No	Host properties, such as the host name, etc.
`ICMP-ERROR-STATISTICS`	`NO`	No	No	Statistics for ICMP error messages.

TABLE 4-85. Statistics Parameters (Continued)

Parameter	Default Value	Interface/ Subnet Qualifier Allowed?	Instances Allowed?	Description
`ICMP-STATISTICS`	`NO`	No	No	Statistics for the Internet Control Message Protocol (ICMP).
`IGMP-STATISTICS`	`NO`	No	No	Statistics for the Internet Group Management Protocol (IGMP).
`INPUT-QUEUE-STATISTICS`	`NO`	Yes	No	Statistics for the input queues.
`INPUT-SCHEDULER-STATISTICS`	`NO`	Yes	No	Statistics for the input schedulers.
`JAMMER-STATISTICS`	`NO`	Yes	No	Statistics for the Jammer model.
`MAC-LAYER-STATISTICS`	`YES`	Yes	No	Statistics for MAC Layer protocols.
`MDP-STATISTICS`	`No`	No	No	Statistics for MDP protocol.
`MOBILE-IP-STATISTICS`	`NO`	No	No	Statistics for the Mobile IP protocol.
`MPLS-LDP-STATISTICS`	`NO`	No	No	Statistics for Multi-Protocol Label Switching (MPLS) Label Distribution Protocol.
`MPLS-STATISTICS`	`NO`	Yes	No	Statistics for MPLS.
`NDP-STATISTICS`	`NO`	No	No	Statistics for the Neighbor Discovery Protocol (NDP).
`NETWORK-LAYER-STATISTICS`	`YES`	No	No	Statistics for Network Layer protocols.
`PHY-LAYER-STATISTICS`	`YES`	Yes	No	Statistics for Physical Layer protocols.
`POLICY-ROUTING-STATISTICS`	`NO`	No	No	Statistics for Policy-based Routing.
`QOSPF-STATISTICS`	`NO`	No	No	Statistics for the QoS Extensions to Open Shortest Path First (QOSPF) protocol.
`QUEUE-STATISTICS`	`YES`	Yes	No	Statistics for the output queues.
`ROUTE-REDISTRIBUTION-STATISTICS`	`NO`	No	No	Statistics for route redistribution.
`ROUTING-STATISTICS`	`YES`	No	No	Statistics for routing protocols.
`RSVP-STATISTICS`	`NO`	No	No	Statistics for the Resource Reservation Protocol (RSVP).
`RTP-STATISTICS`	`NO`	No	No	Statistics for the Real-time Transport Protocol (RTP).
`SCHEDULER-GRAPH-STATISTICS`	`NO`	Yes	No	Statistics for the output scheduler graphs.
`SCHEDULER-STATISTICS`	`NO`	Yes	No	Statistics for the output schedulers.
`SWITCH-PORT-STATISTICS`	`NO`	Yes	Yes	Statistics for a port of the switch.
`SWITCH-QUEUE-STATISTICS`	`NO`	Yes	Yes	Statistics for the switch queues.
`SWITCH-SCHEDULER-STATISTICS`	`NO`	Yes	Yes	Statistics for the switch schedulers.

TABLE 4-85. Statistics Parameters (Continued)

Parameter	Default Value	Interface/ Subnet Qualifier Allowed?	Instances Allowed?	Description
`TCP-STATISTICS`	`YES`	No	No	Statistics for the TCP protocol.
`UDP-STATISTICS`	`YES`	No	No	Statistics for the UDP protocol.
`VOIP-SIGNALLING-STATISTICS`	`NO`	No	No	Statistics for the VoIP Signaling protocol

Statistics Parameter for HLA Interface

In addition to the statistics parameters listed in Table 4-85, there is an additional parameter, `VRLINK-HLA-DYNAMIC-STATISTICS`, for enabling dynamic statistics for an HLA interface. Like the other statistics parameters, this parameter is optional and can take the value `YES` or `NO`. However, this parameter can only be specified at the global level and its default value is `YES`.

4.2.9.1.2 GUI Configuration

To configure statistics collection in the GUI, do the following:

1. Go to one of the following locations:

- To enable statistics collection at the global level, go to **Scenario Properties Editor > Statistics and Tracing > File Statistics**.
- To enable statistics collection for a specific node, go to **Default Device Properties Editor > Node Configuration > File Statistics.**
- To enable statistics collection for a specific interface of a node, go to one of the following locations:
  - **Interface Properties Editor > Interfaces > Interface # > File Statistics.**
  - **Default Device Properties Editor > Interfaces > Interface # > File Statistics**.

In this section, we show how to configure statistics parameters for a node in the **Default Device Properties Editor**. Parameters can be set in the other properties editors in a similar way.

2. Select the statistics to collect by checking or unchecking the appropriate boxes. See Table 4-85 for the list of available statistics.

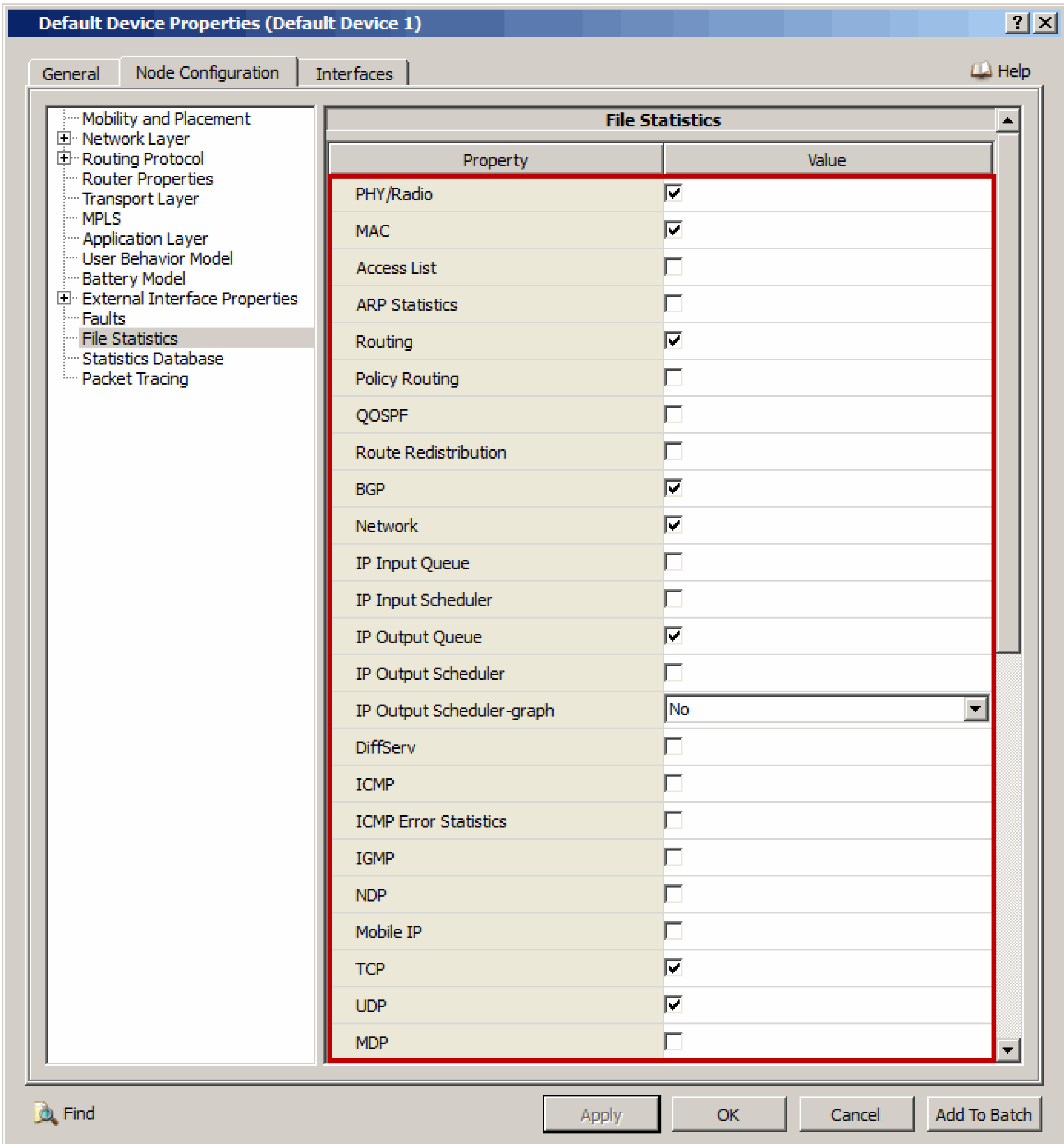


FIGURE 4-53. Enabling Statistics for a Node

TABLE 4-86. Command Line Equivalent of Statistics Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
PHY/Radio	Global, Node, Subnet, Interface	`PHY-LAYER-STATISTICS`
MAC	Global, Node, Subnet, Interface	`MAC-LAYER-STATISTICS`
Access List	Global, Node	`ACCESS-LIST-STATISTICS`

TABLE 4-86. Command Line Equivalent of Statistics Parameters (Continued)

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
ARP Statistics	Global, Node, Subnet, Interface	`ARP-STATISTICS`
Routing	Global, Node	`ROUTING-STATISTICS`
Policy Routing	Global, Node	`POLICY-ROUTING-STATISTICS`
QOSPF	Global, Node	`QOSPF-STATISTICS`
Route Redistribution	Global, Node	`ROUTE-REDISTRIBUTION-STATISTICS`
BGP	Global, Node	`EXTERIOR-GATEWAY-PROTOCOL-STATISTIC`
Network	Global, Node	`NETWORK-LAYER-STATISTICS`
IP Input Queue	Global, Node, Subnet, Interface	`INPUT-QUEUE-STATISTICS`
IP Input Scheduler	Global, Node	`INPUT-SCHEDULER-STATISTICS`
IP Output Queue	Global, Node, Subnet, Interface	`QUEUE-STATISTICS`
IP Output Scheduler	Global, Node, Subnet, Interface	`SCHEDULER-STATISTICS`
IP Output Scheduler-graph	Global, Node, Subnet, Interface	`SCHEDULER-GRAPH-STATISTICS`
DiffServ	Global, Node	`DIFFSERV-EDGE-ROUTER-STATISTICS`
ICMP	Global, Node	`ICMP-STATISTICS`
ICMP Error Statistics	Global, Node	`ICMP-ERROR-STATISTICS`
IGMP	Global, Node	`IGMP-STATISTICS`
NDP	Global, Node	`NDP-STATISTICS`
Mobile IP	Global, Node	`MOBILE-IP-STATISTICS`
TCP	Global, Node	`TCP-STATISTICS`
UDP	Global, Node	`UDP-STATISTICS`
MDP	Global, Node	`MDP-STATISTICS`
RSVP	Global, Node	`RSVP-STATISTICS`
RTP	Global, Node	`RTP-STATISTICS`
Application	Global, Node	`APPLICATION-STATISTICS`
Battery Model	Global, Node	`BATTERY-MODEL-STATISTICS`
Radio Energy Model	Global, Node, Subnet, Interface	`ENERGY-MODEL-STATISTICS`
Cellular	Global, Node	`CELLULAR-STATISTICS`
GSM	Global, Node	`GSM-STATISTICS`
VoIP Signalling	Global, Node	`VOIP-SIGNALLING-STATISTICS`
Switch Port	Global, Node, Subnet, Interface	`SWITCH-PORT-STATISTICS`
Switch Scheduler	Global, Node, Subnet, Interface	`SWITCH-SCHEDULER-STATISTICS`
Switch Queue	Global, Node, Subnet, Interface	`SWITCH-QUEUE-STATISTICS`
MPLS	Global, Node, Subnet, Interface	`MPLS-STATISTICS`
MPLS LDP	Global, Node	`MPLS-LDP-STATISTICS`
Host	Global, Node	`HOST-STATISTICS`
Jammer Statistics	Global, Node, Subnet, Interface	`JAMMER-STATISTICS`

4.2.9.2 Statistics Database

In addition to the file statistics described in Section 4.2.9.1, detailed statistics for some models can be collected in a statistics database.

The statistics database has two types of statistics tables:

- **Scenario Statistics Tables:** These tables contains statistics for the entire scenario
- **Model-specific Statistics Tables:** These tables contains statistics for specific models.

The models listed in Table 4-87 input statistics into various scenario statistics database tables.

TABLE 4-87. Models that Input Data into Scenario Statistics Database Tables

Model	Library
802.3 LAN/Ethernet	Developer
802.11 MAC Protocol	Wireless
802.11a/g PHY Model	Wireless
802.11b PHY Model	Wireless
Abstract Link MAC	Developer
Abstract PHY Model	Wireless
Automatic Model Selection	Urban Propagation
Constant Bit Rate (CBR) Traffic Generator	Developer
Detailed Switch Model	Multimedia and Enterprise
File Transfer Protocol (FTP)	Developer
File Transfer Protocol/Generic (FTP/Generic)	Developer
Forward_App	N/A
HyperText Transfer Protocol (HTTP)	Developer
Internet Group Management Protocol (IGMP)	Developer
Internet Protocol version 4 (IPv4)	Developer
Long Term Evolution (LTE) Layer 2 Model	LTE
Long Term Evolution (LTE) PHY Model	LTE
Microwave Links	Wireless
Multicast Constant Bit Rate (MCBR) Traffic Generator	Developer
Multicast Extensions to OSPF (MOSPF)	Multimedia and Enterprise
Open Shortest Path First version 2 (OSPFv2) Routing Protocol	Multimedia and Enterprise
Protocol Independent Multicast Protocol: Dense Mode (PIM-DM) and Sparse Mode (PIM-SM)	Multimedia and Enterprise
Super Application Traffic Generator	Developer
Traffic Generator (Traffic-Gen)	Developer
Transmission Control Protocol (TCP)	Developer
User Datagram Protocol (UDP)	Developer
Variable Bit Rate (VBR) Traffic Generator	Developer
Voice over Internet Protocol (VoIP)	Multimedia and Enterprise

The models listed in Table 4-88. input data into model-specific database tables.

TABLE 4-88. Models that Generate Model-specific Tables

Model	Library
Automatic Model Selection	Urban Propagation
Internet Group Management Protocol (IGMP)	Developer
Multicast Extensions to OSPF (MOSPF)	Multimedia and Enterprise
Open Shortest Path First version 2 (OSPFv2) Routing Protocol	Multimedia and Enterprise
Protocol Independent Multicast Protocol: Dense Mode (PIM-DM) and Sparse Mode (PIM-SM)	Multimedia and Enterprise

Refer to *EXata Statistics Database User's Guide* for a description of the statistics tables in the database and the parameters for configuring them.

4.2.10 Tracing Packet Headers

EXata provides tracing capabilities that enable a packet to be traced as it traverses the protocol stack at each node on the path from the source to the destination. Parameters that control the trace information printed are explained in this section.

The trace file generated by running EXata has the extension ".trace" and is named according to the rules described in Section 2.1.1.2. The format of the trace file is a little involved and is explained in *EXata Programmer's Guide*.

4.2.10.1 Command Line Configuration

To enable packet tracing in the command line interface, include the parameters listed in Table 4-89 and Table 4-90 in the scenario configuration (.config) file. Table 4-89 lists the higher-level trace parameters. Trace parameters for individual protocols are listed in Table 4-90.

TABLE 4-89. Trace Parameters

Parameter	Value	Description
`PACKET-TRACE` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `NO`	Indicates whether packet tracing is enabled. `YES` : Packet tracing is enabled. `NO` : Packet tracing is disabled.
`TRACE-ALL` *Optional* *Scope:* Global, Node	List: • `YES` • `NO` *Default:* `NO`	Indicates whether tracing is enabled for all protocols. `YES` : Packet tracing is enabled for all protocols. `NO` : Packet tracing is not enabled for all protocols.
`TRACE-DIRECTION` *Optional* *Scope:* Global, Node	List: • `INPUT` • `OUTPUT` • `BOTH` *Default:* `BOTH`	Indicates whether tracing is enabled for incoming packets, outgoing packets, or both. `INPUT` : Only incoming packets are traced. `OUTPUT` : Only outgoing packets are traced. `BOTH` : Both incoming and outgoing packets are traced.

TABLE 4-89. Trace Parameters (Continued)

Parameter	Value	Description
TRACE-INCLUDED-HEADERS *Optional* *Scope:* Global, Node	List: • ALL • SELECTED • NONE *Default:* NONE	Indicates whether upper layer headers are included in the trace. The use of this parameter is explained in Section 4.2.10.1.1.
TRACE-APPLICATION-LAYER *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* YES	Indicates whether tracing is enabled at the Application Layer. YES : Packet tracing is enabled at the Application Layer. NO : Packet tracing is disabled at the Application Layer. If this parameter is included, then it takes precedence over parameter TRACE-ALL.
TRACE-TRANSPORT-LAYER *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* YES	Indicates whether tracing is enabled at the Transport Layer. YES : Packet tracing is enabled at the Transport Layer. NO : Packet tracing is disabled at the Transport Layer. If this parameter is included, then it takes precedence over parameter TRACE-ALL.
TRACE-NETWORK-LAYER *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* YES	Indicates whether tracing is enabled at the Network Layer. YES : Packet tracing is enabled at the Network Layer. NO : Packet tracing is disabled at the Network Layer. If this parameter is included, then it takes precedence over parameter TRACE-ALL.
TRACE-MAC-LAYER *Optional* *Scope:* Global, Node	List: • YES • NO *Default:* YES	Indicates whether tracing is enabled at the MAC Layer. YES : Packet tracing is enabled at the MAC Layer. NO : Packet tracing is disabled at the MAC Layer. If this parameter is included, then it takes precedence over parameter TRACE-ALL.

Whether tracing is enabled at a layer is determined by the parameter TRACE-ALL and the trace parameter for the layer. By default, TRACE-ALL is NO and all the layer trace parameters are YES. To selectively enable tracing at specific layers, do one of the following:

- Set TRACE-ALL to YES and selectively disable tracing at a specific layer by setting the corresponding parameter to NO.
- Set TRACE-ALL to NO and selectively enable tracing at a specific layer by setting the corresponding parameter to YES.

Table 4-90 lists the packet tracing parameters for protocols. For the sake of brevity, this table uses a different format than the other parameter tables and should be interpreted as follows:

- The name of the parameter indicates the protocol that it is applicable to, e.g., `TRACE-CBR` is the trace parameter for the Constant Bit Rate (CBR) protocol. The second column in the table lists the layer at which the protocol operates.
- Each parameter is optional.
- The possible values of each parameter are: `YES` or `NO`. If this parameter is set to `NO` or if tracing is disabled for the protocol's layer (as explained above), then tracing is disabled for the protocol; otherwise, tracing is enabled for the protocol.
- The default value of each parameter is `NO`.
- Each parameter can be specified at the global and node level.
- Instances are not allowed for any of the parameters.

TABLE 4-90. Protocol Trace Parameters

Parameter	Protocol Layer
`TRACE-BELLMANFORD`	Application
`TRACE-CBR`	Application
`TRACE-GEN-FTP`	Application
`TRACE-OLSR`	Application
`TRACE-RIPNG`	Application
`TRACE-SUPERAPPLICATION`	Application
`TRACE-TRAFFIC-GEN`	Application
`TRACE-ZIGBEEAPP`	Application
`TRACE-TCP`	Transport
`TRACE-UDP`	Transport
`TRACE-AODV`	Network
`TRACE-DYMO`	Network
`TRACE-ICMP`	Network
`TRACE-ICMPV6`	Network
`TRACE-IP`	Network
`TRACE-IPV6`	Network
`TRACE-ODMRP`	Network
`TRACE-OSPFv2`	Network
`TRACE-OSPFv3`	Network

4.2.10.1.1 Headers Printed in a Trace Record

The trace information is printed to a trace file in the form of trace records. Each trace record includes information such as the originating node ID, message sequence number, simulation time, and one or more headers in the packet.

The following rules determine which headers are included in a trace record:

- The header of the protocol that initiates the trace (*initiating protocol*) is always printed.

- An upper layer header that is included in the packet is printed only if at least one of the following conditions is satisfied:
  - Parameter `INCLUDED-HEADER` is set to `SELECTED` and tracing is enabled for the upper layer protocol
  - Parameter `INCLUDED-HEADER` is set to `ALL` and tracing is enabled for the initiating protocol

4.2.10.2 GUI Configuration

In the GUI, packet tracing can be enabled only at the global level. The other packet tracing parameters can be set at the global and node levels.

To configure packet tracing parameters, do the following:

1. Go to one of the following locations:

- To configure packet tracing parameters at the global level, go to **Scenario Properties Editor > Statistics and Tracing > Packet Tracing**.
- To configure packet tracing parameters a specific node, go to **Default Device Properties Editor > Node Configuration > Packet Tracing.**

In this section, we show how to configure packet tracing parameters in the **Scenario Properties Editor**. Parameters can be set in the other properties editors in a similar way.

2. To enable packet tracing, set **Enable Packet Tracing** to *Yes* and set the dependent parameters listed in Table 4-91.

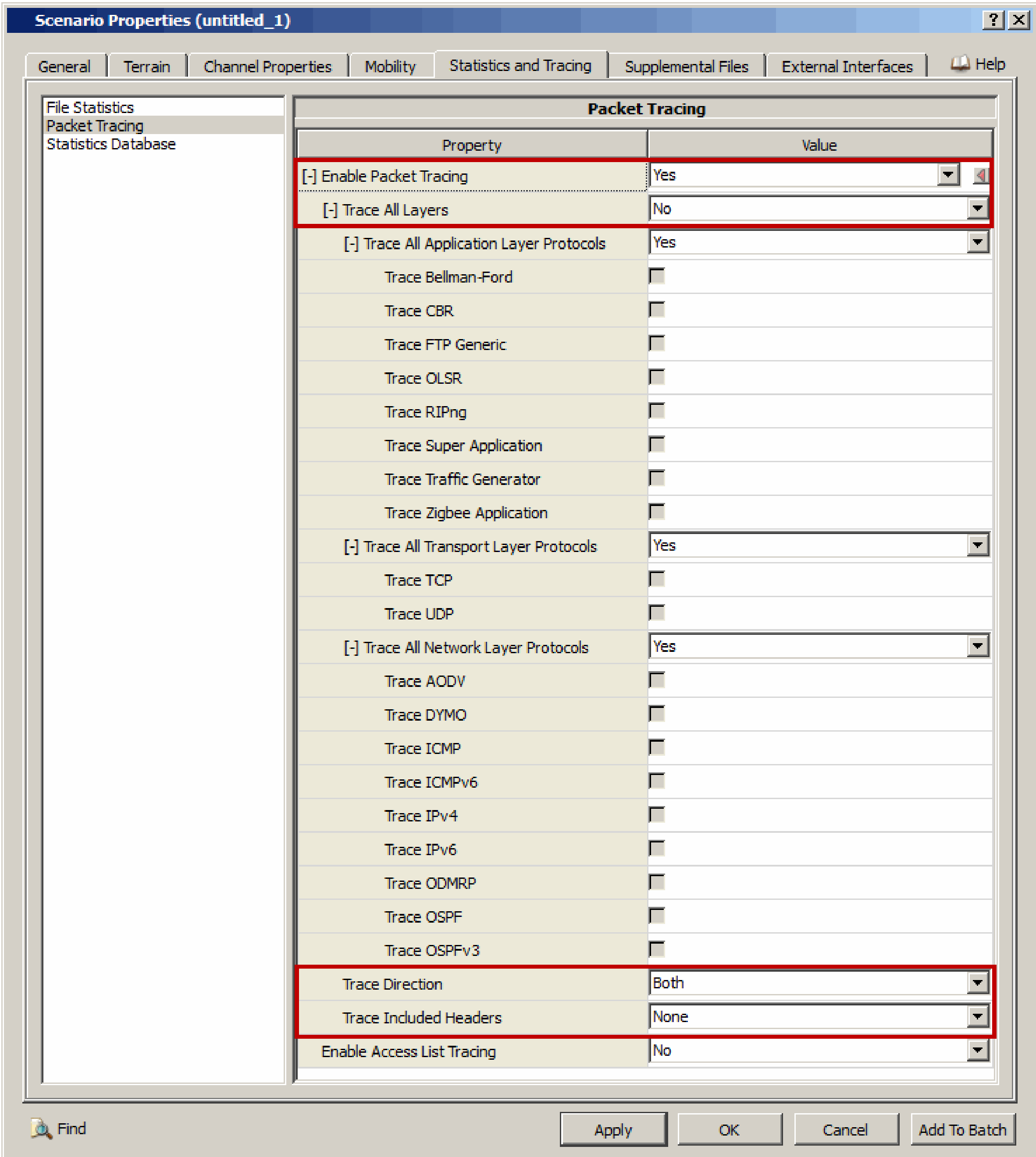


FIGURE 4-54. Enabling Packet Tracing

TABLE 4-91. Command Line Equivalent of Packet Tracing Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Enable Packet Tracing	Global	`PACKET-TRACE`
Trace All Layers	Global, Node	`TRACE-ALL`
Trace Direction	Global, Node	`TRACE-DIRECTION`
Trace Included Headers	Global, Node	`TRACE-INCLUDED-HEADERS`

3. To selectively enable packet tracing for individual layers, set **Trace All** to *No* and set the dependent parameters listed in Table 4-92.

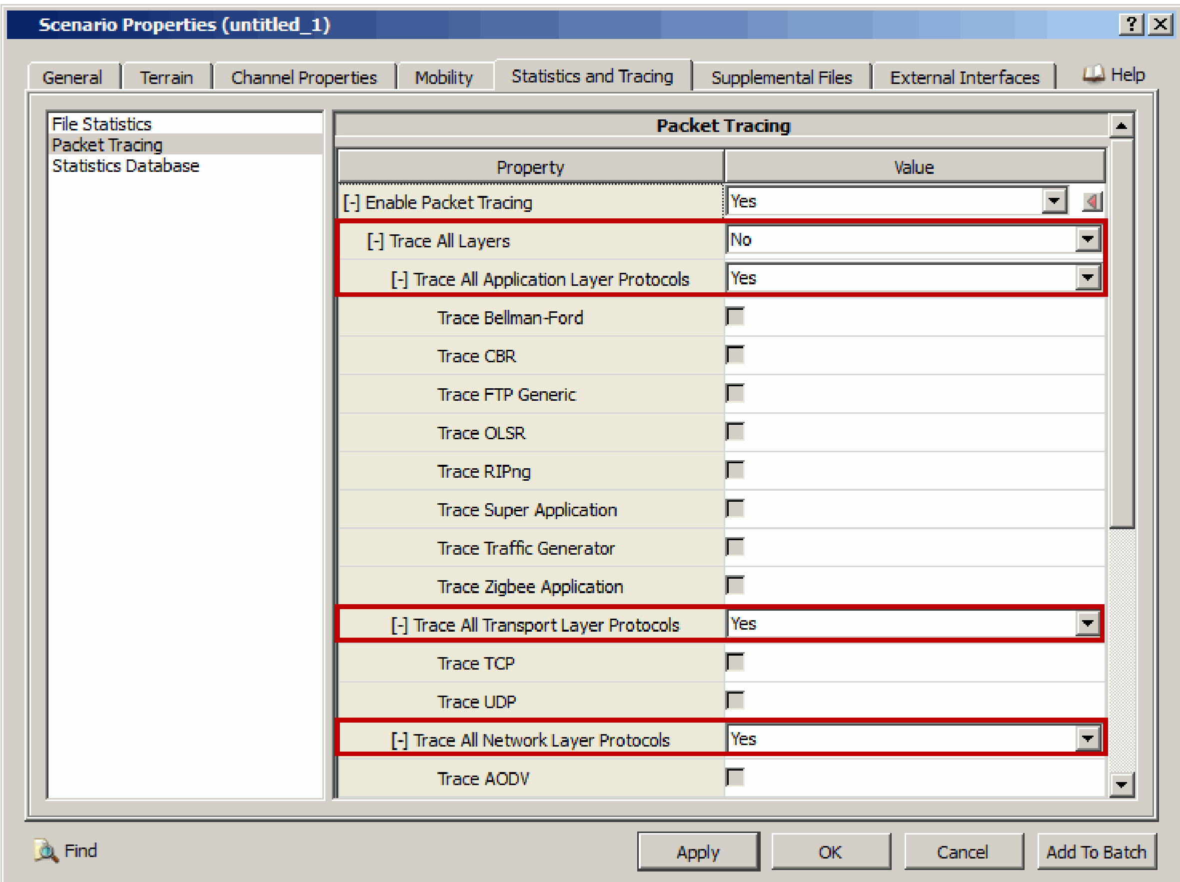


FIGURE 4-55. Enabling Packet Tracing for Specific Layers

TABLE 4-92. Command Line Equivalent of Packet Tracing Parameters for Specific Layers

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Trace All Application Layer Protocols	Global, Node	`TRACE-APPLICATION-LAYER`
Trace All Transport Layer Protocols	Global, Node	`TRACE-TRANSPORT-LAYER`
Trace All Network Layer Protocols	Global, Node	`TRACE-NETWORK-LAYER`

4. To selectively enable packet tracing for a protocol, disable packet tracing of all protocols at the layer and enable packet tracing for the specific protocol by checking the appropriate box. For example, to enable packet tracing for CBR and disable packet tracing for all other application layer protocols, set **Trace All Application Layer Protocols** to *No* and check the box **Trace CBR**.

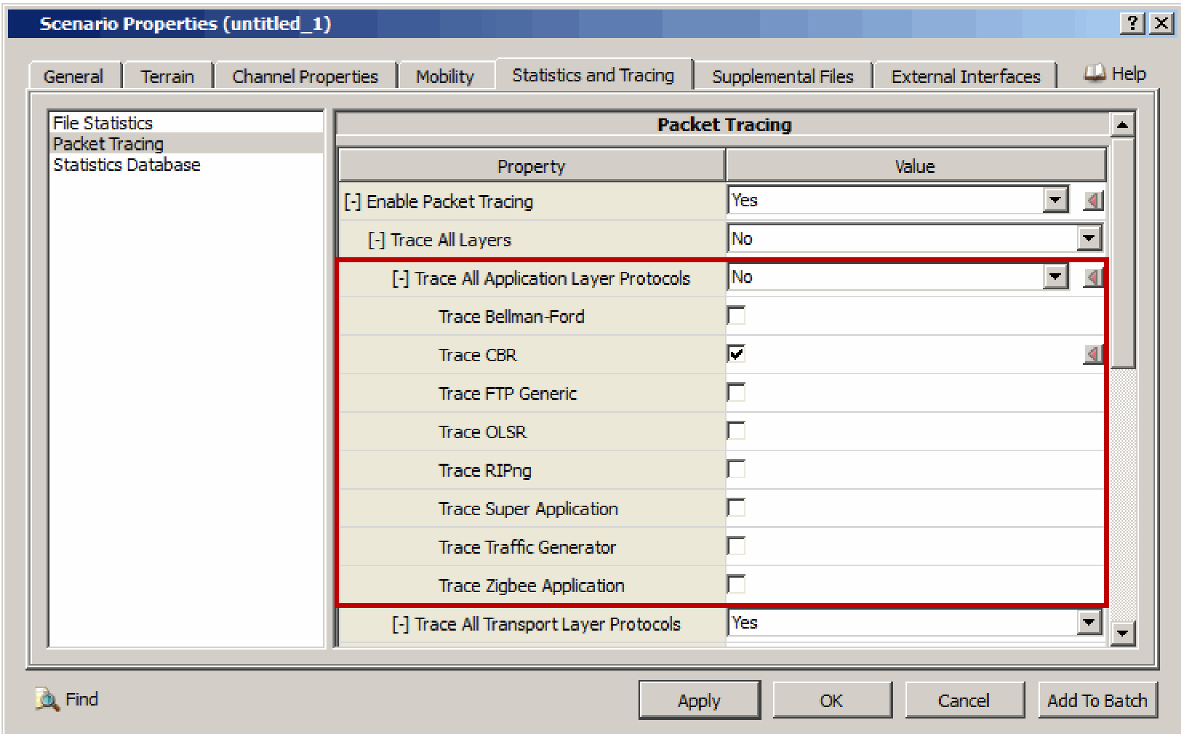


FIGURE 4-56. Enable Packet Tracing for Specific Protocols

Packet tracing for the other protocols can be selectively enabled in a similar way.

TABLE 4-93. Command Line Equivalent of Protocol Trace Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Trace Bellman-Ford	Global, Node	`TRACE-BELLMANFORD`
Trace CBR	Global, Node	`TRACE-CBR`
Trace FTP Generic	Global, Node	`TRACE-GEN-FTP`
Trace OLSR	Global, Node	`TRACE-OLSR`
Trace RIPng	Global, Node	`TRACE-RIPNG`
Trace Super Application	Global, Node	`TRACE-SUPERAPPLICATION`
Trace Traffic Generator	Global, Node	`TRACE-TRAFFIC-GEN`
Trace ZigBee Application	Global, Node	`TRACE-ZIGBEEAPP`
Trace TCP	Global, Node	`TRACE-TCP`
Trace UDP	Global, Node	`TRACE-UDP`
Trace AODV	Global, Node	`TRACE-AODV`
Trace DYMO	Global, Node	`TRACE-DYMO`

TABLE 4-93. Command Line Equivalent of Protocol Trace Parameters (Continued)

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Trace ICMP	Global, Node	`TRACE-ICMP`
Trace ICMPv6	Global, Node	`TRACE-ICMPV6`
Trace IPv4	Global, Node	`TRACE-IP`
Trace IPv6	Global, Node	`TRACE-IPV6`
Trace ODMRP	Global, Node	`TRACE-ODMRP`
Trace OSPF	Global, Node	`TRACE-OSPFv2`
Trace OSPFv3	Global, Node	`TRACE-OSPFv3`

4.2.11 Enabling Runtime Features

Architect provides features for interacting with the simulator during scenario execution and for displaying dynamic information that is useful for analyzing scenarios. This section describes how to configure parameters to enable these features.

4.2.11.1 Dynamic Parameters

Some models in EXata implement *dynamic* parameters to enable interaction with the simulator during run time. Values of dynamic parameters can be observed and/or modified during the simulation. Some dynamic parameters are read-only, i.e., the user can watch their values (for example, IP fragmentation unit size). Other dynamic parameters can be modified by the user during the simulation (for example, transmission power). Values of dynamic parameters can be observed and/or modified in Visualize mode of Architect, as described in Section 6.5.4.2 and Section 6.6.4.

This section describes how to enable dynamic parameters.

4.2.11.1.1 Command Line Configuration

To enable dynamic parameters in the command line interface, include the parameters listed in Table 4-94 in the scenario configuration (.config) file.

TABLE 4-94. Dynamic Parameters Configuration Parameters

Parameter	Value	Description
`DYNAMIC-ENABLED` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `NO`	Indicates whether dynamic parameters are enabled for the scenario.
`DYNAMIC-PARTITION-ENABLED` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Indicates whether partition level dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both `DYNAMIC-ENABLED` and `DYNAMIC-PARTITION-ENABLED` should be `YES`.
`DYNAMIC-NODE-ENABLED` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Indicates whether node level dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both `DYNAMIC-ENABLED` and `DYNAMIC-NODE-ENABLED` should be `YES`.

TABLE 4-94. Dynamic Parameters Configuration Parameters (Continued)

Parameter	Value	Description
DYNAMIC-PHY-ENABLED *Optional* *Scope:* Global	List: • YES • NO *Default:* YES	Indicates whether physical layer dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both DYNAMIC-ENABLED and DYNAMIC-PHY-ENABLED should be YES.
DYNAMIC-MAC-ENABLED *Optional* *Scope:* Global	List: • YES • NO *Default:* YES	Indicates whether MAC layer dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both DYNAMIC-ENABLED and DYNAMIC-MAC-ENABLED should be YES.
DYNAMIC-NETWORK-ENABLED *Optional* *Scope:* Global	List: • YES • NO *Default:* YES	Indicates whether network layer dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both DYNAMIC-ENABLED and DYNAMIC-NETWORK-ENABLED should be YES.
DYNAMIC-TRANSPORT-ENABLED *Optional* *Scope:* Global	List: • YES • NO *Default:* YES	Indicates whether transport layer dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both DYNAMIC-ENABLED and DYNAMIC-TRANSPORT-ENABLED should be YES.
DYNAMIC-APP-ENABLED *Optional* *Scope:* Global	List: • YES • NO *Default:* YES	Indicates whether application layer dynamic parameters are enabled for the scenario. **Note:** For partition level dynamic parameters to be enabled, both DYNAMIC-ENABLED and DYNAMIC-APP-ENABLED should be YES.

4.2.11.1.2 GUI Configuration

To enable dynamic parameters in the GUI, do the following:

1. Go to **Scenario Properties Editor > General > Advanced Settings**.
2. Set **Enable Dynamic Parameters** to *Yes* and set the dependent parameters listed in Table 4-95.

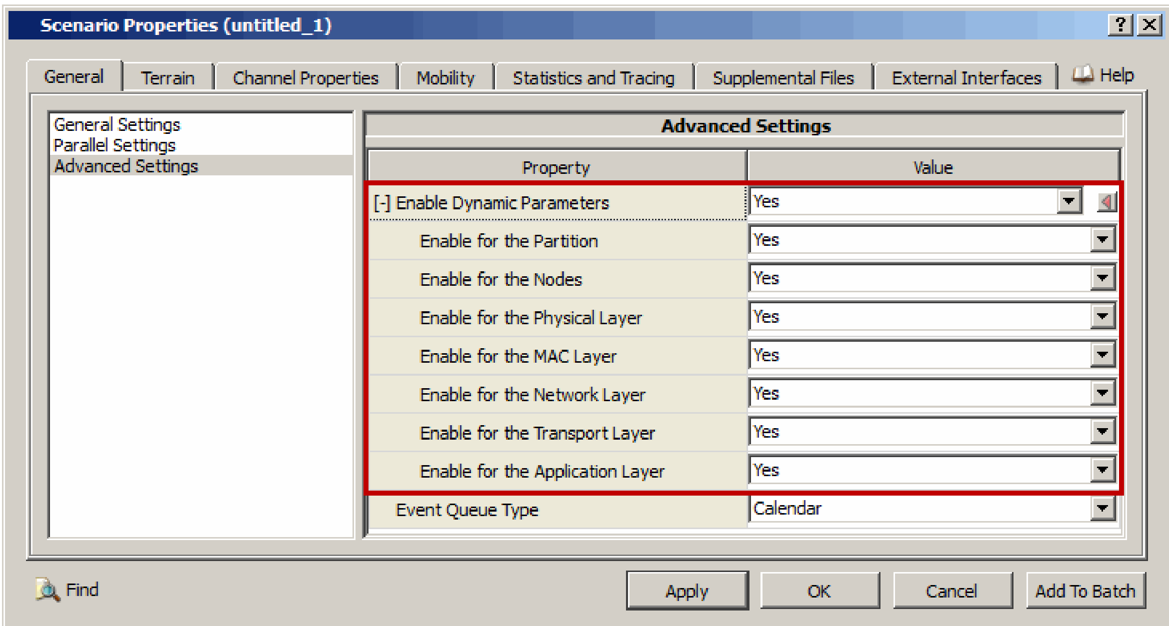


FIGURE 4-57. Enabling Dynamic Parameters

TABLE 4-95. Command Line Equivalent of Dynamic Parameters Configuration Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Enable Dynamic Parameters	Global	DYNAMIC-ENABLED
Enable for the Partition	Global	DYNAMIC-PARTITION-ENABLED
Enable for the Nodes	Global	DYNAMIC-NODE-ENABLED
Enable for the Physical Layer	Global	DYNAMIC-PHY-ENABLED
Enable for the MAC Layer	Global	DYNAMIC-MAC-ENABLED
Enable for the Network Layer	Global	DYNAMIC-NETWORK-ENABLED
Enable for the Transport Layer	Global	DYNAMIC-TRASNPORT-ENABLED
Enable for the Application Layer	Global	DYNAMIC-APP-ENABLED

4.2.11.2 Human-in-the-loop Commands

EXata implements Human-In-The-Loop (HITL) commands to interact with the simulator during scenario execution, such as activate/deactivate nodes and change traffic characteristics. These commands are sent to the simulator over the socket using the HITL interface (see Section 6.5.3.3). The effect of HITL commands can also be simulated by specifying the commands in a file along with the time when each command is to be sent to the simulator.

This section describes how to simulate HITL commands using a file.

4.2.11.2.1 Command Line Configuration

To specify the name of the file containing the HITL commands, include the parameters listed in Table 4-96 in the scenario configuration (.config) file.

TABLE 4-96. HITL Configuration File Parameters

Parameter	Value	Description
`HITL-CONFIG-FILE` *Optional* *Scope:* Global	Filename	Name of the HITL configuration file. The format of this file is described in Section 4.2.11.2.1.1. **Note:** The extension of this file must be “hitl”.

4.2.11.2.1.1 Format of the HITL Configuration File

Each line in the HITL configuration file specifies an HITL command and the time when that command is to be sent to the simulator. Each line has the following format:

```
<time>   <command>
```

where

`<time>` Time when the command is sent to the simulator.

`<command>` HITL command and its associated parameters.

The available HITL commands are described in Table 4-97.

Note: Some model libraries support additional HITL commands. These are described in the model library documentation.

TABLE 4-97. Human-In-The-Loop Commands

Command	Description
`D <node ID>`	This command deactivates the node with specified node ID. A deactivated node can not communicate with other nodes. Example: `D 11` The example above deactivates the node with ID 11. **Note:** When a scenario is run in Architect’s Visualize mode (see Chapter 6), a red circle with a diagonal appears next to a deactivated node indicating that all interfaces of the node have failed.
`A <node ID>`	This command activates the node with the specified node ID. An activated node can communicate with other nodes again. Example: `A 11` The example above activates the node with ID 11.
`P <priority>`	This command changes the precedence of all CBR sessions in the scenario to the specified priority. `<priority>` should be an integer in t he range 0 ≤ `<priority>` ≤ 7. See *Developer Model Library* for details of CBR. Example: `P 3` The above example changes the precedence of all CBR sessions to 3.

TABLE 4-97. Human-In-The-Loop Commands (Continued)

Command	Description
`T <interval>`	This command changes the rate of all CBR sessions in the scenario by changing the CBR inter-packet interval. The new inter-packet interval is equal to `<interval>` milliseconds. `<interval>` should be an integer or real value. Example: The command `T 30` changes the inter-packet interval for all CBR sessions to 30 milliseconds.
`L <rate-factor>`	This command changes the rate of all CBR sessions in the scenario by changing the CBR inter-packet interval. The new inter-packet interval is equal to the product of the current inter-packet interval and `<rate-factor>`. Example: If the current interval is 0.1 second, the command `L 0.1` changes the interval to 0.01 second (= current interval * `<rate-factor>`).

Example:

The following lines show a segment of an HITL configuration file:

```
50S  D 5
100S A 5
200S T 100
250S L 0.2
```

4.2.11.2.2 GUI Configuration

To specify the HITL configuration file in the GUI, do the following:

1. Go to **Scenario Properties Editor > Supplemental Files**.
2. Set **Human in the Loop Configuration File** to the name of the HITL configuration file.

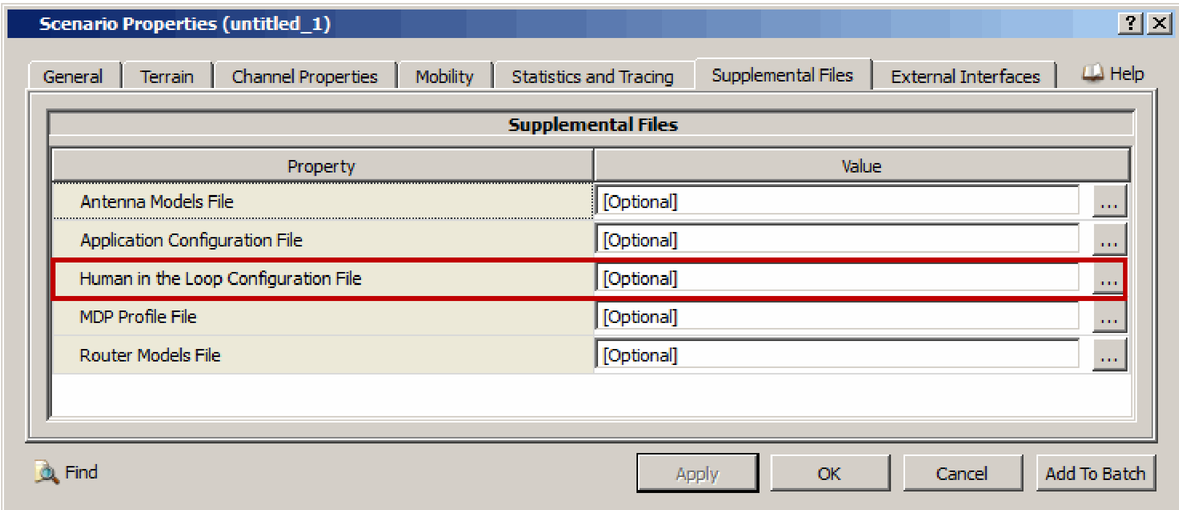


FIGURE 4-58. Specifying HITL Configuration File

TABLE 4-98. Command Line Equivalent of HITL Configuration File Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Human in the Loop Configuration File	Global	`HITL-CONFIG-FILE`

4.3 Configuring the Emulation Environment

This section describes how to configure the emulation features of EXata in command line and using the EXata GUI.

In the EXata GUI, the emulation features are available only in the Emulation mode. The Emulation mode can be enabled from the **Emulation** toolbar.

FIGURE 4-59. Enabling Emulation Mode in EXata GUI

Notes: **1.** In EXata GUI, the execution mode (emulation or simulation) can be selected only in Design mode of Architect.

2. If emulation mode is enabled, then some emulation features can be configured in both Design and Visualize modes, while the other emulation features can be configured only in Design mode.

4.3.1 External Node Mapping

The External Node Mapping creates a mapping between a network interface on an EXata node and a physical operational node. The EXata node that is mapped to an operational host is referred to as the *External Node*. Any traffic originating from the operational host is inserted at the EXata external node, and similarly, any traffic received at the EXata external node is forwarded to the operational host.

External node mappings can be created to set up a test-bed that consists of both simulated nodes (within EXata) and physical nodes (operational hosts).

4.3.1.1 Command Line Configuration

To create a mapping between external and operational hosts for the command line interface, include the parameters listed in Table 4-99 in the scenario configuration (.config) file.

TABLE 4-99. External Node Mapping Parameters

Parameter	Value	Description
`EXATA-EXTERNAL-NODE` *Optional* *Scope:* Interface	IPv4 Address	Address of the operational host to which this interface is mapped.

For example, the following line creates a mapping between the EXata node with the interface address 192.168.0.1 and an operational host with the address 20.100.100.45.

```
[192.168.0.1] EXATA-EXTERNAL-NODE 20.100.100.45
```

The interface address 192.168.0.1 exists within the EXata scenario, and does not correspond to any physical entity. The physical address 20.100.100.45, on the other hand, exists as a physical address on a physical entity, and does not correspond to any EXata node in the scenario.

4.3.1.2 GUI Configuration

External node mappings can be created with the Connection Manager (see Section 5.1.2.1) or by using the mapping editor in EXata GUI. Configuration through the Connection Manager offers additional benefits, such as running multiple applications on different external nodes, which are not available from the EXata GUI mapping editor. It is recommended that the EXata GUI mapping editor only be used for operational hosts that can not run Connection Manager.

To create a mapping between external and operational hosts in EXata GUI, do the following:

1. Launch the mapping editor by clicking the button on the **Emulation** toolbar.

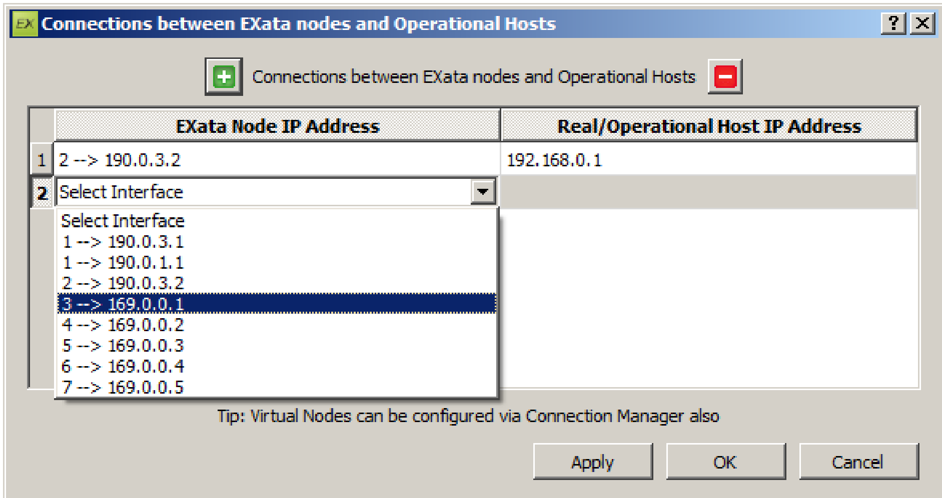


FIGURE 4-60. External Node Mapping Editor

2. To create a new mapping, click the button.
3. From the pull-down menu in the **EXata Node IP Address** column, select the EXata node.
4. In the **Real/Operational Host IP Address** column, enter the IP address of the operational host which is to be mapped to the selected EXata node.
5. To delete a mapping, select the mapping and click the button.

Note: External nodes can be mapped in both Design and Visualize modes.

4.3.2 Internet Gateway Configuration

One of the nodes in the scenario can be configured to be the *Internet Gateway* node, which serves as a gateway between the scenario and the Internet. This feature allows applications running on operational hosts to connect to the Internet or other external networks. Typically, the traffic originating from applications (for example, a web browser) at the operational host is inserted into the EXata emulator where it undergoes the dynamics of the model scenario and is then sent out to the Internet or the external network.

See Section 5.4 for details of Internet Gateway capabilities offered by EXata.

4.3.2.1 Command Line Configuration

To configure the Internet gateway node for the command line interface, include the parameters listed in Table 4-100 in the scenario configuration (.config) file.

TABLE 4-100. Internet Gateway Parameters

Parameter	Value	Description
`INTERNET-GATEWAY` *Optional* *Scope:* Global	Integer *Range:* ≥ 0 *Default:* 0	ID of the node that serves as the Internet Gateway. A value of 0 indicates that the Internet gateway feature is disabled.

4.3.2.2 GUI Configuration

To configure the Internet gateway node in the GUI, do the following:

1. Launch the **Advanced Emulation Configuration** editor by clicking the button on the **Emulation** toolbar and go to the **Internet Gateway** tab.

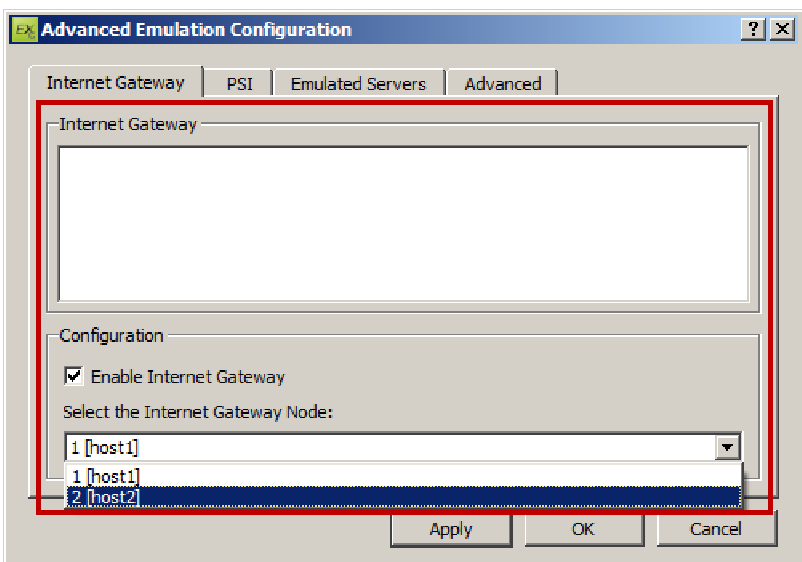


FIGURE 4-61. Configuring Internet Gateway

2. Check the box labeled **Enable Internet Gateway**.
3. From the pull-down menu, select the node that is to serve as the Internet gateway.

Note: Internet gateway can be configured only in Design mode.

4.3.3 Packet Sniffing Interface Configuration

EXata offers the capability of making the traffic inside a scenario visible to an external packet sniffing software (such as Wireshark or Microsoft Network Monitor). The packet sniffing software can thus examine the packets that are transmitted and received within the emulated scenario.

See Section 5.6 for details of Packet Sniffing capabilities offered by EXata.

4.3.3.1 Command Line Configuration

To configure the packet sniffing interface for the command line interface, include the parameters listed in Table 4-101 in the scenario configuration (.config) file.

TABLE 4-101. Packet Sniffing Interface Parameters

Parameter	Value	Description
`PACKET-SNIFFER-NODE` *Optional* *Scope:* Global	Integer *Range:* ≥ `0` *Default:* `0`	ID of the node for which packet sniffing is enabled. Packets transmitted or received by this node are made visible to the external sniffing software. A value of `0` indicates that picket sniffing is enabled for all nodes in the network. If this parameter is not included, then packet sniffing is disabled for all nodes.
`PACKET-SNIFFER-DOT11` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `NO`	Specifies whether 802.11 MAC frames should be displayed on a separate virtual network interface. **Note:** This feature is available only on Linux systems.
`PACKET-SNIFFER-ENABLE-APP` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Specifies whether application packets should be are sniffed.
`PACKET-SNIFFER-ENABLE-TCP` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Specifies whether TCP packets should be sniffed.
`PACKET-SNIFFER-ENABLE-UDP` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Specifies whether UDP packets should be sniffed.
`PACKET-SNIFFER-ENABLE-ROUTING` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Specifies whether routing packets should be sniffed.
`PACKET-SNIFFER-ENABLE-MAC` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `YES`	Specifies whether MAC layer packets should be sniffed.

4.3.3.2 GUI Configuration

To configure the packet sniffing interface in the GUI, do the following:

1. Click the  button on the **Emulation** toolbar. This enables a pull-down menu.
2. From the pull-down menu, select the node(s) to sniff on. You can select any one node or all nodes in the scenario.

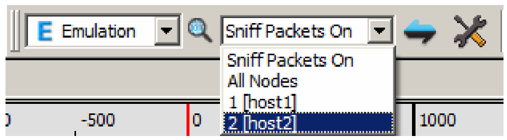


FIGURE 4-62. Selecting Node(s) for Packet Sniffing

3. Launch the **Advanced Emulation Configuration** editor by clicking the  button on the **Emulation** toolbar and go to the **PSI** tab.

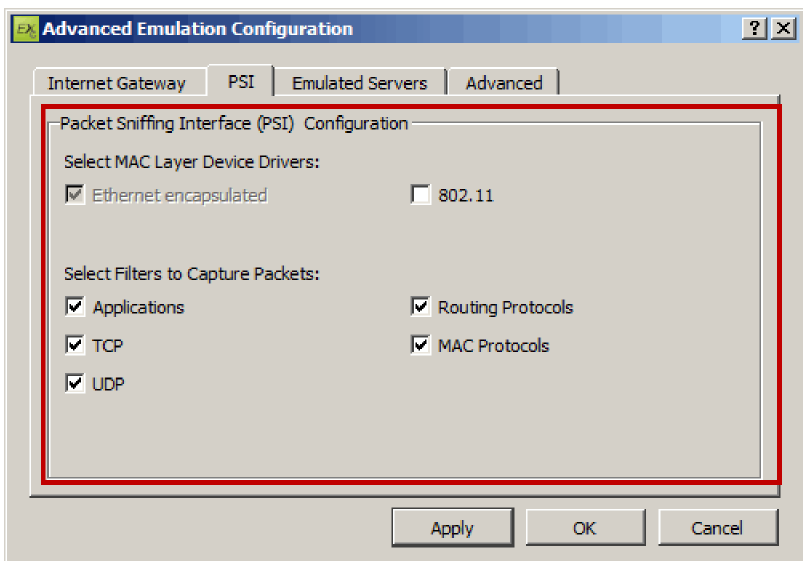


FIGURE 4-63. Configuring Packet Sniffing Interface

4. Select the options by checking the appropriate boxes.

Notes: **1.** The node(s) to sniff on can be selected in both Design and Visualize modes.

2. The packet sniffing interface options can be selected in both Design and Visualize modes.

4.3.4 Emulated Server Configuration

All nodes in an EXata scenario run the FTP server, HTTP server and Telnet server applications at the respective port numbers. Any client application on the operational host can connect to these servers to initiate a session and interact with servers (for example, download a file from FTP, view a web page from the HTTP server, etc.).

EXata allows the user to configure the set of files that are exported by the FTP and HTTP servers. That is, a user may configure a directory on the local computer (running EXata) to serve as the root folder for the FTP and HTTP servers.

4.3.4.1 Command Line Configuration

To configure the emulated server parameters for the command line interface, include the parameters listed in Table 4-102 in the scenario configuration (.config) file.

TABLE 4-102. Emulated Server Parameters

Parameter	Value	Description
`FTP-ROOT-DIRECTORY` *Optional* *Scope:* Global	Path (see note)	Path to the FTP root directory. Files in this directory are exported to FTP clients. **Note:** If this parameter is not specified, then an *in-memory* image of the file system is exported.
`HTTP-ROOT-DIRECTORY` *Optional* *Scope:* Global	Path (see note)	Path to the HTTP root directory. Files in this directory are exported to HTTP clients. **Note:** if this parameter is not specified, then the directory EXATA_HOME/data/webfiles is used as the HTTP root directory.

Note: The path to the FTP or HTTP root directory can be specified as an absolute path, a path relative to the current directory, or by using the environment variable for the EXata installation directory (`$EXATA_HOME` on both Windows and Linux systems).

Following are examples of HTTP root directory paths:

`C:\scalable\exata\5.1\data`	(For Windows)
`/root/scalable/exata/5.1/data`	(For UNIX)
`../../data/terrain/los-angeles-w`	(For Windows and UNIX)
`$EXATA_HOME/data`	(For Windows and UNIX)

4.3.4.2 GUI Configuration

To configure the emulated server parameters in the GUI, do the following:

1. Launch the **Advanced Emulation Configuration** editor by clicking the  button on the **Emulation** toolbar and go to the **Emulated Servers** tab.

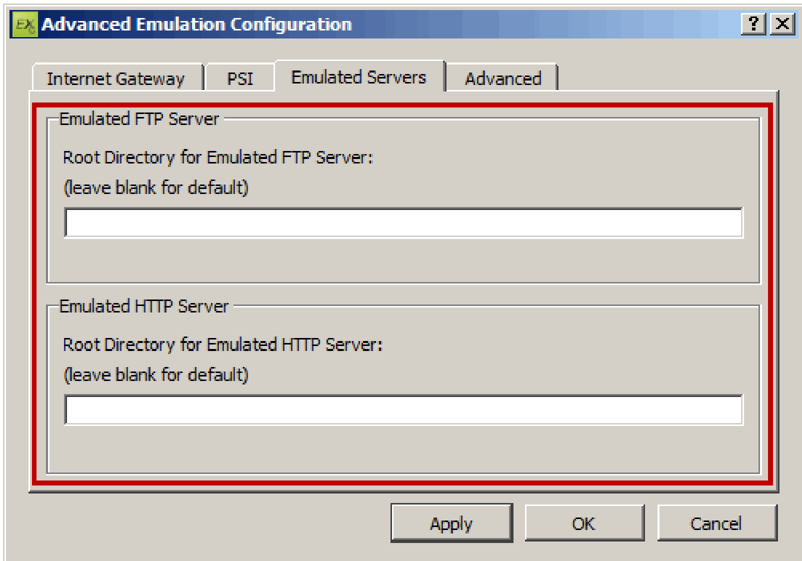


FIGURE 4-64. Configuring Emulated Servers

Note: The path to the FTP or HTTP root directory can be specified as an absolute path, a path relative to the current directory, or by using the environment variable for the EXata installation directory (`$EXATA_HOME` on both Windows and Linux systems).

Following are examples of HTTP root directory paths:

`C:\scalable\exata\5.1\data`	(For Windows)
`/root/scalable/exata/5.1/data`	(For UNIX)
`../../data/terrain/los-angeles-w`	(For Windows and UNIX)
`$EXATA_HOME/data`	(For Windows and UNIX)

2. Enter the path to the FTP and HTTP root directories in the appropriate boxes.

Note: Emulated servers can be configured only in Design mode.

4.3.5 Advanced Emulation Features

This section describes some advanced emulation features and how to configure them.

4.3.5.1 Debugging

EXata provides debugging information to review the connection and traffic between the operational hosts and the emulated network. The level of detail of debugging information can be configured.

Section 4.3.5.1.1 describes how to configure debugging parameters. Section 4.3.5.1.2 describes the debugging information that is displayed during an emulation run.

4.3.5.1.1 Configuring Debugging Parameters

This section describes how to configure debugging parameters for the command line (see Section 4.3.5.1.1.1) and in the GUI (see Section 4.3.5.1.1.2).

4.3.5.1.1.1 Command Line Configuration

To configure the debugging parameters for the command line interface, include the parameters listed in Table 4-103 in the scenario configuration (.config) file.

TABLE 4-103. Debugging Parameters

Parameter	Value	Description
`EXTERNAL-NODE-DEBUG-LEVEL` *Optional* *Scope:* Global	Integer *Range:* `[0, 3]` *Default:* `0`	Level of the debugging information to be displayed. `3` : Show each packet that is received or transmitted by the operational host, show the external interface statistics, and print debug messages. Print debugging statistics every three seconds (see Section 4.3.5.1.2). `2` : Show the external interface statistics and print debug messages. Print debugging statistics every three seconds (see Section 4.3.5.1.2). `1` : Print debug messages. `0` : Do not display any debugging information.

4.3.5.1.1.2 GUI Configuration

To configure the debugging parameters in the GUI, do the following:

1. Launch the **Advanced Emulation Configuration** editor by clicking the  button on the **Emulation** toolbar and go to the **Advanced** tab.

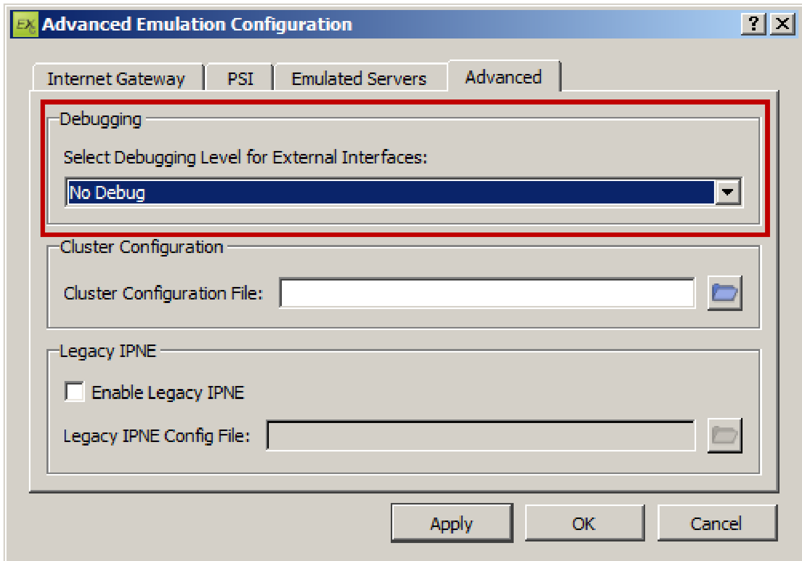


FIGURE 4-65. Configuring Debugging Parameters

2. From the pull-down menu in the **Debugging** section, select the level of debugging information to be displayed.

Note: Debugging options can be configured only in Design mode.

4.3.5.1.2 Debugging Information

If the parameter `EXTERNAL-NODE-DEBUG-LEVEL` is set to 2 or higher (see Section 4.3.5.1.1.1) or an equivalent debugging level is configured in the GUI (see Section 4.3.5.1.1.2), statistics, which can help in debugging, are printed every three seconds when the emulation is run. If the emulation is run from the command line, these statistics are printed to the command window from which the emulation is run. If the emulation is run from the GUI, these statistics are printed to the Output Window of Architect (see Section 6.6.2).

Figure 4-66 shows sample debugging information printed to the output window.

```
EXata Statistics at 56.994 sec
    Deviation from Real Time = +0.016 sec
    External Interface Capture Lag = 0.002 +- 0.015 ms
    Device Capture Statistics:: (this period/cumulative)
      Device:eth0     Received=2/0     Dropped=3/0
    All Traffic Statistics (this period/cumulative)::
        IN::   Packets=3/885  Pkts/sec=1.00/15.52  Bytes=294/1329588  Bytes/sec=98.00/23322.25
        OUT::  Packets=0/9  Pkts/sec=0.00/0.16  Bytes=0/144  Bytes/sec=0.00/2.53
    Application Traffic Statistics::
      Packets IN (this period/cumulative)
        TCP::   Packets=0/2  Pkts/sec=0.00/0.04  Bytes=0/148  Bytes/sec=0.00/2.60
        UDP::   Packets=0/879  Pkts/sec=0.00/15.42  Bytes=0/1329048  Bytes/sec=0.00/23312.78
        Total:: Packets=0/881  Pkts/sec=0.00/15.45  Bytes=0/1329196  Bytes/sec=0.00/23315.38
      Packets OUT (this period/cumulative)
        TCP::   Packets=0/0  Pkts/sec=0.00/0.00  Bytes=0/0  Bytes/sec=0.00/0.00
        UDP::   Packets=0/0  Pkts/sec=0.00/0.00  Bytes=0/0  Bytes/sec=0.00/0.00
        Total:: Packets=0/0  Pkts/sec=0.00/0.00  Bytes=0/0  Bytes/sec=0.00/0.00
    Multicast Traffic Statistics (this period/cumulative)::
        IN::   Packets=0/505  Pkts/sec=0.00/8.86  Bytes=0/763560  Bytes/sec=0.00/13393.58
        OUT::  Packets=0/6  Pkts/sec=0.00/0.11  Bytes=0/216  Bytes/sec=0.00/3.79
```

FIGURE 4-66. Sample Debugging Information

The debugging statistics can be divided into the following groups:

- Time Statistics
- Device Capture Statistics
- Traffic Statistics

Time Statistics

The time statistics are printed in the following format:

```
EXata statistics at <sim-time> sec::
    Deviation from Real Time = <deviation> sec
    External Interface Capture Lag = <avg-lag> +- <std-dev> msec
```

where

<sim-time> Simulation time when the statistics are printed.

<deviation> Difference between the real time elapsed since the beginning of the simulation and the simulation time when the statistics are printed.

This value should be close to zero. A higher value indicates that EXata is unable to keep up with the real time.

<avg-lag> The deviation in capturing packets from the network interface against the real time deadline, averaged over all packets received in the last three seconds.

This value should be close to zero. A higher value indicates sluggishness in the host Operating System to deliver packets to the EXata process.

<std-dev> Standard deviation of the packet capture lag.

Device Capture Statistics

The device capture statistics show the packets dropped due to buffer overflow (this can occur under high traffic load from operational hosts).

The packet drop statistics are printed for each device that is connected to an operational host in the following format:

```
Device Capture Statistics (this period/cumulative)::
   Device:<device-name>  Received= <r1>/<r2>  Dropped= <d1>/<d2>
```

where

`<device-name>`	Device name.
`<r1>`	Total number of packets received on this device in the last three seconds.
`<r2>`	Total number of packets received on this device (from the beginning of the simulation to the current time).
`<d1>`	Total number of packets dropped by this device in the last three seconds.
`<d2>`	Total number of packets dropped by this device (from the beginning of the simulation to the current time).

Note: Some versions of the libpcap library on Linux have a bug due to which the packet statistics are not correctly displayed. If the number of received packets incorrectly displayed as 0, consider updating the libpcap library.

Traffic Statistics

The traffic statistics show the statistics for all traffic and for application traffic in the following format:

```
Total Traffic Statistics (this period/cumulative)::
 IN::  <Total-in-stats>
 OUT:: <Total-out-stats>
Application Traffic Statistics::
 Packets IN (this period/cumulative)
    TCP::<TCP-in-stats>
    UDP::<UDP-in-stats>
    Total::<Total-app-in-stats>
 Packets OUT (this period/cumulative)
    TCP::<TCP-out-stats>
    UDP::<UDP-out-stats>
    Total::<Total-app-out-stats>
Multicast Traffic Statistics (this period/cumulative)::
 IN::  <Multicast-in-stats>
 OUT:: <Multicast-out-stats>
```

where

<Total-in-stats>	Statistics for all (application and control) incoming traffic (from the operational hosts to EXata). These are displayed in the following format: **Packets=<p1>/<p2> Pkts/sec=<pr1>/<pr2> Bytes=<b1>/b2> Bytes/sec=<br1>/br2>** where **<p1>** Number of packets received in the last three seconds **<p2>** Number of packets received from the beginning of the simulation to the current time **<pr1>** Rate of packets received in the last three seconds, in packets/sec **<pr2>** Rate of packets received from the beginning of the simulation to the current time, in packets/sec **<b1>** Number of bytes received in the last three seconds. **<b2>** Number of bytes received from the beginning of the simulation to the current time **<br1>** Rate of bytes received in the last three seconds, in packets/sec **<br2>** Rate of bytes received from the beginning of the simulation to the current time, in packets/sec
<Total-out-stats>	Statistics for all (application and control) outgoing traffic (from EXata to the operational hosts). These are displayed in the same format as the total incoming traffic statistics.
<TCP-in-stats>	TCP statistics for incoming traffic. These are displayed in the same format as the total incoming traffic statistics.
<UDP-in-stats>	UDP statistics for incoming traffic. These are displayed in the same format as the total incoming traffic statistics.
<Total-app-in-stats>	Statistics for all incoming application traffic (TCP and UDP). These are displayed in the same format as the total incoming traffic statistics.
<TCP-out-stats>	TCP statistics for outgoing traffic. These are displayed in the same format as the total incoming traffic statistics.
<UDP-out-stats>	UDP statistics for outgoing traffic. These are displayed in the same format as the total incoming traffic statistics.
<Total-app-out-stats>	Statistics for all outgoing application traffic (TCP and UDP). These are displayed in the same format as the total incoming traffic statistics.
<Multicast-in-stats>	Statistics for incoming multicast traffic. These are displayed in the same format as the total incoming traffic statistics.
<Multicast-out-stats>	Statistics for outgoing multicast traffic. These are displayed in the same format as the total incoming traffic statistics.

4.3.5.2 Cluster Computing Configuration

EXata fully supports Symmetric Multiprocessor (SMP) and cluster computing environments for external interfaces. Whereas the SMP environment is transparent to the user, for the cluster environment the user may configure the external interfaces to run on a single cluster node or all cluster nodes.

See Section 5.7 for details of distributed platform capabilities offered by EXata. This section describes how to configure multiple instances of external interfaces: one each on each cluster node.

4.3.5.2.1 Command Line Configuration

To configure the cluster parameters for the command line interface, include the parameters listed in Table 4-104 in the scenario configuration (.config) file.

TABLE 4-104. Cluster Parameters

Parameter	Value	Description
`CLUSTER-HOST-FILE` *Optional* *Scope:* Global	Filename	Name of the hostfile that is used by the MPI runtime environment. See *EXata Distributed Reference Guide* for details of the hostfile.

4.3.5.2.2 GUI Configuration

To configure the cluster parameters in the GUI, do the following:

1. Launch the **Advanced Emulation Configuration** editor by clicking the  button on the **Emulation** toolbar and go to the **Advanced** tab.

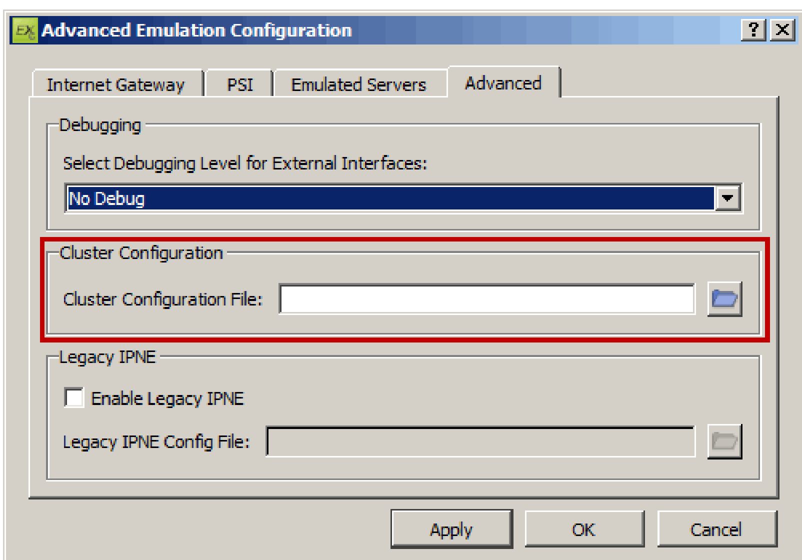


FIGURE 4-67. Configuring Cluster Configuration Parameters

2. In the **Cluster Configuration** section, click the  button and select the cluster configuration file from the file selector that is launched.

Note: Clusters can be configured only in Design mode.

4.3.5.3 Legacy IP Network Emulation (IPNE)

Legacy IP Network Emulation (IPNE) is a feature available as an addon to EXata. This feature is deprecated in EXata, but it is provided for backward compatibility.

4.3.5.3.1 Command Line Configuration

To configure the Legacy IPNE parameters for the command line interface, include the parameters listed in Table 4-105 in the scenario configuration (.config) file.

TABLE 4-105. Legacy IPNE Parameters

Parameter	Value	Description
`IPNE` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `NO`	Enables legacy IPNE. **Note:** The Connection Manager is disabled if legacy IPNE is enabled.
`IPNE-CONFIG-FILE` *Dependency:* `IPNE = YES` *Required* *Scope:* Global	Filename	Name of the IPNE configuration file. Refer to *Network Emulation Interface Model Library* for a description of the IPNE configuration file.

4.3.5.3.2 GUI Configuration

To configure the legacy IPNE parameters in the GUI, do the following:

1. Launch the **Advanced Emulation Configuration** editor by clicking the button on the **Emulation** toolbar and go to the **Advanced** tab.

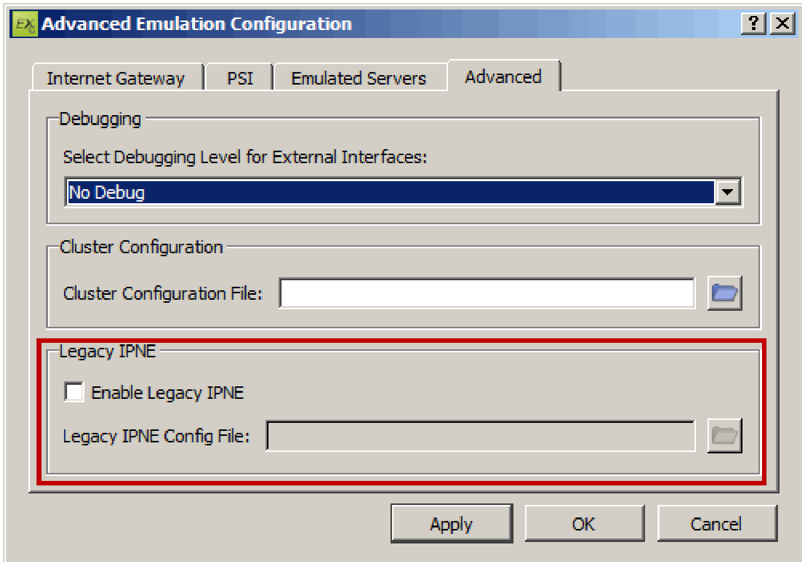


FIGURE 4-68. Configuring Legacy IPNE Parameters

2. In the **Legacy IPNE** section, check the box labeled **Enable Legacy IPNE**.
3. Click the button and select the IPNE configuration file from the file selector that is launched.

Note: Legacy IPNE can be configured only in Design mode.

4.3.6 Warm-up Phase

In some scenarios, it may take some time for routes in the network to converge. It is often desirable to inject external traffic into the network after the routes converge. To speed up the execution of the scenario, a warm-up phase can be configured. In the warm-phase, the scenario runs as fast as possible, i.e., in simulation mode. After the warm-up phase ends, the scenario runs in real-time, i.e., in emulation mode.

The scenario can be configured to treat external packets received during the warm-up phase in one of two ways:

- all external packets received during the warm-up phase are dropped without being processed, or
- each external packet received during the warm-up phase is delivered to its destination with zero delay.

Note: The warm-up phase operation is different if the Socket interface is used. Refer to *Federation Interfaces Library* for details.

4.3.6.1 Command Line Configuration

To configure the warm-up phase for the command line interface, include the parameters listed in Table 4-106 in the scenario configuration (.config) file.

TABLE 4-106. Warm-up Phase Parameters

Parameter	Value	Description
`EXTERNAL-WARM-UP-TIME` *Optional* *Scope:* Global	Time *Range:* ≥ `0S` *Default:* `0S`	Length of the warm-up phase.
`EXTERNAL-WARM-UP-DROP` *Optional* *Scope:* Global	List: • `YES` • `NO` *Default:* `NO`	Indicates whether packets received from external sources in the warm-up phase are dropped or delivered to the destination with zero delay. By default external packets are delivered to the destination with zero delay.

4.3.6.2 GUI Configuration

To configure the warm-up phase in the GUI, perform the following steps:

1. Go to **Scenario Properties Editor > External Interfaces > Warm-up Phase**.
2. To enable the warm-up phase, set **Enable Warm-up Phase** to *Yes* and set the dependent parameters listed in Table 4-107.

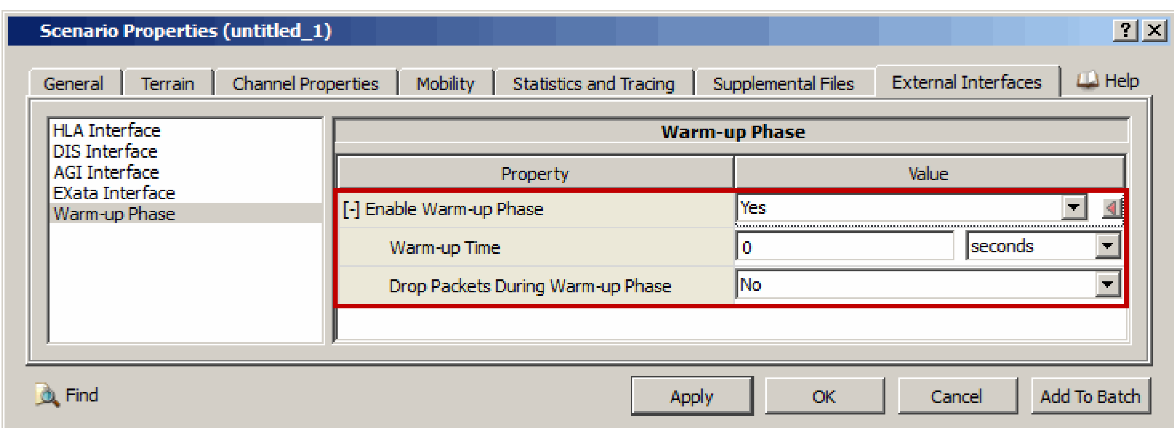


FIGURE 4-69. Setting Warm-up Phase Parameters

TABLE 4-107. Command Line Equivalent of Warm-up Phase Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Warm-up Time	Global	`EXTERNAL-WARM-UP-TIME`
Drop Packets During Warm-up Phase	Global	`EXTERNAL-WARM-UP-DROP`

4.4 Configuring the Multi-core/Multi-processor Environment

EXata is designed to take advantage of multiple processing units (cores or processors) to achieve speed-up. When EXata is run on a multi-core or multi-processor system, simulation tasks can be distributed among the processing units. This can potentially result in faster simulations.

There are two types of multi-processing unit systems: shared memory systems and distributed systems. When EXata is run on a shared memory system, compilation and setting the environment do not require any additional steps. To run EXata on a shared memory system from the command line, the number of processing units is specified by using the **`-np`** option (see Section 2.1.2).

When running EXata on a distributed system, the process for compilation and setting the environment is different. See *EXata Distributed Reference Guide* for instructions for compiling, setting the environment, and running EXata on distributed systems.

For running EXata on multiple processing units, all nodes in the scenario have to be divided into partitions. Each partition is assigned to one processing unit. The partitioning of nodes can be done automatically by EXata, or the user may opt to do the partitioning manually. This step is applies to both shared memory and distributed systems.

4.4.1 Command Line Configuration

Table 4-108 describes the partitioning parameters.

TABLE 4-108. Partitioning Parameters

Parameter	Value	Description
`PARTITION-SCHEME` *Optional* *Scope:* Global	List: • `AUTO` • `MANUAL` *Default:* `AUTO`	Scheme for distribution (partitioning) of nodes among processors. `AUTO` : Nodes are assigned to partitions automatically. Nodes are distributed in a card dealing fashion in the order in which the node identifiers appear in the scenario configuration (.config) file. `MANUAL` : Each node is assigned to a partition by the user by using the `PARTITION` parameter.
`PARTITION` *Optional* *Scope:* Node	Integer *Range:* `[0, N-1]` (see description)	Partition number to which the node belongs. The partition number should be between 0 and N-1, where N is the number of processors specified on the command line (see Section 2.1.2). **Note:** This parameter must be specified if `PARTITION-SCHEME` is set to `MANUAL`

Example:

Including the following parameters in the scenario configuration (.config) file places nodes 1 through 10 in partition 0 and nodes 11 through 20 in partition 1:

```
PARTITION-SCHEME MANUAL
[1 thru 10] PARTITION 0
[11 thru 20] PARTITION 1
```

To run a scenario on a shared-memory system with two processing units, use the following command.

```
exata scenario.config -np 2
```

Limitations

If EXata is run on multiple processing units and automatic node partitioning is selected, then all nodes belonging to a 802.3 subnet, to a switched ethernet subnet, or to a subnet with MAC protocol as Wormhole MAC are placed in the same partition. If manual node partitioning is selected, then the user must place all nodes belonging to a 802.3 subnet or to a switched ethernet subnet in the same partition.

4.4.2 GUI Configuration

To configure a scenario for a multi-processor environment in the GUI, the partitioning scheme is configured at the global level and partition numbers are assigned to nodes at the node level.

Configuring Partitioning Scheme

To configure the partitioning scheme in the GUI, do the following:

1. Go to **Scenario Properties Editor > General > Parallel Settings**.
2. Set **Parallel Partition Scheme** to the desired value.

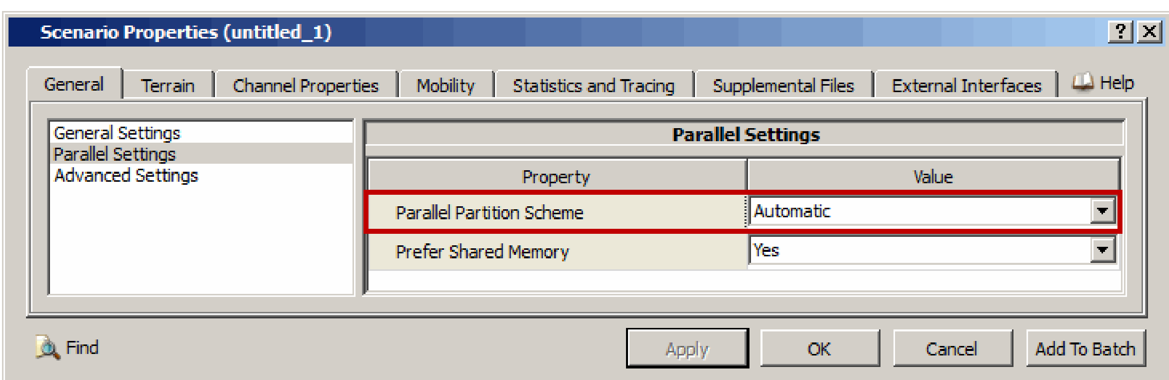


FIGURE 4-70. Specifying Partitioning Scheme

TABLE 4-109. Command Line Equivalent of Partition Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Parallel Partition Scheme	Global	`PARTITION-SCHEME`

Configuring Partition Numbers

To configure the partition number for a node in the GUI, do the following:

1. Go to **Default Device Properties Editor > General**.
2. Set **Partition** to the desired value.

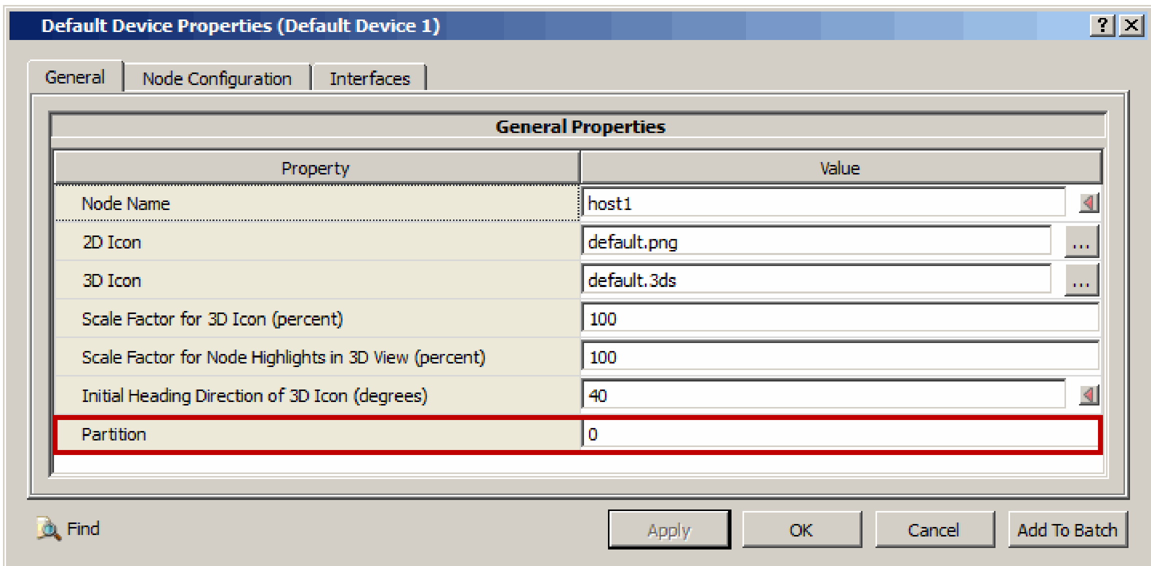


FIGURE 4-71. Specifying the Partitioning Number

TABLE 4-110. Command Line Equivalent of Partitioning Scheme Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Partition	Node	`PARTITION`

4.5 Performance Optimization Features

This section describes features for improving the execution speed of scenarios. Section 4.5.1 describes parameters that can be used to optimize the simulation's run-time performance by reducing the amount of wireless signal propagation computation. Section 4.5.2 describes how to configure the simulator’s event queue to optimize performance.

4.5.1 Execution Speed and Accuracy Trade-off

In simulation of wireless networks, calculation of pathloss between transmitters and receivers presents a significant computational overhead. Detailed pathloss calculations yield more accurate simulation results, but at the expense of simulation speed. Simulation performance can be improved if pathloss calculations are abstracted, but this typically results in the loss of accuracy. This section describes parameters that can be configured to achieve the desired trade-off between speed and accuracy.

A common technique to improve the performance of a wireless network simulation is to limit the number of signal strength computations using some common assumptions. One such assumption is that nodes that are far away from each other cannot communicate directly and their signals do not interfere with each other. However, directionally transmitted signals with high EIRP (Effective Isotropic Radiated Power) do propagate over a long distance and may cause high interference at far sight. Further, even if a single signal does not cause noticeable interference, power from many weak signals can be accumulated enough to cause high interference. Therefore, failing to consider such signals could introduce high inaccuracy in the network performance prediction.

This section describes parameters that can be used to optimize the simulation's run-time performance by reducing the amount of wireless signal propagation computation. Appropriate values for these parameters depend on the scenario and the desired trade-off between speed and accuracy.

Section 4.5.1.1describes how to configure the speed and accuracy trade-off parameters and Section 4.5.1.2 describes the parameters in detail.

4.5.1.1 Configuring Speed and Accuracy Trade-off Parameters

This section describes how to configure the speed and accuracy trade-off parameters for the command line interface and in the GUI.

4.5.1.1.1 Command Line Configuration

To configure the speed and accuracy trade-off parameters for the command line interface, include the parameters listed in Table 4-111 in the scenario configuration (.config) file.

TABLE 4-111. Simulation Speed and Accuracy Trade-off Parameters

Parameter	Value	Description
`PROPAGATION-MAX-DISTANCE` *Optional* *Scope:* Global *Instances:* channel index	Real *Range:* `> 0.0` *Default*: `0.0` (see note) *Unit:* meters	Maximum distance for which a node's transmission is considered for communication or interference (see Section 4.5.1.2.1). **Note:** If this parameter is set to `0.0`, then it is not considered in the estimation of a node's propagation range, i.e., the maximum distance is effectively infinity. In this case, the propagation range is determined only by the parameter `PROPAGATION-LIMIT`.
`PROPAGATION-LIMIT` *Optional* *Scope:* Global *Instances:* channel index	Real *Default:* `-110.0` *Unit:* dBm	Threshold for delivering signals to nodes. Signals received with power below this limit (before accounting for antenna gain at the receiver) are not delivered to nodes. This parameter is meant for optimizing simulation performance. A lower value of the parameter results in a more accurate simulation but at the expense of longer execution time (see Section 4.5.1.2.2).

TABLE 4-111. Simulation Speed and Accuracy Trade-off Parameters (Continued)

Parameter	Value	Description
MOBILITY-POSITION-GRANULARITY *Optional* *Scope:* Global, Node	Real *Range:* > 0.0 *Default:* 1.0 *Unit*: meters	Distance by which a node moves in a single step. The mobility position granularity also impacts the frequency of pathloss updates, and hence the accuracy and execution speed. See Section 4.5.1.2.3.
PROPAGATION-COMMUNICATION-PROXIMITY *Optional* *Scope:* Global *Instances:* channel index	Real *Range:* > 0.0 *Default*: 400.0 *Unit:* meters	Communication proximity used to calculate the frequency of pathloss updates (see Section 4.5.1.2.4). This parameter should be set to the approximate optimistic radio range.
PROPAGATION-PROFILE-UPDATE-RATIO Optional *Scope:* Global *Instances:* channel index	Real *Range:* [0.0, 1.0] *Default:* 0.0	Update ratio used to calculate the frequency of pathloss updates (see Section 4.5.1.2.4). A larger value of this parameter results in a more aggressive optimization.

Note: Parameters MOBILITY-POSITION-GRANULARITY, PROPAGATION-COMMUNICATION-PROXIMITY, and PROPAGATION-PROFILE-UPDATE-RATIO do not work properly when certain external programs (such as HLA) control nodes' mobility.

4.5.1.1.2 GUI Configuration

See Section 4.2.6.2 and Section 4.2.7.2 for configuring speed and accuracy trade-off parameters for the GUI.

4.5.1.2 Relationship between Speed and Accuracy Trade-off Parameters

This section describes the relationship between the execution speed and accuracy trade-off parameters.

Figure 4-72 gives a pictorial view of how the parameters should be set. PROPAGATION-COMMUNICATION-PROXIMITY should be somewhat larger than (perhaps 1.5 times) the effective radio range, and PROPAGATION-MAX-DISTANCE should be as close as possible to the distance where the PROPAGATION-LIMIT takes effect.

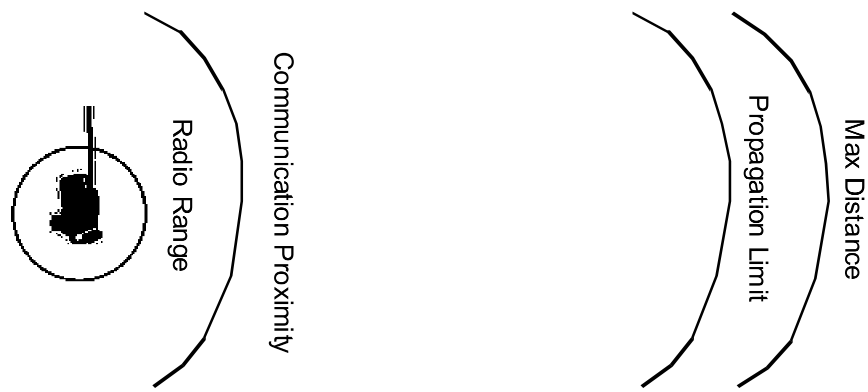


FIGURE 4-72. Relationship between Speed and Accuracy Trade-off Parameters

4.5.1.2.1 Maximum Propagation Distance Parameter

The `PROPAGATION-MAX-DISTANCE` parameter is used to strictly prohibit communications between nodes that are far apart. Transmissions to nodes that are more than this distance away are ignored in signal strength calculations. By default, this optimization is disabled (the maximum propagation distance is assumed to be infinity), i.e., signal strength is calculated for all nodes listening to the transmit channel.

Because out-of-range signals are still used to calculate interference, this value should be set to several times larger than the normal communication range of the radios in use: at least 5 times larger for ground based radios and 10 times for airborne radios. An ideal value would be the distance just outside the range projected by the `PROPAGATION-LIMIT` parameter.

4.5.1.2.2 Propagation Limit Parameter

The `PROPAGATION-LIMIT` parameter sets the propagation limit: signals calculated to have less receive power (before accounting for antenna gain at the receiver) than this limit will not be delivered.

Most other discrete-event simulation tools limit the propagation of signals substantially in order to improve the runtime performance without examining the validity of the propagation limit. By setting an overly short propagation limit, the simulator can not only save the computation for propagation path profiling, but also the number of events to be scheduled for interference. However, the prediction of network performance becomes highly inaccurate as the amount of interference is significantly underestimated.

An appropriate value for this variable depends on the capabilities of the radios being simulated. The limit should be lower for highly sensitive radios that can receive low power signals. A good value can be determined empirically by running a scenario repeatedly while raising the limit and keeping the highest value that doesn't significantly impact the network performance.

4.5.1.2.3 Mobility Granularity Parameter

The `MOBILITY-POSITION-GRANULARITY` parameter specifies the distance by which a node moves in each step, i.e., the minimum distance by which the node moves when its position is updated. Since signal strength between two nodes only needs to be recalculated when one of the nodes' position is updated, mobility distance granularity affects the number of position updates as well as the number of signal strength calculations. Finer position granularity leads to more position updates, more pathloss updates, and generally greater accuracy. Courser granularity leads to fewer position and pathloss updates and less accuracy, but faster execution.

4.5.1.2.4 Communication Proximity and Update Ratio Parameters

The parameters `PROPAGATION-COMMUNICATION-PROXIMITY` and `PROPAGATION-PROFILE-UPDATE-RATIO` work together to limit the number of signal computations for nodes that are out of the normal communication range of the radios being used, but still within the propagation limit and the maximum communication distance, as described earlier:

When nodes are within, or just outside the nominal communication range of their radio systems, the strength of signals between the two nodes should be calculated with the highest possible fidelity. But signals between nodes which are farther apart can be treated more abstractly. When this optimization is activated, i.e., when these two parameters are specified for a channel, the frequency of signal strength recomputations decreases as the distance between the transmitting and receiving nodes increases.

The signal strength is recomputed only when the distance between the nodes changes by the larger of `MOBILITY-POSITION-GRANULARITY` and *min-distance*, where *min-distance* is calculated by the following formula:

> *min-distance* = (*Current-Distance* - `PROPAGATION-COMMUNICATION-PROXIMITY`) * `PROPAGATION-PROFILE-UPDATE-RATIO`

where

Current-Distance Current distance between the nodes.

The nodes' positions continue to be updated according to the given mobility position granularity, but not all such moves cause a new computation of pathloss. Similar to propagation limit and mobility distance granularity optimizations, this optimization may affect the accuracy of the simulation and, therefore, the values must be carefully chosen.

4.5.2 Simulator Event Queue

The type of data structure used by the simulator to maintain its event queue can have a significant impact on its performance. This section describes how to configure the data structure type for the event queue.

Notes:
1. The choice of event queue data structure does not affect the simulation results but may impact the execution time of the scenario.
2. It is recommended that only users who are familiar with the details of discrete-event simulation modify the data structure type.

4.5.2.1 Command Line Configuration

To configure the event queue data structure type for the command line interface, include the parameters listed in Table 4-112 in the scenario configuration (.config) file.

TABLE 4-112. Event Queue Type Parameters

Parameter	Value	Description
`SCHEDULER-QUEUE-TYPE` *Optional* *Scope:* Global	List: • `CALENDAR` • `SPLAYTREE` • `STDLIB` *Default:* `CALENDAR`	Specifies the type of data structure used by the simulation kernel for the event queue. `CALENDAR` : Uses a priority queue that organizes events into buckets. `SPLAYTREE` : Uses a separate priority queue for each node. `STDLIB` : Uses a single global Standard Template Library (STL) queue.

4.5.2.2 GUI Configuration

To configure the event queue data structure type in the GUI, do the following:

1. Go to **Scenario Properties Editor > General > Advanced Settings**.
2. Set **Event Queue Type** to the desired value.

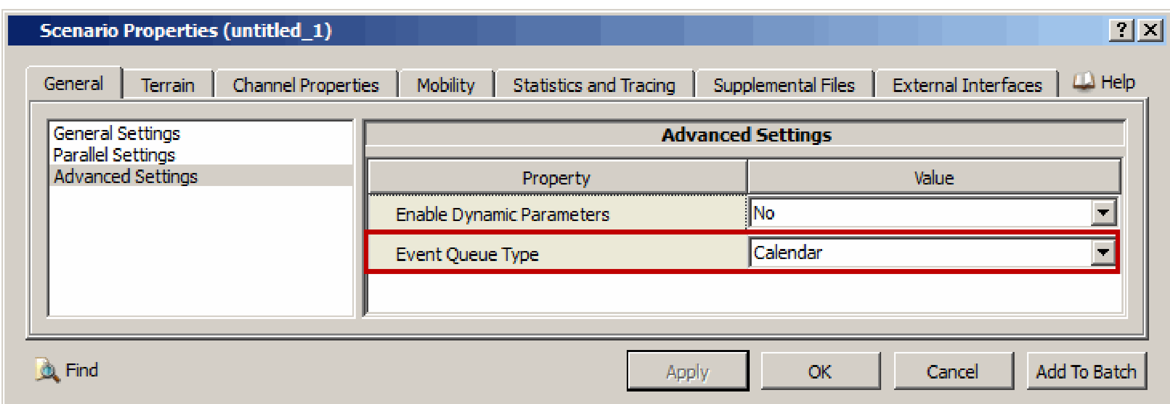


FIGURE 4-73. Configuring Event Queue Type

TABLE 4-113. Command Line Equivalent of Event Queue Type Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Event Queue Type	Global	`SCHEDULER-QUEUE-TYPE`

4.6 Advanced Features in Network Modeling

This section gives an overview of some advanced features used in network simulation. The details of the features are provided in the model libraries.

4.6.1 Cyber Models

EXata provides a simulation and emulation tool that can assess net-centric cyber models and evaluate their performance in experimental scenarios. EXata implements the following cyber models:

- IPSec: At the Network Layer, EXata provides support for IP Security (IPSec), which provides a cryptographically based security for IPv4 and IPv6.
- WEP/CCMP: At the link layer, EXata provides support for Wired Equivalent Privacy (WEP) and its secured successor, Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). WEP is a MAC Layer security protocol intended to provide security for the wireless LAN equivalent to the security provided in a wired LAN. CCMP is an IEEE 802.11i encryption protocol designed to replace WEP.
- Certificate Model: The Certificate model simulates the generation of certificates for unique network addresses.
- Firewall Model: The firewall model in EXata is a *packet-based stateless software* firewall. That is, the firewall model is a software process that inspects each packet to determine if the packet should be allowed or denied access.
- Information Assurance Hierarchical Encryption Protocol (IAHEP): IAHEP is an encryption protocol that allows two or more secure enclaves to exchange data over an untrusted network.
- Internet Security Association and Key Management Protocol with Internet Key Exchange (ISAKMP-IKE): ISAKMP-IKE combines the security concepts of authentication, key management, and security associations to establish secure communications on the Internet. Internet Key Exchange (IKE) is a hybrid protocol to obtain authenticated keying material for use with ISAKMP and for other security associations.
- Public Key Infrastructure (PKI): A PKI is an infrastructure that uses digital certificates as an authentication mechanism and is built to better manage certificates and their associated keys. A digital certificate is itself a way to reliably identify the user or computer claiming to be the owner of a specific public key.
- Secure Neighbor Model: The Secure Neighbor model simulates the authentication by a node of each of its one-hop neighbors' identity and location, in a mobile environment.
- Adversary Model: The Adversary model can simulate two types of attacks: Active (or Wormhole) attack, in which an adversary carries information traveling faster than the speed of light, and Passive (or Eavesdrop) attack, in which wireless traffic is intercepted by an eavesdrop entity.
- Anonymous On-Demand Routing (ANODR): ANODR provides a net-centric anonymous and untraceable routing scheme for mobile ad-hoc networks. The protocol provides mobile anonymity and data confidentiality.
- Denial of Service (DOS) Attack Model: A Denial-of-Service (DOS) attack is the act of overwhelming the resources of a victim computer or network so that the victim cannot service requests from other clients. The clients, therefore, are denied service from the victim computer or network.
- Signal Intelligence (SIGINT) Attack Model: Signals Intelligence is an act of gathering information by intercepting and analyzing the signals. No attempt is made to decode the signal. Only the characteristics of signals, such as frequency range, power of transmission, RF signatures, etc., are determined.
- Virus Attack Model: In EXata, a virus attack is modeled as the attacker node sending packets with payloads that contain signatures of some well-known attacks. Note that these packets *do not* contain

any actual virus payload, only their signatures. It is expected that any Intrusion Detection Systems (IDS) or Anti-Virus Software can detect the signature of these packets and classify them as malicious.

- Wireless Eavesdropping Model: Eavesdropping is a *passive* attack where an intruder node attempts to capture private information from a network. In wireless eavesdropping, the intruder node configures its radio to be on the same channel as the victim network and promiscuously listens for broadcast transmissions that are destined for member nodes of the network.
- Wireless Jamming Model: Jamming is transmission of radio signals at sufficiently high energy to cause disruption of communication for nearby radios. The signals transmitted by jammers interfere with other legitimate signals in the vicinity of the jammer, causing the signal to noise ratio of the latter signals to drop significantly and resulting in corruption of those signals.
- CPU and Memory Resource Model: The CPU and Memory Resource Model monitors the allocation, consumption, and depletion of resources for a node. This model is used in conjunction with the DOS attack model. The DOS attack model attempts to consume the resources at the victim node, causing the victim node to fail when the resources are completely depleted.

Refer to *Cyber Model Library* for details of the cyber models.

4.6.2 Simple Network Management Protocol (SNMP)

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.

Refer to *Network Management Model Library* for details of the SNMP model.

4.6.3 Battery Models

Battery models capture the characteristics of real-life batteries and can be used to predict their behavior under different design choices, such as system architecture, power management policy, etc.

Battery models capture the characteristics of real-life batteries, and can be used to predict their behavior under different design choices, such as system architecture, power management policy, etc. Battery models are useful tools for a battery-driven system design approach, because they enable analysis of the discharge behavior of the battery under different design choices (for example, system architectures, power management policies, and transmission power control), without resorting to time consuming (and expensive) prototyping and measurement for each alternative.

EXata provides several battery models which are described in *Wireless Model Library*.

4.6.4 Custom Antenna Models

EXata supports three standard antenna models: omni-directional, switched-beam, and steerable. The antenna gain for the omni-directional antenna is the same in all directions. For switched-beam and steerable antennas, the antenna gains in different directions are read from the azimuth pattern file and (optionally) the elevation pattern file. These pattern files are specified using the *traditional* format.

EXata also supports the patterned antenna model, which uses pattern files that can be specified in Open-ASCII (2-D and 3-D) and NSMA formats, in addition to the traditional format. The patterned antenna model is highly customizable. Another input file, the antenna model file, is used for the patterned antenna model. The name of the antenna model file is specified in the scenario configuration (.config) file by using the parameter `ANTENNA-MODEL-CONFIG-FILE` (see Section 4.2.8.1.3). The antenna model file contains definitions of one or more antenna models. Each antenna model definition comprises a model name followed by the parameters for that model. The configuration (.config) file refers to a model in the antenna model file by the model's name.

Note: The antenna model file can also be used for defining customized omni-directional, switched-beam, and steerable antenna models as well. In this case, a set of values for the antenna parameters (gain, height, efficiency, different losses, and azimuth and elevation pattern files) are associated together and given an antenna model name. This name can be used in the scenario configuration (.config) file to assign an antenna model to a node.

Custom antenna models are of four types: omni-directional, switched-beam, steerable, and patterned. The antenna model file supports many parameters that allow detailed descriptions of patterned antenna models. Refer to *Wireless Model Library* for a description of custom antenna models.

4.6.5 Weather Effects

Weather effects can affect signal propagation. Weather patterns that move can be modeled in EXata. A weather configuration file is used to specify the shape, movement, altitude, and intensity of one or more weather patterns. This information is used in pathloss calculations.

Refer to *Wireless Model Library* for details of configuring weather effects.

4.6.6 Switches

Switches are MAC Layer devices that connect multiple LAN segments and perform the following functions:

- Provide physical isolation of segments
- Eliminate collision between segments
- Provide buffering of frames

The switched model is based on the IEEE 802.1 specification and implements the following features:

- Conformance with the MAC protocol models implemented at the switch ports. Currently, two models are supported – Link and MAC 802.3.
- Relay of frames between ports including conversion of frame format, if required.
- Basic filtering service to enable relay between ports of the switch.
- Multiple priority queue-based buffering at each port. However, the supported MAC protocols do not carry priority information in frames.
- Spanning tree algorithm to determine a loop-free path between connected LANs.

Refer to *Multimedia and Enterprise Model Library* for details of the switch model.

4.6.7 Interface Faults

Faults can occur at specific interfaces of a node. Faults are modeled by means of a fault file which specifies the up and down times of interfaces affected by faults. Faults can be static or dynamic. A static fault causes an interface to be unavailable at a pre-determined time for a pre-determined length of time. A dynamic fault can occur at a random time for a random length of time and may occur repeatedly.

Refer to *Developer Model Library* for details of specifying faults.

4.6.8 Asynchronous Transfer Mode (ATM)

Asynchronous Transfer Mode (ATM) is a connection-oriented cell relay protocol. Information bit streams are conveyed in small fixed-size cells (53 bytes). The ATM model in EXata implements the ATM layer 2, ATM signaling layer, and ATM Adaptation Layer Type 5 (AAL5). It also supports interoperability between IP networks and ATM networks, known as IP over ATM.

The following features are implemented in the EXata ATM model:

- Cell construction
- Cell reception and header validation
- Cell relaying, forwarding and copying
- Cell multiplexing and demultiplexing.
- Interpretation-only data cell and signaling cell
- Explicit forward congestion indication
- Point-to-point connection
- Connection assignment and removal

Refer to *Developer Model Library* for details of the EXata ATM model.

4.6.9 Multi-Protocol Label Switching (MPLS)

Multi-Protocol Label Switching (MPLS) is a data-carrying mechanism that belongs to the family of packet-switched networks. MPLS operates between layer 2 and layer 3 of the OSI model. It is designed to provide a unified data-carrying service for both packet-switched and circuit-switched clients which provide a datagram service. MPLS can be used for different kinds of traffic, including IP packets and ATM, SONET, and Ethernet frames.

MPLS support in EXata is composed of a label-swapping framework between the Network Layer (IP) and MAC Layer, which interacts with a label distribution protocol, generally at the Application or Transport Layers.

Label distribution protocols handle the assignment of labels to Forward Equivalence Classes and the distribution of these labels to routers within the MPLS cloud that need to be aware of them. MPLS allows for the existence of multiple label distribution protocols. EXata supports the following label distribution protocols:

- Label Distribution Protocol (LDP): LDP is the name of a specific label distribution protocol described in RFC 3036. It allows label switched routers to exchange messages via UDP and TCP (session-related, label advertisement, and notification).
- Resource Reservation Protocol - Traffic Engineering (RSVP-TE): RSVP-TE, described in RFC 3209, is a traffic-engineering extension to RSVP which allows the protocol to create label-switched paths using RSVP as a signaling protocol.

Refer to *Multimedia and Enterprise Model Library* for details of the EXata MPLS model.

4.6.10 Router Models

Routers are modeled as special devices. A router model characterizes the hardware and software capabilities of the router, including the backplane throughput, queue type, and scheduler type. EXata provides pre-configured models for many popular routers used in enterprise networks. Users can also configure their own router models.

Refer to *Multimedia and Enterprise Model Library* for details of configuring router models.

4.6.11 Router Configuration

In EXata, any node can be configured to be a router. There are some additional parameters for routers which are specified in a separate file, the router configuration file. The name of the router configuration file is specified in the scenario configuration (.config) file by using the parameter `ROUTER-CONFIG-FILE` (see Section 4.2.8.3). Typically, the protocols running at the router are specified in this manner. See Section 4.6.10 for specifying the system properties of the router.

The router-related parameters fall under the following categories:

- Policy-based Routing Parameters: Policy-based routing provides a mechanism to mark packets so that certain kinds of traffic receive differentiated, preferential treatment. Policy-based routing allows for forwarding of data packets based on policies defined by the network administrator.
- Router Access List Parameters: The router access list performs access control by accepting or denying packets based on the traffic type or network address.
- Route Redistribution Parameters: Route redistribution is used by a gateway router which connects two or more routing domains to advertise routes learnt from one domain to the other domains.
- Route Map: A route map is used to control redistribution, to control and modify routing information, and to define policies in policy-based routing. A route map defines criteria that packets should meet and the action to ba taken when the criteria are met.
- Hot Standby Router Protocol (HSRP) Parameters: HSRP allows a host to specify a virtual next hop router to forward packets. Routers participating in the same standby group dynamically determine the active and standby routers. Only the active router forwards packets.

Parameters for policy-based routing, routing access list, route redistribution, and route map are specified in the router configuration file, while the HSRP parameters are specified in the scenario configuration (.config) file.

Refer to the Policy-based Routing, Router Access Lists, Route Redistribution, Route Map, and HSRP sections of *Multimedia and Enterprise Model Library* for details of configuring routers.

4.6.12 Quality of Service (QoS) Modeling

To support quality of service (QoS) in networks, various QoS mechanisms can be employed. These include Integrated Service (InterServ)/Resource Reservation Protocol (RSVP), the differentiated services (DiffServ) framework, Multi-Protocol Label Switching (MPLS), traffic engineering (e.g., scheduling), and QoS routing. QoS routing protocols provide bandwidth and delay guarantees to QoS flows. In wireless networks, some MAC protocols can provide QoS by giving a higher precedence to high priority packets.

In EXata, Qos mechanisms at three layers are modeled:

- Application Layer: Some traffic generator models support QoS parameters. These are explained below.
- Network Layer: The DiffServ framework can be employed to provide QoS. Refer to *Multimedia and Enterprise Model Library* for details of the DiffServ model.
- MAC Layer: EXata models two QoS-capable MAC protocols: IEEE 802.11e MAC and IEEE 802.16 MAC. Refer to *Wireless Model Library* for a description of the IEEE 802.11e MAC model and *Advanced Wireless Model Library* for a description of the IEEE 802.16 MAC model.

QoS Parameters in Traffic Generator Models

Many traffic generators in EXata support the option of specifying the QoS a data packet should have when being delivered to the destination. The QoS policy can be specified by giving the data packets a Precedence, Differentiated Service Code Point (DSCP), or Type of Service (ToS) value. The Precedence, DSCP, or ToS value is specified as an optional parameter, along with the other parameters of the traffic generator. Only one of the three QoS parameters can be specified for a traffic generator.

Each of the three QoS parameters is converted to an 8-bit value which is assigned to the ToS field of the IP header.

- **Precedence**: The value range for Precedence is 0 to 7. The Precedence value maps to the most significant three bits of the ToS field of the IP header. The remaining bits of the ToS field are 0. See RFC 791 for more details.
- **DSCP**: The value range for DSCP is 0 to 63. The DSCP value maps to the most significant six bits of the ToS field of the IP header. See RFC 2474 for more details.
- **ToS**: The value range for ToS is 0 to 255. The ToS value maps to the entire ToS field of the IP header.

The following traffic generator models support QoS parameters:

- CBR
- FTP/Generic
- Lookup
- Traffic-Gen
- Super-Application
- VoIP

4.6.13 Voice over IP (VoIP)

Voice over IP (VoIP) is an application for routing voice conversations over the Internet or through any other IP-based network. The protocols used to carry voice traffic over the IP network are referred to as VoIP protocols.

The EXata VoIP suite of protocols includes the H.323, SIP, RTCP, and RTP protocols, and the VoIP Traffic Generator and VoIP Jitter Buffer models.

The VoIP traffic Generator simulates real-life telephone conversations. The initiator of the conversation generates real-time traffic with an exponential distribution function.

In the context of VoIP, jitter is defined as the difference between the expected time of arrival of a packet and the actual time of arrival. Jitter is caused primarily by delays and congestion in the packet network. JItter causes discontinuity in the real-time voice stream. To minimize the delay variations, a jitter buffer is implemented which temporarily stores arriving packets. The EXata VoIP model enables the user to configure the jitter buffer in detail.

EXata also has detailed implementations of H.323 and Session Initiation protocol (SIP). H.323 is commonly used in VoIP, Internet telephony, and IP-based video-conferencing. The H.323 standard is based on the Internet Engineering Task Force (IETF) Real-Time Protocol (RTP) and Real-Time Control Protocol (RTCP), with additional protocols for call signaling and data and audiovisual communication.

SIP is an alternative to H.323. It is an Application Layer signaling protocol (end-to-end) used to set up, modify, and terminate multimedia sessions over the Internet. SIP modeled on Internet protocols such as HTTP and SMTP.

Refer to *Multimedia and Enterprise Model Library* for a detailed description of the VoIP suite of models.

4.6.14 Satellite Models

EXata's satellite models provide a number of features to assist the system architect analyze the performance of satellite-enabled networks. These include:

- Different beam signatures from a cellular-style reuse (spot-beam) to global multicast (potato or CONUS beam).
- Advanced MAC simulation of multiple uplinks (return channels) per downlink (forward channel).
- Advanced modulation techniques and forward error correction simulation.
- Adaptive modulation techniques on the forward and reverse directions of the satellite to adapt to dynamic transmission conditions.
- Quality of service and traffic conditioning to ensure fairness to all users.
- Support for advanced media access and resource allocation algorithms for packet scheduling.
- Different equipment configurations such as payload processing and repeater (bent-pipe) systems.

Refer to *Wireless Model Library* for details of the satellite models.

4.6.15 External Interfaces

This section gives an overview of the external interfaces supported in EXata.

4.6.15.1 VR-Link Interface

The VR-Link networking toolkit from MAK Technologies (www.mak.com) provides a protocol-independent API that abstracts away the details of simulation networking protocols. The EXata VR-Link interface allows you to network simulators and virtual reality applications using High Level Architecture (HLA) or Distributed Interactive Simulation (DIS) protocol.

4.6.15.1.1 High-Level Architecture (HLA) Interface

HLA is a general-purpose architecture for distributed computer simulation systems. Computer simulations can interact (i.e., communicate data and synchronize actions) with other computer simulations regardless of the computing platforms. The interaction between simulations is managed by a Run-Time Infrastructure (RTI). HLA is an interoperability standard for distributed simulation used to support analysis, engineering and training in a number of different domains.

EXata implements an HLA interface, using which EXata can interact with other software programs which also have an HLA interface. The EXata HLA interface has been used to successfully interface EXata with CAE STRIVE using HLA directly, and with OneSAF Testbed Baseline (OTB) using separate Distributed Interactive Simulation (DIS)/HLA gateway software.

The EXata HLA interface is described in *Federation Interfaces Library*.

4.6.15.1.2 Distributed Interactive Simulation (DIS) Interface

DIS is an IEEE standard for interfacing multiple simulation tools into a single, real-time exercise. The DIS standard defines a legacy network protocol used for enabling the inter-operation of two or more software programs (usually simulation software). The transport of information between simulators is performed using UDP and multicast IP. Although formally superseded by HLA, DIS still remains popular today for its simplicity of operation and the ease in which a DIS interface can be created.

The EXata DIS interface is described in *Federation Interfaces Library.*

4.6.15.2 Socket Interface

The Socket Interface provides inter-process communication between EXata and an external program over a TCP socket, with EXata acting as the server and the external program as the client. Several types of messages can be sent between the two processes.

The EXata Socket interface is described in *Federation Interfaces Library*.

4.6.15.3 Satellite ToolKit (STK) Interface

The EXata STK interface provides a way to interface EXata with the Satellite ToolKit (STK) developed by Analytical Graphics, Inc. (AGI) and function in a client-server environment. This is achieved by using AGI's STK/Connect library. The STK/Connect library contains messaging capabilities that allow third-party applications like EXata to directly interact with STK.

The EXata STK interface communicates with STK to visualize events in real-time. For instance, you can feed real-time telemetry data from the launch and early orbit of a mission. As a scenario, the data can be viewed in 2D or 3D to visualize the mission and assist in understanding and resolving any issues that may arise. The EXata/STK Integration module provides a framework to interact with STK through STK/Connect. To create custom enhancements, you can easily add support for other STK/Connect features.

Refer to *Developer Model Library* for details of the EXata STK interface.

5 Using Emulation

EXata allows a large category of applications to be executed within the EXata emulation network, including end-to-end networking applications, Internet-based applications, and SNMP-based applications. Additionally, EXata allows third party packet sniffing software to capture and analyze the emulated traffic.

This chapter describes the steps involved in preparing an emulation testbed and configuring applications to run on it. It is assumed that a network scenario is available to be executed by EXata, either in the EXata GUI or via the command line interface. See Chapter 4 for details of configuring network scenarios in EXata or use one of the pre-configured scenarios available in the folder EXATA_HOME/scenarios.

Note: To use the features described in this chapter, EXata must be run from an administrator or root user account.

Section 5.1 describes how to configure the EXata testbed consisting of Operational Hosts (the physical hardware running real applications) and the Emulation Server (the machine running EXata).

Section 5.2 describes how to run real applications on an emulated network. This section also describes EXata's emulated server applications and the configuration requirements for running multicast applications on an emulated network.

Section 5.3 describes how to connect operational hosts running a routing protocol to emulated nodes within EXata.

Section 5.4 describes how to run applications that need to access the Internet or other networks via an emulated network.

Section 5.5 describes the how to use network management protocols in EXata.

Section 5.6 describes the EXata packet sniffing interface that enables any third party packet sniffing and analysis software to capture and display the traffic within an emulated network.

Section 5.7 describes how to set up an EXata emulation testbed on parallel and distributed platforms.

5.1 Configuring the EXata Testbed

To connect hardware-in-the-loop and to run applications on emulated nodes, including Internet-based applications and network managers, it is required to configure a testbed consisting of one or more operational hosts and an emulation server. Configuring the testbed requires configuration of the network (see Section 5.1.1) and establishing a connection between each operational host and the emulation server (see Section 5.1.2).

5.1.1 Network Configuration

This section describes how to configure the network for the EXata testbed. Section 5.1.1.1 describes the procedure for configuring the network when the operational hosts and the emulation server are in the same LAN. Section 5.1.1.2 describes the steps for configuring the network when the operational hosts and the emulation server are in different LANs.

5.1.1.1 Network Configuration for a Single LAN

This section describes how to configure the testbed network if the operational hosts and the emulation server are in the same LAN. Section 5.1.1.1.1 describes the procedure when all operational hosts have the same subnet address. Section 5.1.1.1.2 describes the procedure when the operational hosts have different subnet addresses.

5.1.1.1.1 Connecting Operational Hosts with Same Subnet Address

If all operational hosts and the emulation server are in the same LAN and all operational hosts have the same subnet address, then ensure that the emulation server also has the same subnet address. No other configuration is required for the emulation server.

5.1.1.1.2 Connecting Operational Hosts with Different Subnet Addresses

If all operational hosts and the emulation server are in the same LAN but the operational hosts have different subnet addresses, then multiple IP addresses need to be assigned to the interface on the emulation server: one from each of the subnets to which the operational hosts belong. Section 5.1.1.1.2.1 describes how to assign multiple IP addresses to an interface on a Windows platform. Section 5.1.1.1.2.2 describes how to assign multiple IP addresses to an interface on a Linux platform.

As an example, consider the emulation test-bed configuration shown in Figure 5-1. The emulation server has one physical network card with assigned address 10.10.0.100. There are two operational hosts, with addresses 10.10.0.1 and 20.20.0.1, both of which must connect to the emulation server. In this configuration, whereas Operational Host 1 can connect to the emulation server, Operational Host 2 will not be able to establish a connection because the IP address of the emulation server belongs to a different subnet.

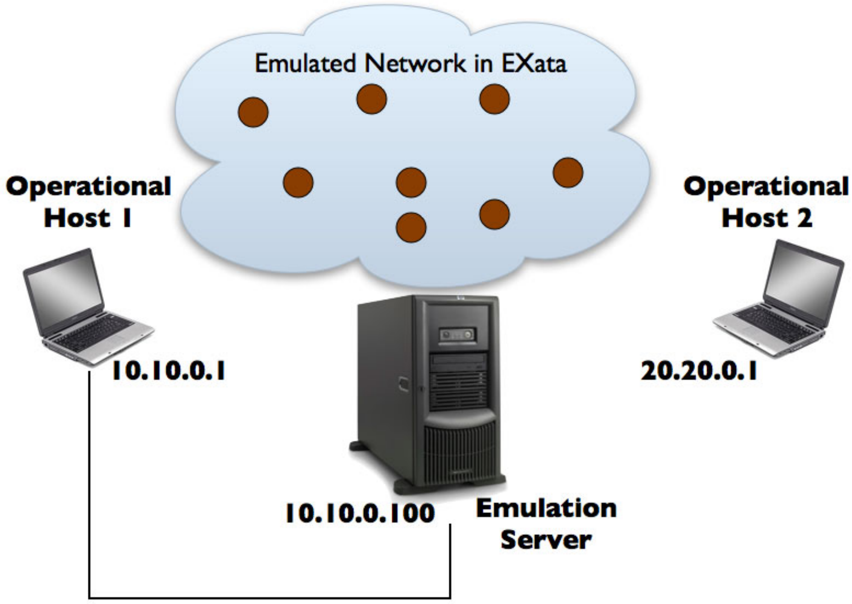


FIGURE 5-1. Operational Hosts with Different Subnet Addresses

Following the procedure described in this section, the interface on the emulation server can be assigned an additional IP address which is in the same subnet as Operational Host 2. Operational Host 2 can now establish a connection with the emulation server as well, as shown in Figure 5-2.

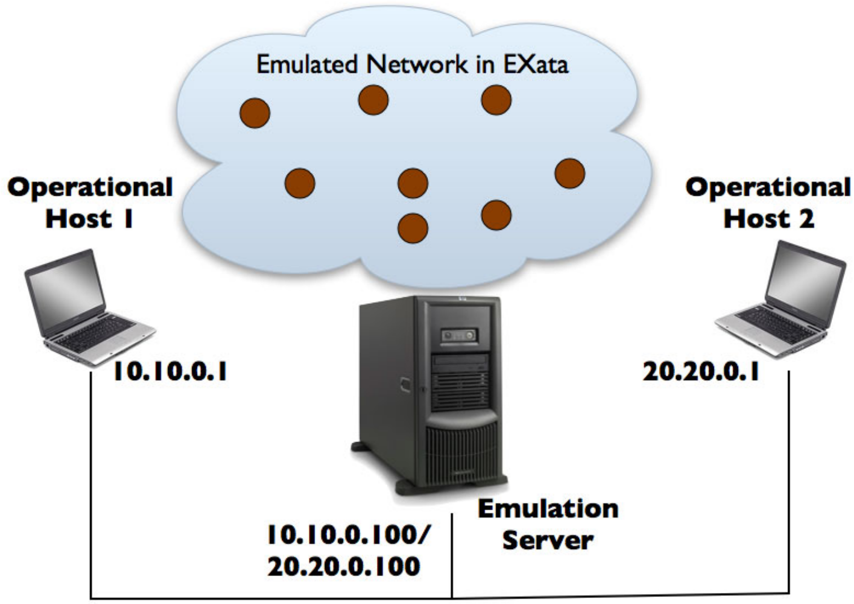


FIGURE 5-2. Multiple IP Addresses Assigned to a Shared Interface

5.1.1.1.2.1 Emulation Server Configuration in Windows

To assign multiple IP addresses to a shared interface of the emulation server running on a Windows 7 or Windows 8 platform, perform the following steps:

1. Ensure that the shared interface on the emulation server is assigned an IP address in the same subnet as one of the operational hosts.
2. Identify the additional IP addresses that must be configured for the shared interface on the emulation server. This can be achieved by identifying the operational hosts that share the same interface but belong to other subnets and selecting an IP address in each of those subnets.
3. To assign an IPv4 address, do the following:
   a. Go to **Start > Control Panel > Network and Sharing Center > Change adapter settings**. Right-click on the desired network interface and select **Properties**.
   b. In the **Networking** tab, select **Internet Protocol Version 4 (TCP/IPv4)** from the scroll down list and click on the **Properties** button. This opens the **Internet Protocol Version 4 (TCP/IPv4) Properties** editor (see Figure 5-3).
   c. Select **Use the following IP address**. You may notice some IP address already assigned in the editor.

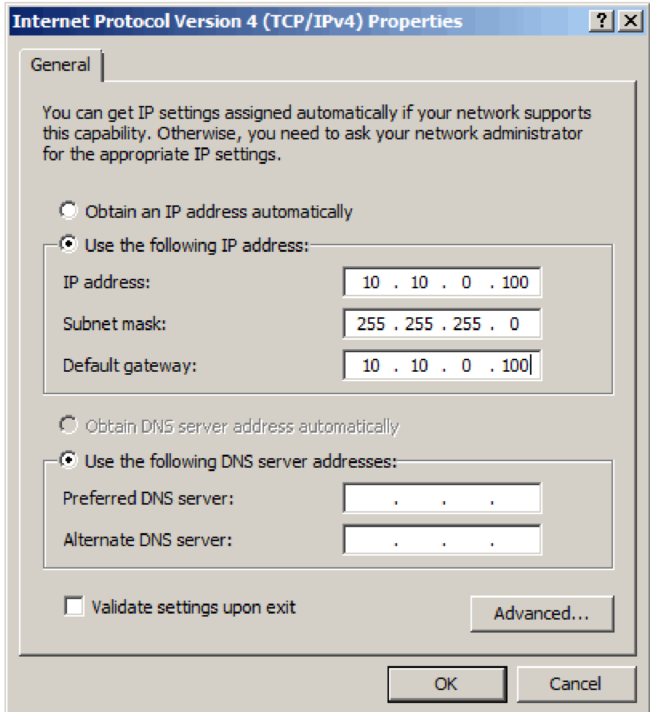


FIGURE 5-3. Internet Protocol Version 4 (TCP/IPv4) Properties Editor

d. Click on the **Advanced** button. This opens the **Advanced TCP/IP Settings** editor. Click on the **IP Settings** tab (see Figure 5-4).

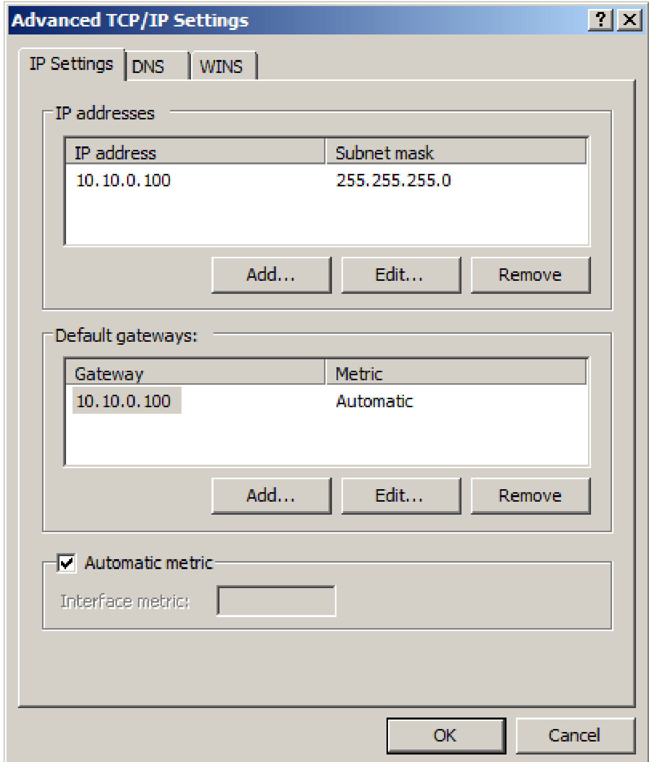


FIGURE 5-4. Advanced TCP/IP Settings Editor for IPv4

e. Click on the **Add** button under the **IP addresses** section. This opens the **TCP/IP Address** editor (see Figure 5-5).

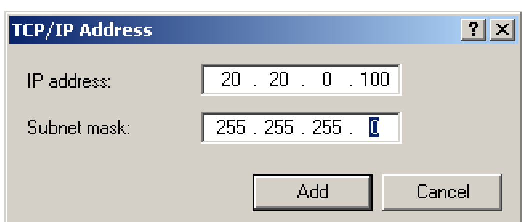


FIGURE 5-5. TCP/IP Address Editor for IPv4

f. Enter the IPv4 address and subnet mask to assign to the interface and click **Add**.

g. Repeat the previous two steps to assign any additional IPv4 addresses to the same interface.

h. Click **OK** on the open editors to finish the configuration.

4. To assign an IPv6 address, do the following:

a. Go to **Start > Control Panel > Network and Sharing Center > Change adapter settings**. Right-click on the desired network interface and select **Properties**.

b. In the **Networking** tab, select **Internet Protocol Version 6 (TCP/IPv46)** from the scroll down list and click on the **Properties** button. This opens the **Internet Protocol Version 6 (TCP/IPv6) Properties** editor (see Figure 5-6).

c. Select **Use the following IPv6 address**. You may notice some IP address already assigned in the editor.

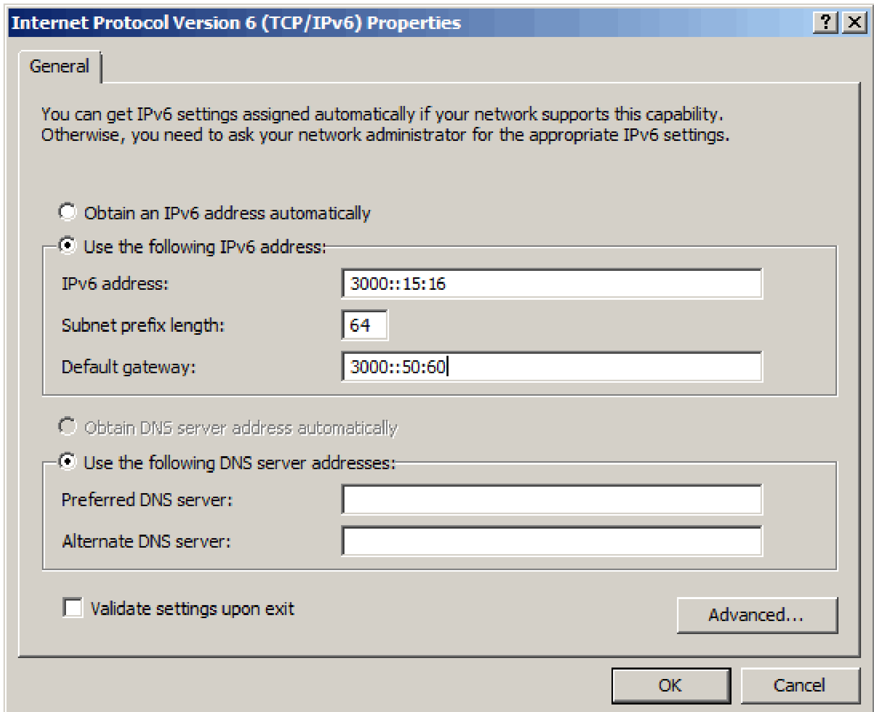


FIGURE 5-6. Internet Protocol Version 6 (TCP/IPv6) Properties Editor

d. Click on the **Advanced** button. This opens the **Advanced TCP/IP Settings** editor. Click on the **IP Settings** tab (see Figure 5-4).

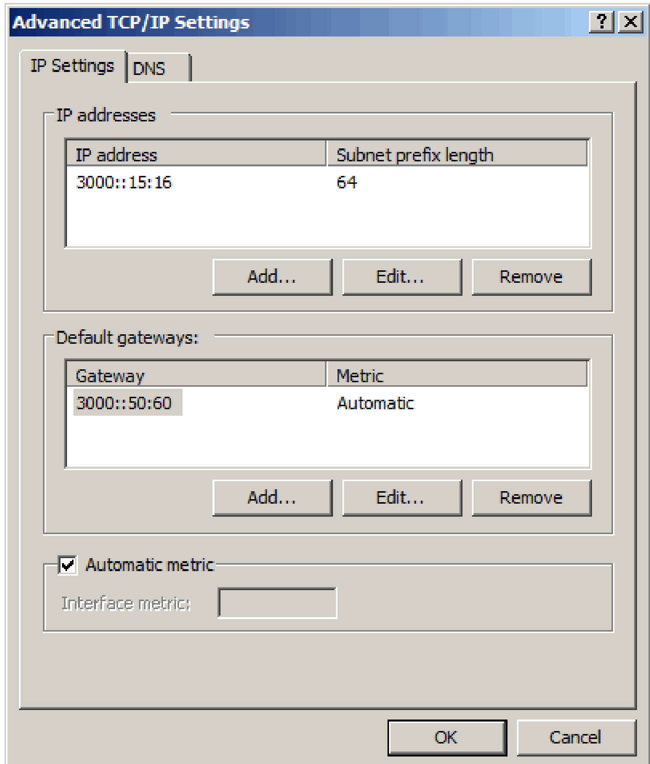


FIGURE 5-7. Advanced TCP/IP Settings Editor for IPv6

e. Click on the **Add** button under the **IP addresses** section. This opens the **TCP/IP Address** editor (see Figure 5-5).

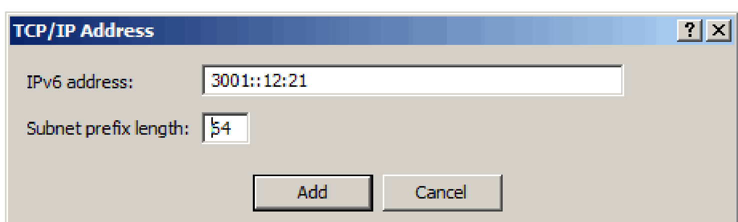


FIGURE 5-8. TCP/IP Address Editor for IPv6

f. Enter the IPv6 address and subnet prefix length to assign to the interface and click **Add**.

g. Repeat the previous two steps to assign any additional IPv6 addresses to the same interface.

h. Click **OK** on the open editors to finish the configuration.

To verify the IP address assignment, open a command window and type **`ipconfig`**. You should see all the assigned addresses for the network interface.

To verify connectivity, ping the emulation server from the operational hosts. Use the subnet-specific IP address of the emulation server interface when pinging from each operational host.

5.1.1.1.2.2 Emulation Server Configuration in Linux

To assign multiple IP addresses to a shared interface of the emulation server running on a Linux platform, perform the following steps:

1. Ensure that the emulation server interface is assigned an IP address in the same subnet as one of the operational hosts.
2. Identify the additional IP addresses that must be configured for the shared interface on the emulation server. This can be achieved by identifying the operational hosts that share the same interface but belong to other subnets and selecting an IP address in each of those subnets.
3. To assign an additional IPv4 address to the interface, open a command window and type:

```
ifconfig <device>:<address-num> <IPv4-address> netmask <subnet-mask>
```

where

<device>	Name of the network interface device that is used to connect to the operational hosts, e.g., `eth0`
<address-num>	Address index of the IP address to be assigned to the interface, e.g., `1`, `2`. A unique address index should be used for each IP address assigned to the interface.
<IPv4-address>	IPv4 address to be assigned to the interface
<subnet-mask>	Subnet mask to be assigned to the interface

Repeat this for every additional IPv4 address that needs to be assigned to the interface. For example, the following commands assign two additional IPv4 addresses to interface `eth0`:

```
ifconfig eth0:1 20.20.0.100 netmask 255.255.255.0
ifconfig eth0:2 30.30.0.100 netmask 255.255.255.0
```

4. To assign an additional IPv6 address to the interface, open a command window and type:

```
ifconfig <device> inet6 add <IPv6-address>/<prefix-length>
```

where

<device>	Name of the network interface device that is used to connect to the operational hosts, e.g., `eth0`
<IPv6-address>	IPv6 address to be assigned to the interface
<prefix-length>	Prefix length

Repeat this for every additional IPv6 address that needs to be assigned to the interface. For example, the following commands assign two additional IPv6 addresses to interface `eth0`:

```
ifconfig eth0 inet6 add 3000::15:16/64
ifconfig eth0 inet6 add 3001::12:21/64
```

To verify the IP address assignment, open a command window and type `ifconfig`. You should see all the assigned addresses for the network interface.

To verify connectivity, ping the emulation server from the operational hosts. Use the subnet-specific IP address of the emulation server interface when pinging from each operational host.

5.1.1.2 Network Configuration for Multiple LANs

This section describes how to configure the testbed network if the operational hosts and the emulation server are in different LANs (see Figure 5-9).

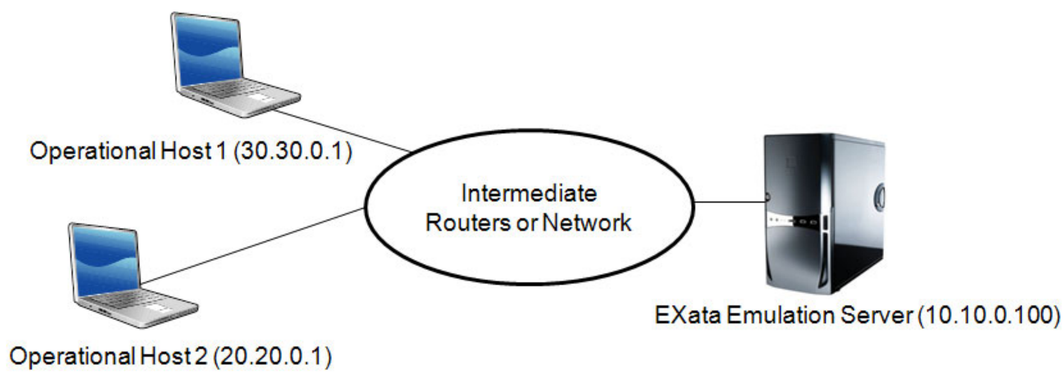


FIGURE 5-9. Connecting Operational Hosts and Emulation Server in Different LANs

In such a network configuration, a Virtual Private Network (VPN) must be created between each operational host and the emulation server. Furthermore, it must be ensured that the VPN tunnel supports Ethernet encapsulation.The scripts provided for OpenVPN with EXata ensure Ethernet encapsulation. For other VPN software, refer to the product documentation.

The rest of this section describes how to use OpenVPN software (http://www.openvpn.net) to create a VPN. For other VPN products and solutions, refer to the product's documentation.

Section 5.1.1.2.1 describes how to install the VPN software and Section 5.1.1.2.2 describes how to configure a VPN on operational hosts and the emulation server.

5.1.1.2.1 Installing OpenVPN

OpenVPN must be installed on all operational hosts and the emulation server.

5.1.1.2.1.1 Installing OpenVPN on Windows

To install OpenVPN on a Windows platform, do the following:

1. Download the Windows installer from http://www.openvpn.net/index.php/open-source/downloads.html.
2. Run the installer.

   Note: You must have administrative privileges to run the OpenVPN installer.

5.1.1.2.1.2 Installing OpenVPN on Linux

If the Linux distribution provides a command line utility (e.g., *apt-get*, *yum*) or a GUI-based utility to download and install software, you may use it to install the OpenVPN package from http://www.openvpn.net/index.php/open-source/downloads.htm.

Otherwise, perform the following steps to install OpenVPN:

1. Download the OpenVPN tarball from http://www.openvpn.net/index.php/open-source/downloads.html.
2. Open a command window and change the directory to the location where the OpenVPN package was downloaded.
3. Use the following command to extract the software:

```
tar xzvf openvpn-<version>.tar.gz
```

   where **<version>** is the version number of the downloaded OpenVPN tarball.

4. Use the following commands to install the software:

```
./configure
make
make install
```

Note: You may also need to install LZO library on your system. Refer to the download webpage or the README file in OpenVPN package for downloading and installing this library.

5.1.1.2.2 Configuring VPN

This section describes how to configure VPN on the operational hosts and the emulation server.

5.1.1.2.2.1 Verifying Connectivity

Before configuring and running OpenVPN, verify that all the operational hosts can connect to and are reachable from the emulation server. This can be accomplished by pinging each operational host from the the emulation server and pinging the emulation server from each operational host.

5.1.1.2.2.2 Running OpenVPN on the Emulation Server

Running OpenVPN on Windows

To run OpenVPN on an emulation server running on a Windows platform, do the following:

1. Open a command window and change the directory to EXATA_HOME/interfaces/lib-emulation/openvpn.
2. Type the following command:

```
openvpn.exe server-config-windows
```

To verify successful execution, open another command window and type **ipconfig**. You should see a new network interface named `Ethernet TAP 32`, with an IPv4 address of 192.168.0.1.

Running OpenVPN on Linux

To run OpenVPN on an emulation server running on a Linux platform, do the following:

1. Open a command window and change the directory to EXATA_HOME/interfaces/lib-emulation/openvpn.
2. Type the following command:

```
openvpn server-config-linux
```

To verify successful execution, open another command window and type **ifconfig**. You should see a new network interface named `tap1`, with an IPv4 address of 192.168.0.1.

5.1.1.2.2.3 Running OpenVPN on Operational Hosts

This section describes how to run OpenVPN on an operational host.

Note: You must start OpenVPN on the emulation server before running it on an operational host.

Running OpenVPN on Windows

To run OpenVPN on an operational host running on a Windows platform, do the following:

1. Copy the EXATA_HOME/interfaces/lib-emulation/openvpn folder from emulation server to the operational host.
2. Open a command window and change the directory to the openvpn folder.
3. In a text editor, open the file client-config-windows in the openvpn folder and locate the following line:

```
remote <IP Address>
```

Change `<IP Address>` to the IPv4 address of the emulation server.

4. Save and close this file.
5. Type the following command:

```
openvpn.exe client-config-windows
```

To verify successful execution, open another command window and type **ipconfig**. You should see a new network interface named `Ethernet TAP 32`, with an IPv4 address belonging to the subnet with subnet address 192.168.0.0/8.

Running OpenVPN on Linux

To run OpenVPN on an operational host running on a Linux platform, do the following:

1. Copy the EXATA_HOME/interfaces/lib-emulation/openvpn folder from emulation server to the operational host.
2. Open a terminal window and change the directory to the openvpn folder.
3. In a text editor, open the file client-config-linux in the openvpn folder and locate the following line:

```
remote <IP Address>
```

Change `<IP Address>` to the IPv4 address of the emulation server.

4. Save and close this file.

5. Type the following command:

```
openvpn client-config-linux
```

To verify successful execution, open another command window and type **ifconfig**. You should see a new network interface named `tap1`, with an IPv4 address belonging to the subnet with subnet address 192.168.0.0/8.

5.1.1.2.2.4 Verifying VPN Connection

To verify the VPN connection, type the following in a command window on each operational host:

```
ping 192.168.0.1
```

This will attempt to connect to the VPN interface on the emulation server.

5.1.2 Connecting Operational Hosts with the Emulation Server

This section describes how to configure operational hosts to establish a connection with the emulation server, either by using the Connection Manager application (see Section 5.1.2.1) or manually (see Section 5.1.2.2).

5.1.2.1 Configuration Using Connection Manager

The Connection Manager application runs on an operational host and is responsible for managing the operational host's networking configuration. Connection Manager is used to establish a connection between the host and the emulation server. When a connection is established, Connection Manager displays the list of emulated network nodes. Connection Manager can also be used to run multiple applications on different emulated nodes from the same operational host (see Section 5.2.1.1.2).

Unless the operational host is unable to run Connection Manager (for example, on non-computing platforms such as routers, non-standard operating system environments such as real time systems, etc.), it is recommended to use Connection Manager to manage connections between operational hosts and the emulation server.

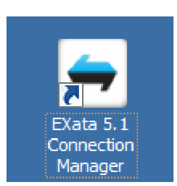
> **Note:** Connection Manager is downloaded separately from EXata and is installed on a machine different from the one running EXata. Refer to *EXata Installation Guide* for instructions for installing Connection Manager on the operational hosts.

The rest of this section describes how to launch Connection Manager in Windows and Linux, and how to configure the emulation server manually and automatically.

Starting Connection Manager in Windows

To start Connection Manager in Windows, do one of the following:

- Double-click the EXata Connection Manager icon on the Windows desktop.



- Select **Start > All Programs > EXata-Connection-Manager 5.1 > Connection Manager 5.1**.
- Open a command-line window, change the directory to C:\Program Files\Scalable\EXata\5.1 (or the location where Connection Manager is installed), and type **`exata-connection-manager.exe`**.

Note: You need administrator privileges to run the EXata Connection Manager.

If you are logged in as a non-Administrator on Windows, you can right-click Connection Manager icon, select **Run as..** and provide the Administrator account name and password.

Starting Connection Manager in Linux

To start Connection Manager in Linux, open a command-line window, type **`cd ~/EXata-Connection-Manager/5.1`** (or the location where Connection Manager is installed), and type **`./exata-connection-manager`**.

Note: You need root privileges to run EXata Connection Manager.

If you are logged in as a non-root account on Linux, you can launch Connection Manager by typing **`sudo ./exata-connection-manager`** and providing the root password.

Upon launching Connection Manager, a connection can be established between the operational host and the emulation server, either by manually providing the address of the emulation server, or by letting Connection Manager automatically detect the emulation server.

Manual Specification of Emulation Server

Before manually specifying the emulation server in ConnectionManager, ensure the following:

- The operational host and the emulation server are connected with each other (you can use the **`ping`** command to verify this).
- On the emulation server, the EXata process is not blocked by the firewall.
- On the operational host, Connection Manager is not blocked by the firewall.

To specify the emulation server manually, provide the address of the emulation server to Connection Manager as follows:

1. Start Connection Manager and select **Manually configure EXata hostname or IP Address**.

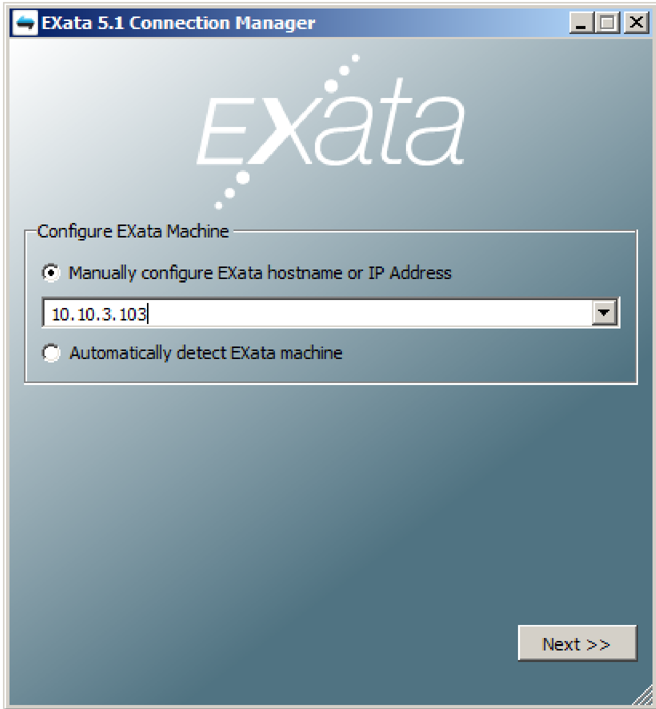


FIGURE 5-10. Manual Specification of Emulation Server

2. Enter the IP address (e.g., 10.10.3.103 or fe80::216:35ff:fe64:3a5a) or host name (e.g., server01.mydomain) of the emulation server. You can also select from previously entered addresses from the pull down menu.

Note: If you are providing a hostname, ensure that the operational host is reachable from the local DNS server.

3. Click **Next**.

   Connection Manager will attempt to configure the network settings on the operational host to create an emulated link between this host and the emulation server. If Connection Manager is successful in creating a connection, the address configuration screen will be replaced with Network and Application configuration screen (see Figure 5-11). If this is the case, skip the remaining steps.

   (If EXata is not running on the emulation server, the left screen in Figure 5-11 is displayed. If EXata is running on the emulation server, a screen similar to the right screen in Figure 5-11 is displayed.)

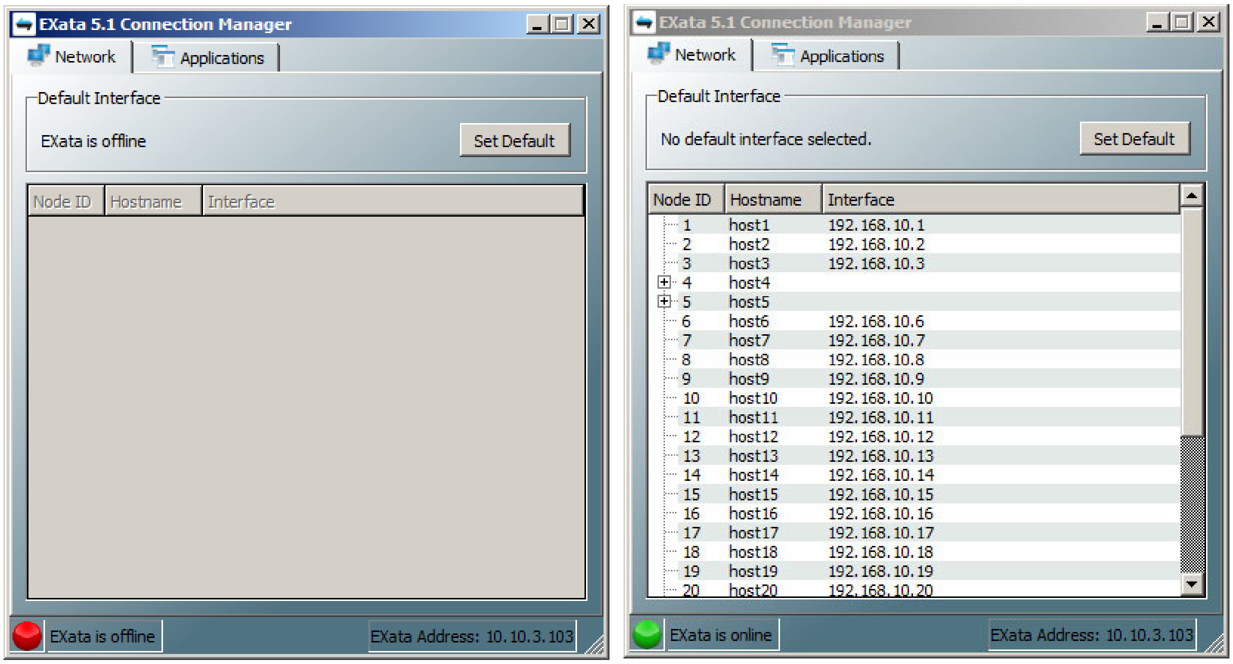


FIGURE 5-11. Emulation Server Successfully Connected

If, however, Connection Manager is unable to create a connection, it will display a message (see Figure 5-12) asking for further network configuration information.

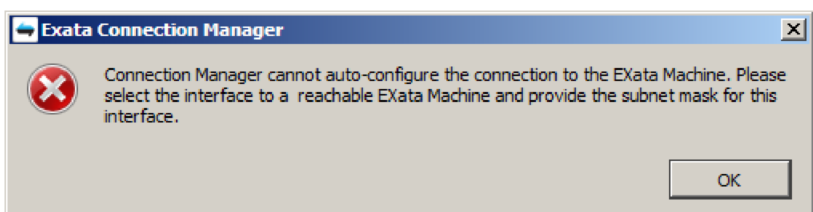


FIGURE 5-12. Connection Manager Failed to Connect to Emulation Server

4. If Connection Manager is unsuccessful in creating a connection, click **OK** on the displayed message (see Figure 5-12). This changes the Connection Manager display (see Figure 5-13) to let you enter additional configuration information. (The left screen in Figure 5-13 is for IPv4 connections and the right screen is for IPv6 connections.)

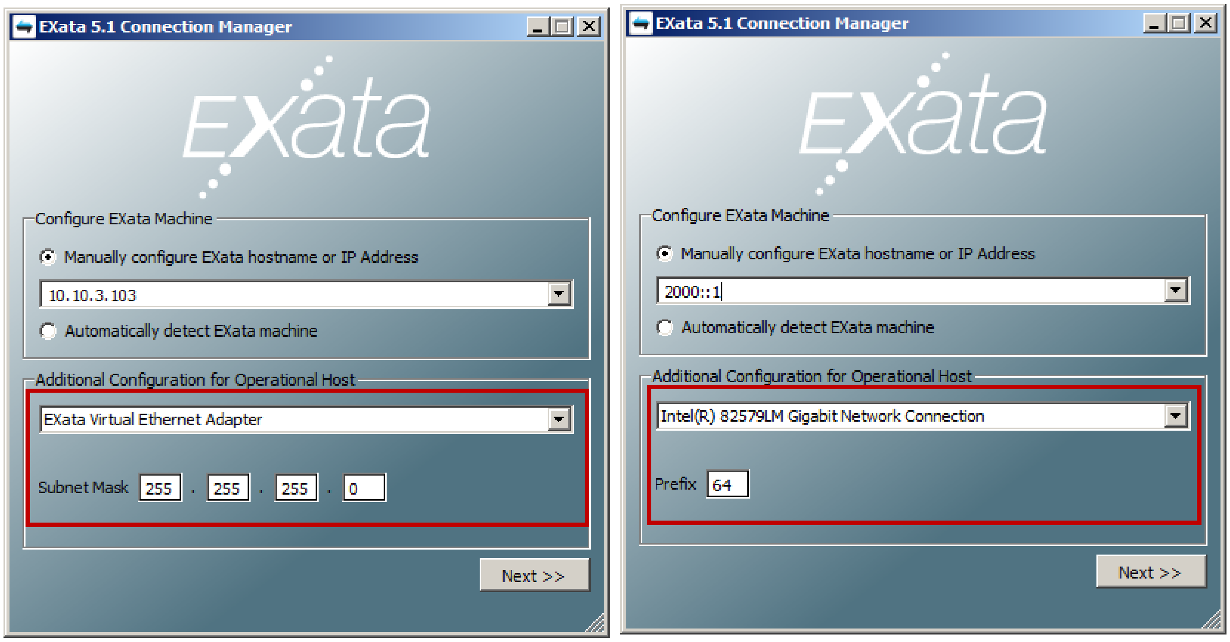


FIGURE 5-13. Advanced Configuration Options

a. Connection Manager lists the available network interfaces available on the operational host. From the pull-down menu, choose the interface that has been configured in the same subnet as the emulation server.
b. Enter the subnet mask (for IPv4) or prefix (for IPv6) of the network interface device *on the emulation server* that is used to connect to this operational host.

Note: To identify this subnet mask, open a command-line window and type `ipconfig`. Note the subnet mask for the appropriate network interface.

c. Click **Next**.
d. If Connection Manager is unable to generate a valid IP address in the same subnet as the emulation server, it will display a message (see Figure 5-14) asking to enter a valid IP address.

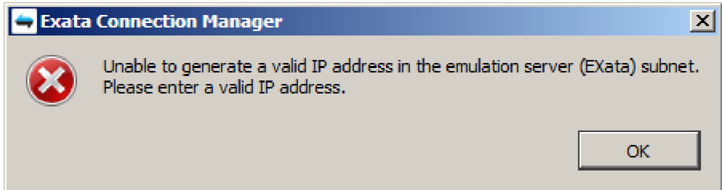


FIGURE 5-14. Connection Manager Failed to Generate a Valid IP Address

If this happens, click **OK** on the displayed message. This changes the Connection Manager display to let you enter a valid IP address and subnet mask (for IPv4) or network prefix (for IPv6) to be assigned to the selected interface.

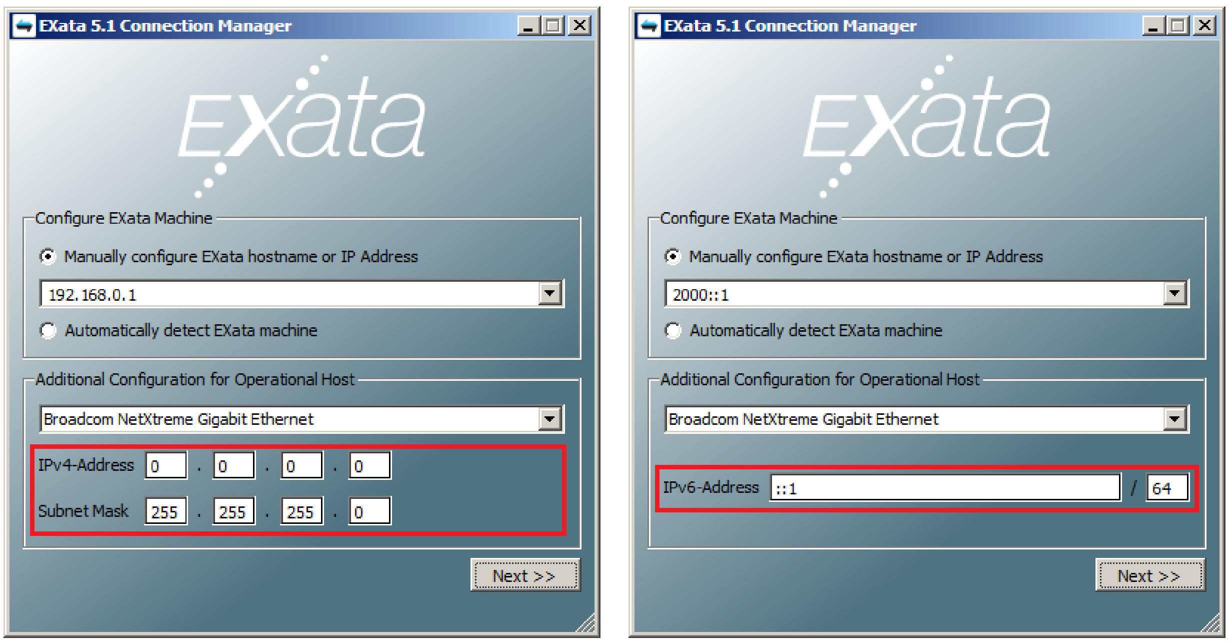


FIGURE 5-15. Assign Valid IP Address

Note: If Connection Manager still cannot connect to the emulation server, it is likely, that the operational host and the emulation server are not connected with each other. Check the network cables for proper connectivity and consult your network administrator

Automatic Detection of Emulation Server

Connection Manager can also auto-detect emulation servers in the same subnet. For this mode to work, ensure the following:

- A scenario is running on EXata, either via GUI or command line.
- The operational host and the emulation server are connected with each other (you can use the `ping` command to verify this).
- On the emulation server, the EXata process is not blocked by the firewall.
- On the operational host, Connection Manager is not blocked by the firewall.

If all the above conditions are met, you can automatically configure the emulation server by doing the following:

1. Start Connection Manager and select **Automatically detect EXata machine**.

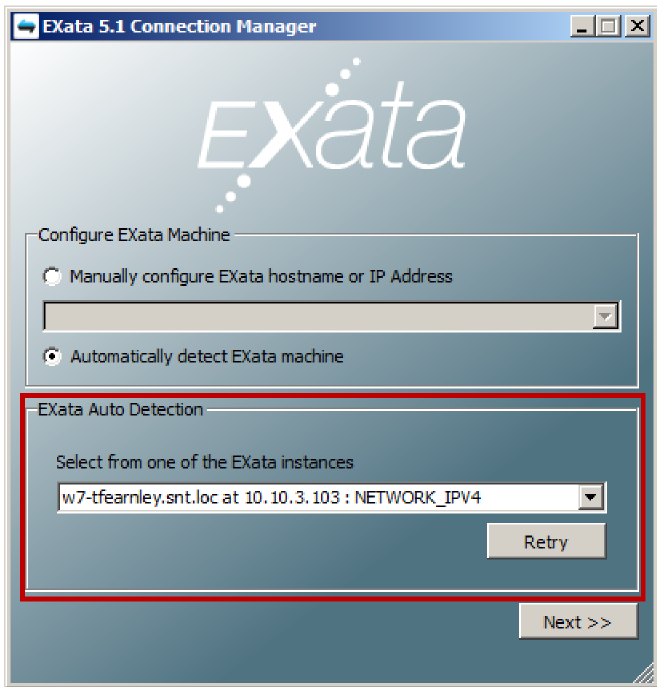


FIGURE 5-16. Automatically Detecting Emulation Servers

2. If Connection Manager is able to detect any running EXata processes, it will display all the detected emulation servers in a pull down menu (see Figure 5-16). If an IPv4 scenario is running on an emulation server, then the label **NETWORK_IPV4** appears next to the interface address of the emulation server. If an IPv6 scenario is running on an emulation server, then the label **NETWORK_IPV6** appears next to the interface address of the emulation server. If a mixed IPv4/IPv6 scenario is running on an emulation server, then there are two entries for the server in the list: one for IPv4 and the other for IPv6. Select the emulation server that you wish to connect to.
3. Click **Next**.

Verifying the Connectivity

After Connection Manager has successfully connected to the emulation server, start the scenario in EXata in *Emulation mode* if you have not already done so.

Connection Manager will indicate in its status bar that it has detected an EXata process (see Figure 5-17) and display a list of the nodes in the scenario being executed by EXata.

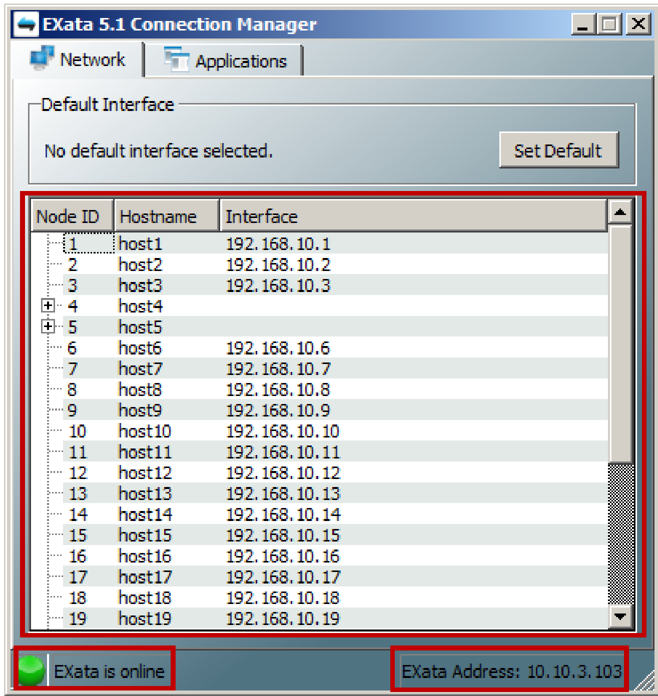


FIGURE 5-17. List of Emulated Nodes in Network Tab

Note: If a node has only one interface, then its address is displayed in the same row as the ID of the node. If a node has multiple interfaces, then the list of interfaces and their addresses can be displayed by clicking on the '+' sign before the node ID.

Connection Manager lists all nodes and interfaces in the scenario. If you have connected the operational host to the emulation server using an IPv4 (IPv6) address, then only emulated interfaces with IPv4 (IPv6) addresses are enabled and can be selected as the default external node (see Section 5.2.1.1.1).

When Connection Manager forms a connection to an EXata process, a message is also displayed in the system tray icon (see Figure 5-18).

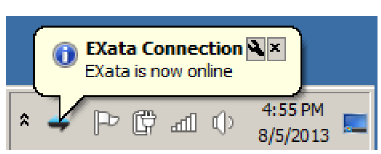


FIGURE 5-18. Connection Manager System Tray Icon

When the EXata execution is terminated (by pressing the **Stop** button in the GUI toolbar, or terminating the EXata process on the command line interface), the list of nodes will disappear and Connection Manager will indicate that EXata is offline, both in the status bar and the system tray.

5.1.2.2 Manual Configuration

Even though Connection Manager is the recommended approach for managing the operational host, there might be reasons where Connection Manager cannot be installed or executed on the operational host. For such cases, the operational host must be manually configured to connect with the emulation server. In addition, the external node mapping parameters must be configured as described in Section 4.3.1.

This section describes the manual configuration of the operational host.

Note: With this manual configuration, multiple applications can not be run on different emulated nodes (see Section 5.2.1.1.2).

5.1.2.2.1 Manual Configuration in Windows

To configure a Windows operational host, perform the following steps:

1. Assign an IP address to the operational host, if it is not already assigned one. The operational host should be in the same subnet as the emulated server. Ensure that the emulation server is reachable from the operational host. To verify this connectivity, open a command window and type:

   ```
   ping <IPv4-address>
   ```

   or

   ```
   ping -6 <IPv6-address>
   ```

   where

`<IPv4-address>`	IPv4 address of the emulation server
`<IPv6-address>`	IPv6 address of the emulation server

   Successful responses to the ping requests indicate a proper connection between the operational host and emulation server.

   If the ping program reports failure in connection, ensure that the operational host and emulation server are properly connected with network cables and they have IP addresses belonging to the same subnet. Consult the networking administrator if the problem still persists.

2. On the operational host, set the emulation server to be the default gateway.

   - To configure the default gateway for IPv4 networks on a Windows 7 or Windows 8 platform, do the following:

     a. Go to **Start > Control Panel > Network and Sharing Center > Change adapter settings**. Right-click on the desired network interface and select **Properties**.

     b. In the **Networking** tab, select **Internet Protocol Version 4 (TCP/IPv4)** from the scroll down list and click on the **Properties** button.

     c. Select **Use the following IPv4 address**. Configure the default gateway as shown in the Figure 5-19. The figure illustrates an example of an operational host with IPv4 address `10.200.0.45`, subnet mask `255.0.0.0`, and default gateway `10.200.0.100`, which is the IPv4 address of the emulation server.

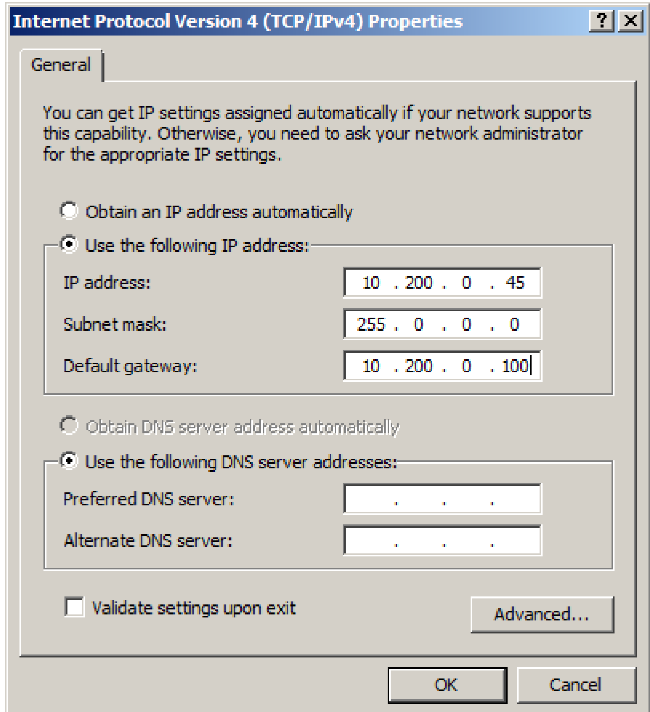


FIGURE 5-19. Setting Default IPv4 Gateway in Windows Operational Host

- To configure the default gateway for IPv6 networks on a Windows 7 or Windows 8 platform, do the following:
  - **a.** Go to **Start > Control Panel > Network and Sharing Center > Change adapter settings**. Right-click on the desired network interface and select **Properties**.
  - **b.** In the **Networking** tab, select **Internet Protocol Version 6 (TCP/IPv6)** from the scroll down list and click on the **Properties** button.
  - **c.** Select **Use the following IPv6 address**. Configure the default gateway as shown in the Figure 5-20. The figure illustrates an example of an operational host with IPv6 address `3000::15:16`, subnet prefix length `64`, and default gateway `3000::50:60`, which is the IPv6 address of the emulation server.

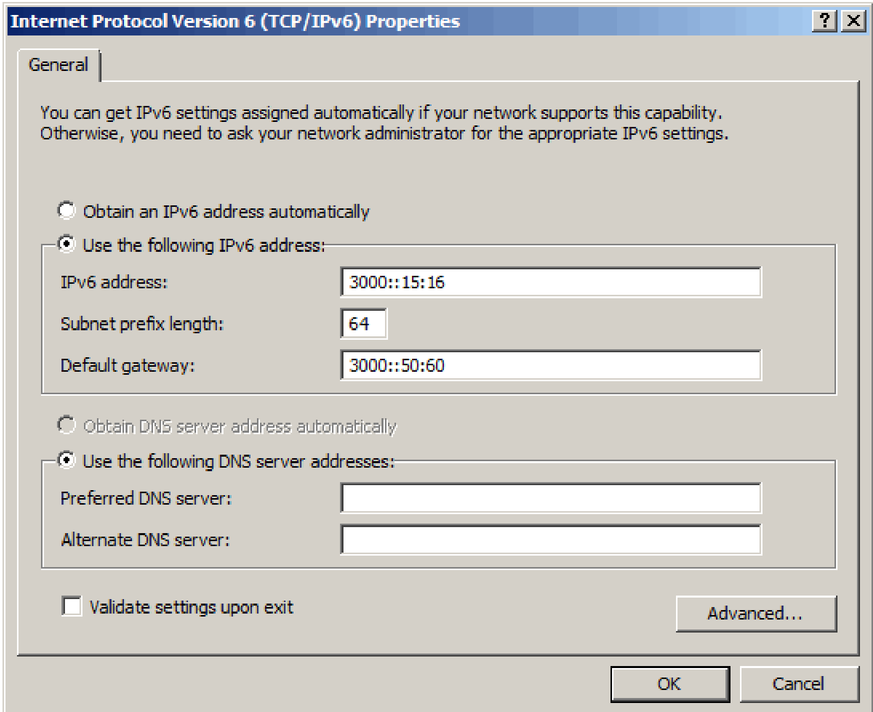


FIGURE 5-20. Setting Default IPv6 Gateway in Windows Operational Host

3. Set a default route to the emulation server.

- To set a default IPv4 route on a Windows 7 or Windows 8 platform, open a command window, and type:

```
route add <net-address> mask <subnet-mask> <IPv4-address>
```

where

<net-address>	Network address of the emulation server
<subnet-mask>	Subnet mask of the network
<IPv4-address>	IPv4 address of the emulation server

The network address is obtained by keeping the bits in the IP address that belong to the subnet mask, and clearing out everything else (also referred to as taking *bitwise-AND* of IP address and the subnet mask). For the example in Figure 5-19, the command is:

```
route add 10.0.0.0 mask 255.0.0.0 10.200.0.100
```

- To set a default IPv6 route on a Windows 7 or Windows 8 platform, open a command window, and type:

```
netsh interface ipv6 add route <net-address> <interface> <IPv6-address>
```

where

<net-address	IPv6 address of the emulation server with the subnet prefix length
<interface>	Interface name
<IPv6-address>	IPv6 address of the emulation server

For the example in Figure 5-20, the command is:

```
netsh interface ipv6 add route 3000::/64 "Local area connection 4"
3000::50:60
```

5.1.2.2.2 Manual Configuration in Linux

To configure a Linux operational host, perform the following steps:

1. Assign an IP address to the operational host, if it is not already assigned one. The operational host should be in the same subnet as the emulated server. Ensure that the emulation server is reachable from the operational host. To verify this connectivity, open a command window and type:

```
ping <IPv4-address>
```

or

```
ping6 <IPv6-address>
```

where

<IPv4-address>	IPv4 address of the emulation server
<IPv6-address>	IPv6 address of the emulation server

Successful responses to the ping requests indicate a proper connection between the operational host and emulation server.

If the ping program reports failure in connection, ensure that the operational host and emulation server are properly connected with network cables and they have IP addresses belonging to the same subnet. Consult the networking administrator if the problem still persists.

2. On the operational host, set the emulation server to be the default gateway.

- To configure a default IPv4 gateway, open a command window, and type (all parameters must be entered on the same line):

```
sudo route add default gw <IPv4-address>
```

  where

`<IPv4-address>`	IPv4 address of the emulation server

  As an example, consider the IP address of the operational host as `10.200.0.45` and that of the emulation server as `10.200.0.100`, then the default gateway can be configured as:

```
sudo route add default gw 10.200.0.100
```

- To configure a default IPv6 gateway, open a command window, and type (all parameters must be entered on the same line):

```
sudo route -A inet6 add <net-address> gw <IPv6-address> dev
<interface>
```

  where

`<net-address`	Network address with the subnet prefix length
`<IPv6-address>`	IPv6 address of the emulation server
`<interface>`	Interface name

  For the example in Figure 5-20, the command is:

```
sudo route -A inet6 add ::/64 gw 3000::50:60 dev eth0
```

Note: If you are logged in as root in the terminal window, you can omit the **sudo** keyword.

3. Set a default route to the emulation server.

- To set a default IPv4 route, open a command window, and type (all parameters must be entered on the same line):

```
sudo route add –net <net-address> netmask <subnet-mask> gw
<IPv4-address>
```

  where

`<net-address>`	Network address of the emulation server
`<subnet-mask>`	Subnet mask of the network
`<IPv4-address>`	IPv4 address of the emulation server

The network address is obtained by keeping the bits in the IP address that belong to the subnet mask, and clearing out everything else (also referred to as taking *bitwise-AND* of IP address and the subnet mask). For the example in Figure 5-19, the command is:

```
sudo route add -net 10.200.0.0 netmask 255.255.255.0 gw 10.200.0.100
```

- To set a default IPv6 route, open a command window, and type (all parameters must be entered on the same line):

```
sudo route -A inet6 add <net-address> gw <IPv6-address> dev
<interface>
```

where

`<net-address`	Network address with the subnet prefix length
`<IPv6-address>`	IPv6 address of the emulation server
`<interface>`	Interface name

For the example in Figure 5-20, the command is:

```
sudo route -A inet6 add 3000::/64 gw 3000::50:60 dev eth0
```

Note: If you are logged in as root in the terminal window, you can omit the **sudo** keyword.

5.2 Running Applications

This section describes how to set up applications at operational hosts to run on the emulated network.

Section 5.2.1describes how to launch applications on operational hosts. Section 5.2.2 describes the emulated server applications implemented in EXata. Section 5.2.3 describes the additional configuration requirements for running multicast applications on an emulated network.

Note: Additional instructions for Internet-based applications and network managers are provided in Section 5.3 and Section 5.5, respectively.

5.2.1 Launching Applications

This section describes how to launch applications on operational hosts that connect to EXata nodes within the emulated network scenario.

If you have connected the operational host with the emulation server using Connection Manager (see Section 5.1.2.1), follow the instructions in Section 5.2.1.1.

If you have connected the operational host with the emulation server manually (see Section 5.1.2.2), follow the instructions in Section 5.2.1.2.

5.2.1.1 Launching Applications Using Connection Manager

Connection Manager offers an easy mechanism to launch applications on the operational hosts that connect to the emulated nodes in EXata.

Connection Manager can be used to launch applications in two ways:

- Assign one emulated node in EXata to map to the operational host. All applications launched on the operational host will run on this emulated node (see Section 5.2.1.1.1).
- When launching an individual application, select the emulated node it should run on. In this way, different applications on the same operational host can connect to different emulated nodes (see Section 5.2.1.1.2).

5.2.1.1.1 Launching Applications on a Single Emulated Node

In this mode, the operational host can be a viewed as a physical counterpart of one of the emulated nodes: any application launched on this operational host will send and receive traffic to and from the protocol stack of that particular emulated node. To achieve this, it is required to configure the EXata emulator to create a one-to-one mapping between the emulated node and the operational host. This section describes how to create such a mapping using Connection Manager (Section 5.2.1.2 describes the how to create this mapping without using Connection Manager).

Figure 5-21 provides the schematic diagram for this mode of operation. A mapping between the operational host and one node in the emulated network (henceforth referred to as the *External Node*) is created. Subsequently, any application launched on the operational host will use the emulated protocol stack of the External Node for communications. Note that as many operational hosts are required as the desired number of External Nodes, since there is a unique one-to-one mapping between the operational hosts and External Nodes. Section 5.2.1.1.2 describes the EXata mode that removes this restriction.

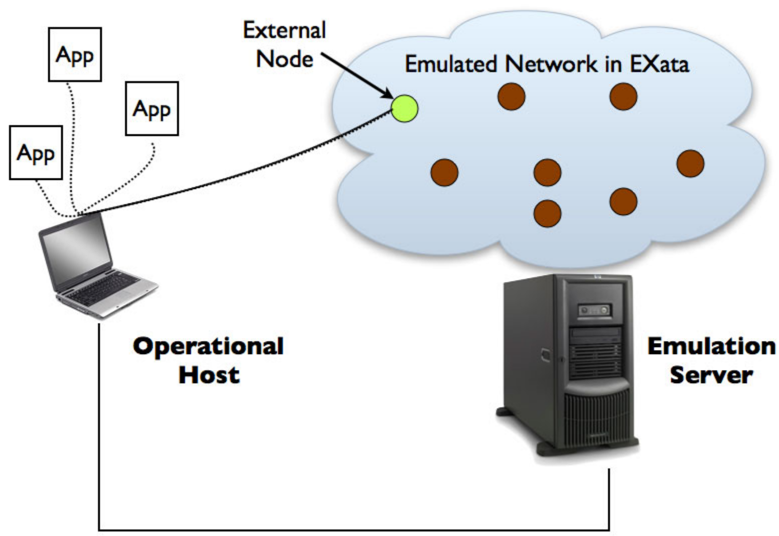


FIGURE 5-21. Applications on Operational Host Connecting to Default External Node in EXata

Setting a Default External Node

To create a one-to-one mapping between the operational host and an External Node in EXata, perform the following steps:

1. On the emulated server, run an EXata scenario in emulation mode, either from the GUI or command line interface. (In the rest of the section we assume that EXata was launched from GUI.)

   Connect the operational host and the emulated server using Connection Manager, as described in Section 5.1.2.1.

2. Wait a few seconds until Connection Manager recognizes that EXata is running and displays a list of all the nodes in the **Network** tab.

3. Select one of the nodes in Connection Manager's display window by clicking on it, and click the **Set Default** button.

   If an emulated node has multiple interfaces, then expand the display for the node and select one interface. Figure 5-22 shows an example where node 5 (host5) has two interfaces with addresses `192.168.10.5` and `192.0.1.1`.

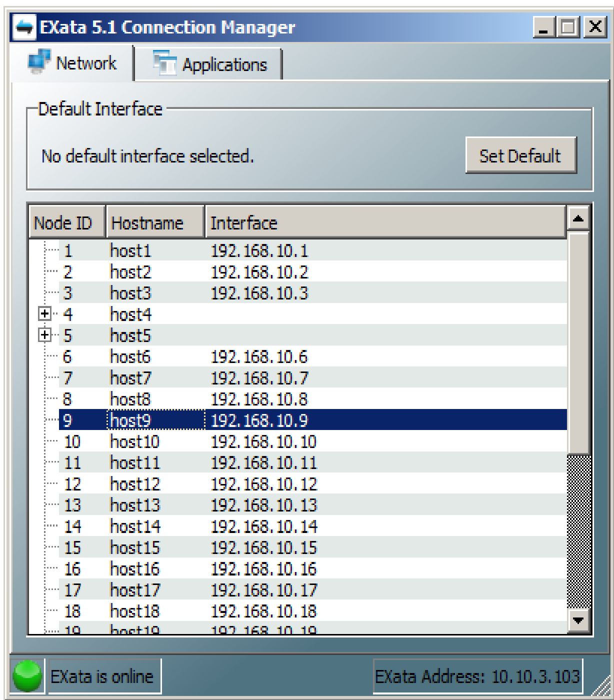


FIGURE 5-22. Setting Default External Node

The selected interface is displayed in the **Default Interface** box and is highlighted in the node list.

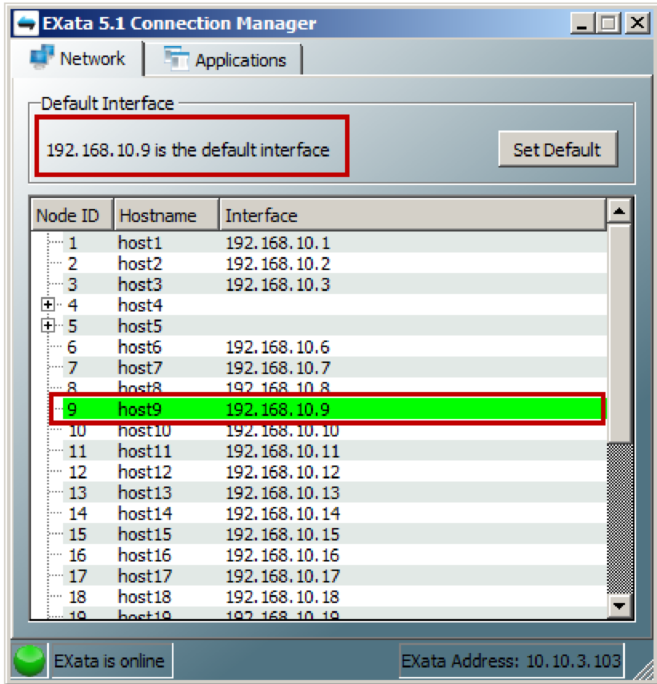


FIGURE 5-23. Default Node Configured

If EXata GUI is running, the selected node is highlighted on the canvas by a purple triangle (see node 3 in Figure 5-24).

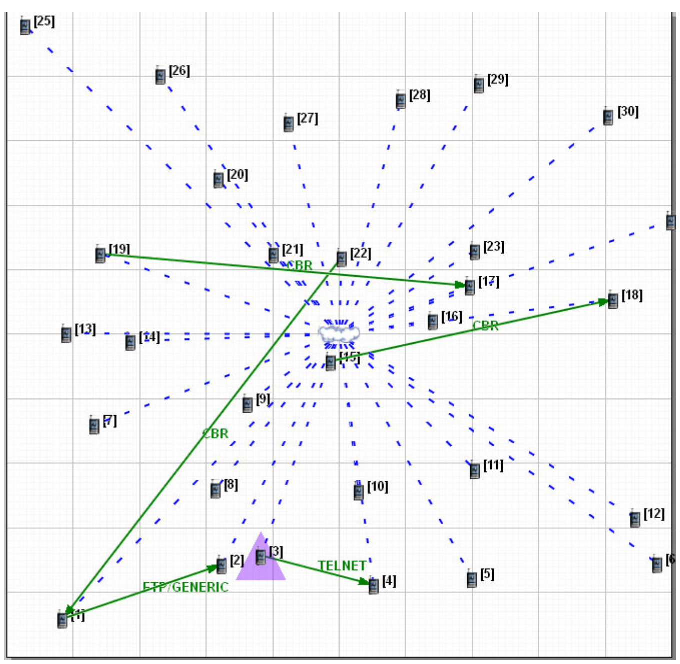


FIGURE 5-24. EXata GUI Displaying the Default External Node

4. Verify by connecting to other nodes in the scenario by using one of the EXata emulated server applications (see Section 5.3).

An application run from the operational host will now also run on the default node.

Unsetting the External Node

To tear down an established connection between the operational host and the External Node, you can unset the default node by following these steps:

1. Click on the interface that is currently configured as the External Node.
2. The button text in the **Default Interface** text box will change to **Reset Default**. Click on the button.

   The **Default Interface** box displays that no default interface is set and the unset interface is no longer highlighted in the list.

   If EXata GUI is running, the node is no longer highlighted on the canvas.

Changing the External Node

While the scenario is running in EXata, you can change the operational host to connect to any other emulated node. This will enable the operational host to insert or receive traffic from different nodes in the scenario during the same execution run.

To change the External Node, do one of the following:

- Reset the current External Node and set a new one, as described above.
- Select the new node by clicking on it in the node list and click the **Set Default** button.

5.2.1.1.2 Launching Applications on Different Emulated Nodes

In this mode, when launching an application, the user can select on which emulated node in EXata the application should be executed. Thus, it is possible to run multiple applications on different emulated nodes, all from the same operational host. See Figure 5-25.

> **Note:** This feature is available only on operational hosts running Windows 7 (32-bit) or Windows 8 (32-bit and 64-bit) editions.

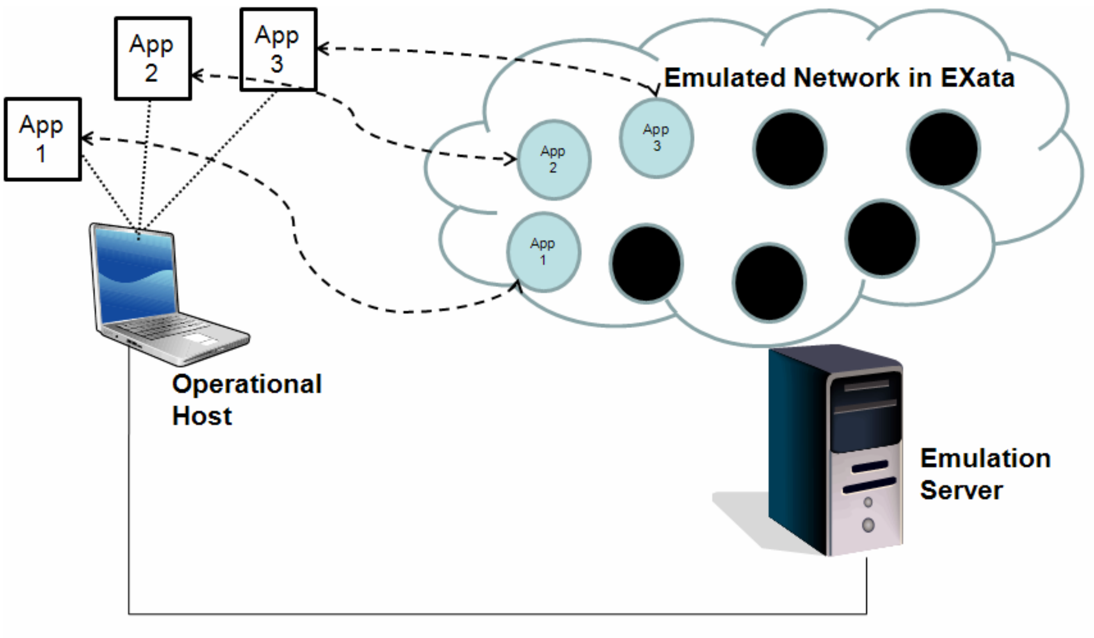


FIGURE 5-25. Multiple Applications on a Single Operational Host Connecting to Different Emulated Nodes

5.2.1.1.2.1 Launching an Application on a Selected Emulated Node

To select an emulated node on which to run an application at the time of launching an application, perform the following steps:

1. Set up a connection between the operational host and the emulation server using Connection Manager, as described in Section 5.1.2.1. Ensure that the scenario is running in EXata in Emulation mode and Connection Manager displays the list of nodes in the **Network** tab.
2. To launch an application, right-click on the application icon on the desktop and click on **Run on EXata node**.

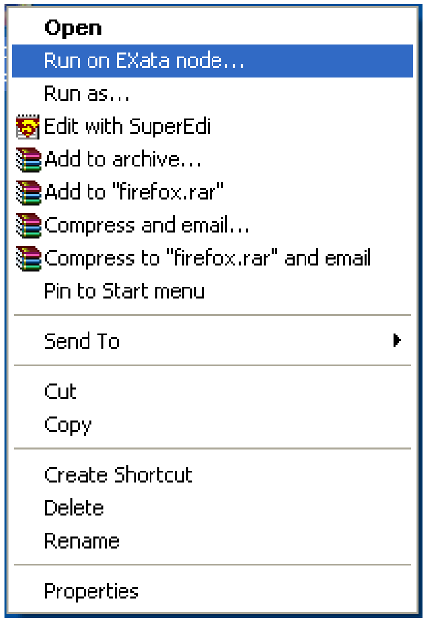


FIGURE 5-26. Running Application on an EXata Node

3. A dialog box appears that lists all the nodes in the scenario. Enter a node ID or hostname or select a node from the pull-down list and click **OK**.

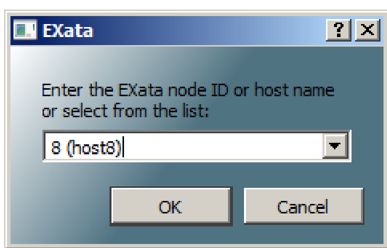


FIGURE 5-27. Selecting an Emulated Node

Once the application has started, the **Application** tab of Connection Manager will display the application name and the emulated node (called an *External Application Node*) on which it is configured to run.

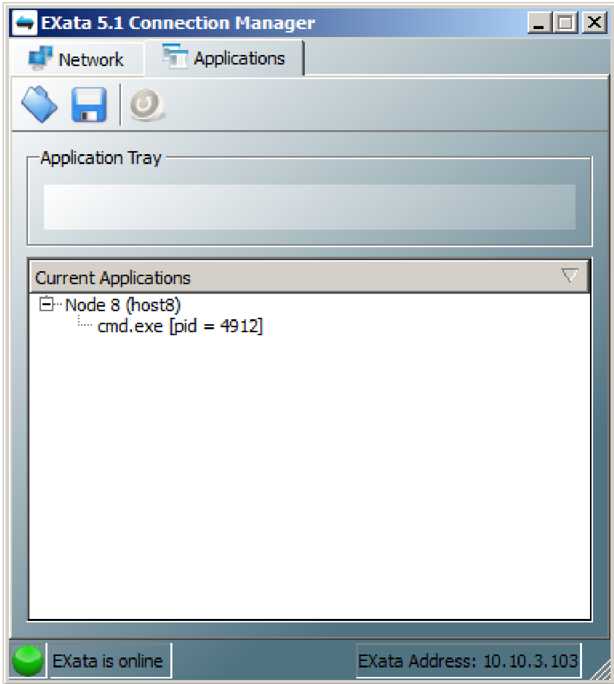


FIGURE 5-28. Application Tab of Connection Manager

Tip: You can double-click the application in the **Application** tab of Connection Manager to bring the application window to the front.

Note: To run another application from command line, first launch the command window application (cmd.exe) to run on the emulated node via the steps outlined above. Then, run the application from this command window. This approach works because any child process launched by a process that was configured to use EXata, will also continue to use the same configuration.

If EXata GUI is running, the selected node is highlighted on the canvas by a green triangle (see node 8 in Figure 5-29).

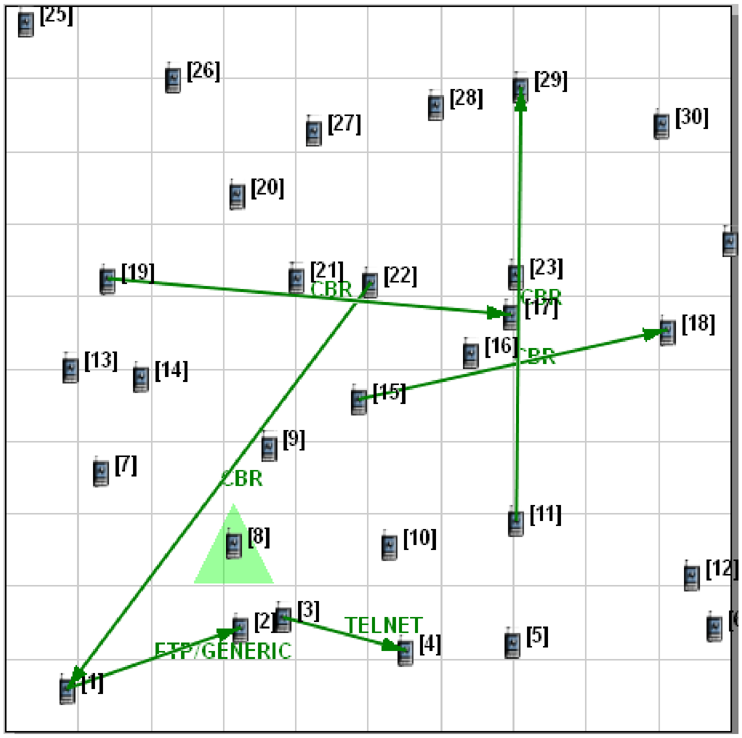


FIGURE 5-29. EXata GUI Displaying an External Application Node

5.2.1.1.2.2 Restoring an Application Session

An application session is the record of information for all the currently running applications. The information includes the name of application, the path of the application executable binary, and the ID of the emulated node it is running on. Figure 5-30 shows an example of application session of two applications running over two different emulated nodes.

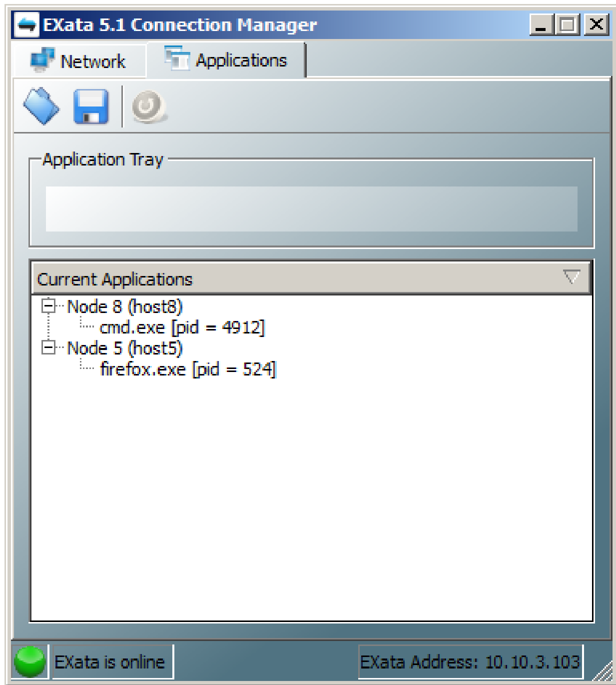


FIGURE 5-30. An Active Application Session

At the termination of the EXata process, either manually by the user or through the successful completion of the scenario, all applications running on the emulated nodes are terminated. However, before terminating the applications, Connection Manager saves the last application session. When the EXata process is restarted, Connection Manager provides the option to restore the last application session.

To restore the last application session from an EXata run, perform the following steps:

1. Restart the scenario in EXata in emulation mode from the GUI or the command line.
2. In the **Applications** window of Connection Manager, click on the **Restore** button in the toolbar.

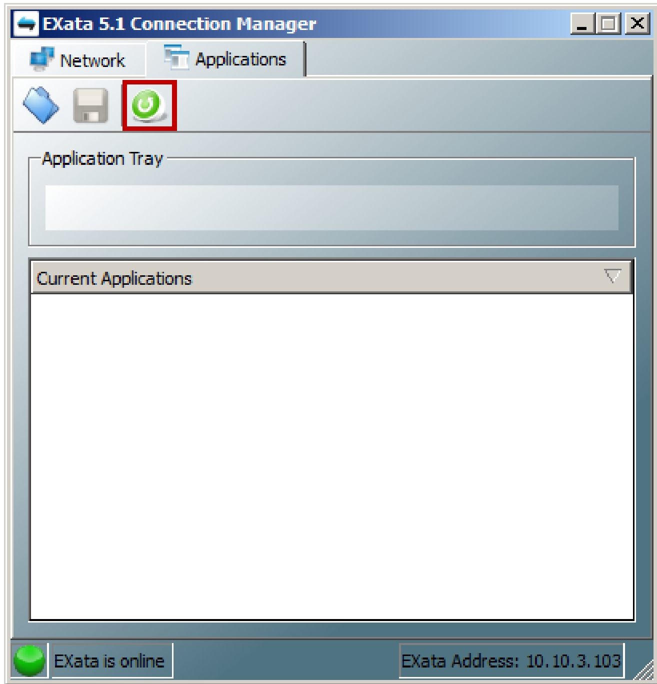


FIGURE 5-31. Restore Application Session Button

5.2.1.1.2.3 Saving and Loading Application Sessions

You can also save an application session in a file while a scenario is running and load it at a later time.

To save an application session while a scenario is running, do the following:

1. In the **Applications** window of Connection Manager, click on the **Save** button in the toolbar.
2. Provide the filename and the directory path where to save the application session information.

To load a previously saved application session, do the following:

1. Ensure that the EXata is running in Emulation mode.
2. In the **Applications** window of Connection Manager, click on the **Load** button in the toolbar.
3. In the file browser, select the name of the application session file to load.

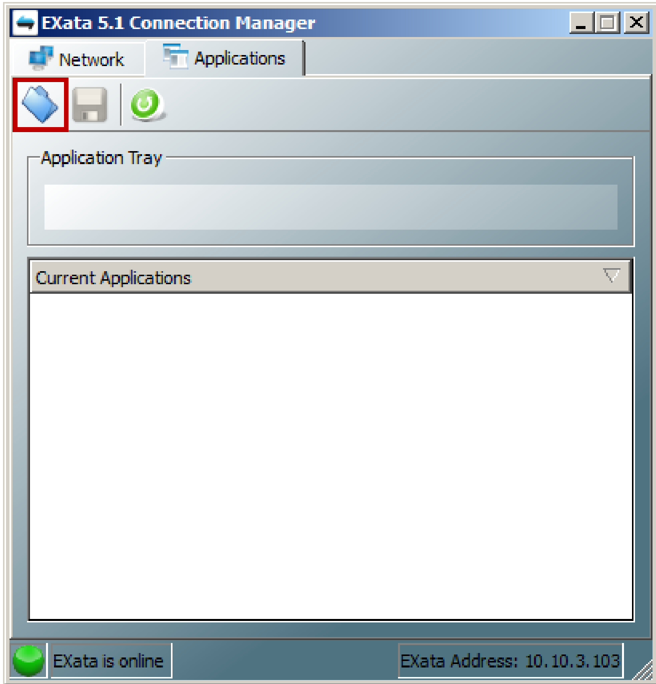


FIGURE 5-32. Load Application Session Button

5.2.1.1.2.4 Quick Launch Application Tray

The **Applications** tab in Connection Manager also features a quick launch application tray to quickly launch the frequently used applications.

To add an application to the quick launch tray, drag and drop the application icon from the desktop into the tray area. Figure 5-33 shows an example with applications loaded in the quick launch tray.

To launch an application, click on the icon in the tray. This will open a dialog box (see Figure 5-27) to select an emulated node on which to run the application.

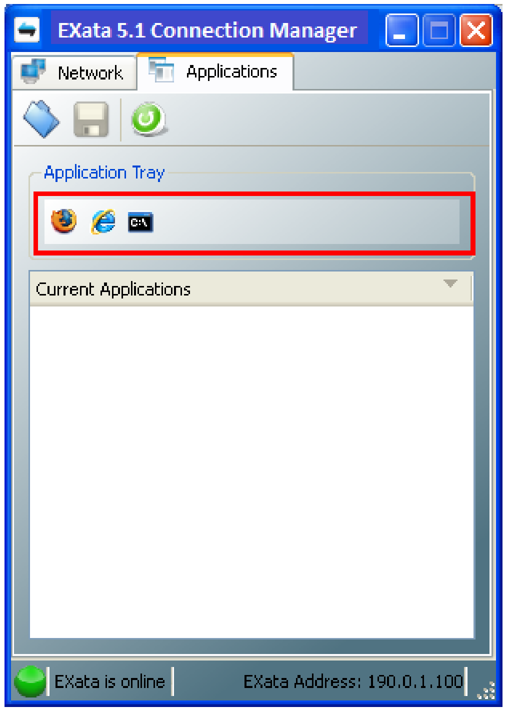


FIGURE 5-33. Quick Launch Application Tray

5.2.1.2 Launching Applications without Using Connection Manager

The operational hosts that do not run Connection Manager can be configured to connect with the emulated nodes with the mapping editor in the GUI or through the command line interface.

See Section 4.3.1 for details on how to connect to operational hosts to emulated nodes.

Note: This mode sets up a default emulated node for an operational host. All applications running on the operational host are run on the same default emulated node. It is not possible to select different emulated nodes for different applications in this mode.

5.2.2 EXata Emulated Server Applications

This section describes how to use the emulated server applications implemented in EXata.

The emulated server applications run on all emulated nodes in the scenario. A client application running on an operational host can access the corresponding emulated server application in exactly the same way as it would access a server application running on a physical machine.

EXata implements the following server applications:

- FTP Server
- HTTP Server
- TELNET Server

All three applications can be run in the default EXata node mode (see Section 5.2.1.1.1) or the manual emulated node selection mode (see Section 5.2.1.1.2).

Emulated FTP Server

To access the emulated FTP server application from an operational host, do the following:

1. Start a command window application to run on the emulated node.
2. In the command window, type the following command:

   `ftp <server-ID>`

   where

`<server-ID>`	IP address or host name of the emulated server node

3. Provide the following authentication details in the FTP client:

   Username: `anonymous`

   Password: `(leave blank)`

   or

   Username: `(leave blank)`

   Password: `(leave blank)`

You can now list and download files from the emulated FTP server. (Note that you can not upload files to the emulated FTP server.) The following FTP commands are supported by the emulated FTP server:

ABOR	PASV	STOR
AUTH	PORT	SYST
LIST	PWD	TYPE
NLST	QUIT	USER
PASS	RETR	XPWD

Emulated HTTP Server

To access the emulated HTTP server application from am operational host, do the following:

1. Start a web browser to run on the emulated node.
2. Enter the IP address or the host name of emulated server node in the address bar.

   Note: If an IPv6 address is used to access the emulated server node, then the address must be enclosed between `[` and `]`.

Emulated TELNET Server

To access the emulated TELNET server application from an operational host, do the following:

1. Start a command window application to run on the emulated node.
2. In the command window, type the following command:

```
telnet <server-ID>
```

where

`<server-ID>`	IP address or host name of the emulated server node

The EXata emulated server provides a limited set of commands that can be accessed by the TELNET client. Type **`help`** in the TELNET client window to show a list of supported commands.

5.2.3 Configuration for Multicast Applications

When a unicast application is run from an operational host, packets are routed to the destination node via the emulated network running on the emulation server. The destination node does not receive packets directly from the source node even if they are both on the same subnet.

For a multicast application, however, application packets can be directly received by destination nodes if they are on the same subnet as the source node. To make all multicast application traffic go through the emulated network, direct routing of application traffic between source and destination nodes must be disabled. This is achieved by setting up firewalls on all operational hosts. The firewall on each operational host is configured to block traffic from all other operational hosts.

To configure a testbed for a multicast application, do the following:

1. On each operational host running Windows, configure a firewall to accept multicast application packets only from the emulation server, as described in Section 5.2.3.1.
2. On each operational host running Linux, configure a firewall to block all traffic from each of the other operational hosts, as described in Section 5.2.3.2.

Configuring the firewalls on all operational hosts in this manner ensures that only those multicast application packets will be received at a destination that are sent by the emulation server (and, therefore, have traversed the emulated network).

To configure a firewall on a Windows machine, the IP address of the emulation server is needed. To configure a firewall on a Linux machine, the hardware (MAC) addresses of the other operational hosts are needed. Section 5.2.3.3 describes how to identify the IP and hardware addresses on Windows and Linux machines.

5.2.3.1 Configuring Firewalls on Windows

To configure a firewall on a Windows 7 or Windows 8 platform that accepts multicast application packets only from the emulation server, perform the following steps:

1. Select **Start > Control Panel > Windows Firewall**. Click on **Advanced settings**.

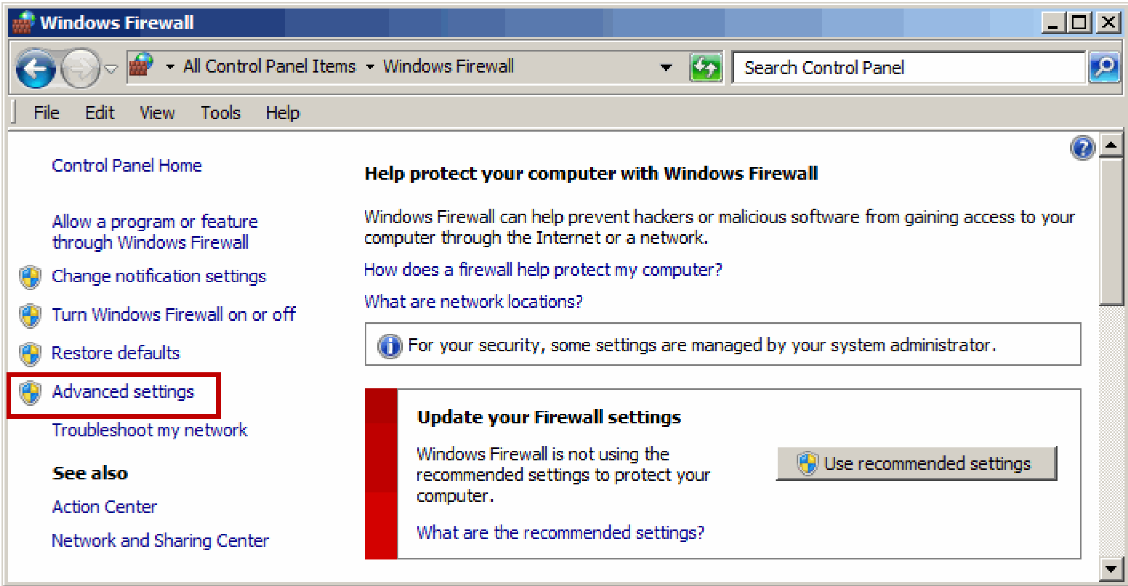


FIGURE 5-34. Windows Firewall Configuration

2. Under **Getting Started**, click on **Inbound Rules**.

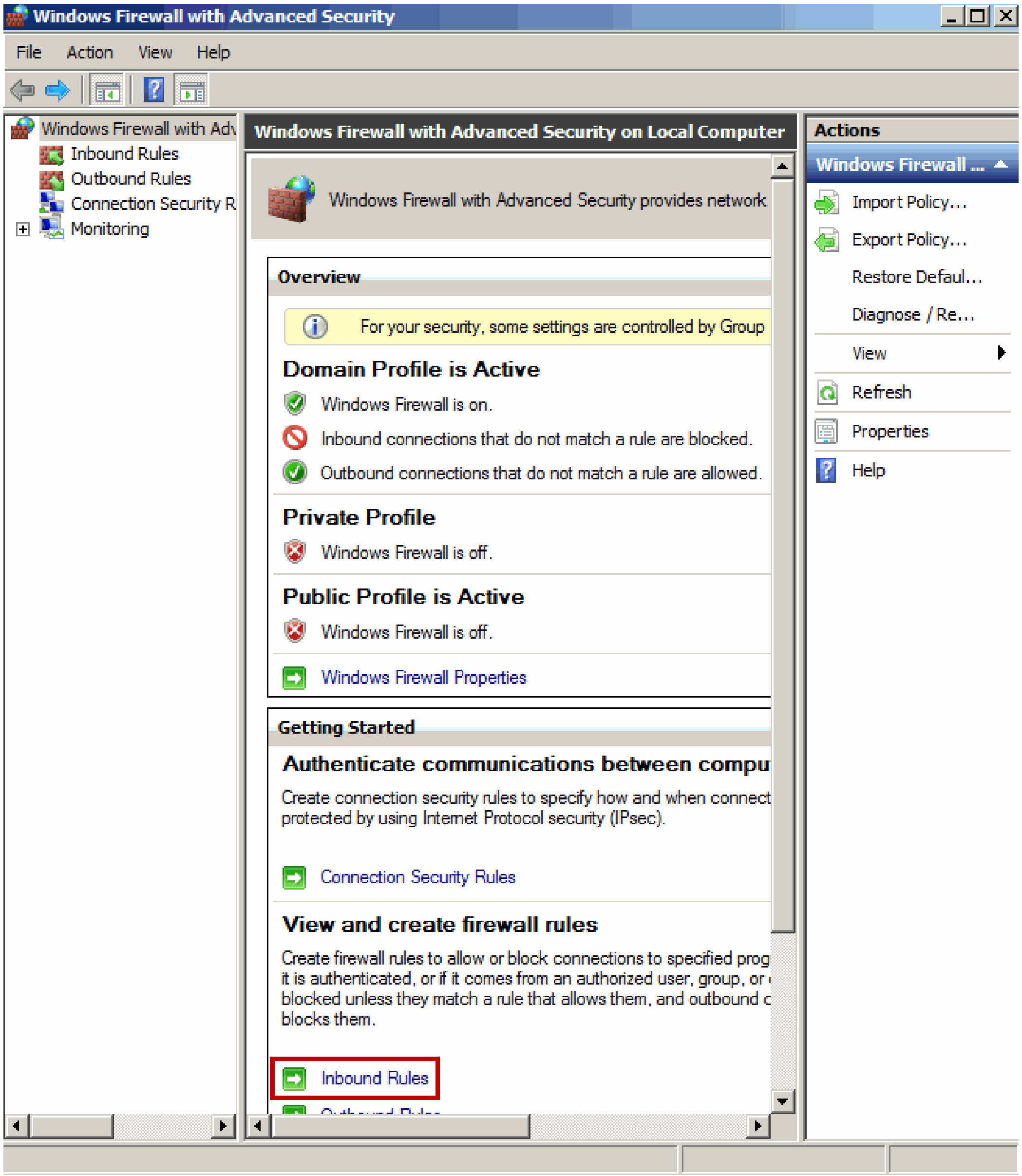


FIGURE 5-35. Selecting Inbound Rules

3. Launch the **New Rule Wizard** by clicking on **New Rule**.

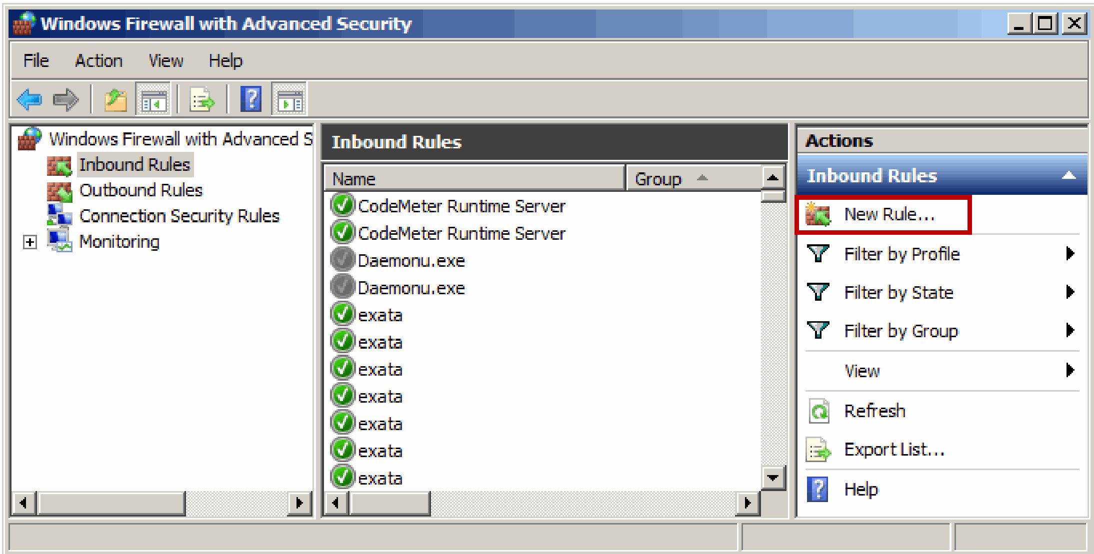


FIGURE 5-36. Launching the New Rule Wizard

4. In the **New Rule Wizard**, select **Program** and click **Next**.

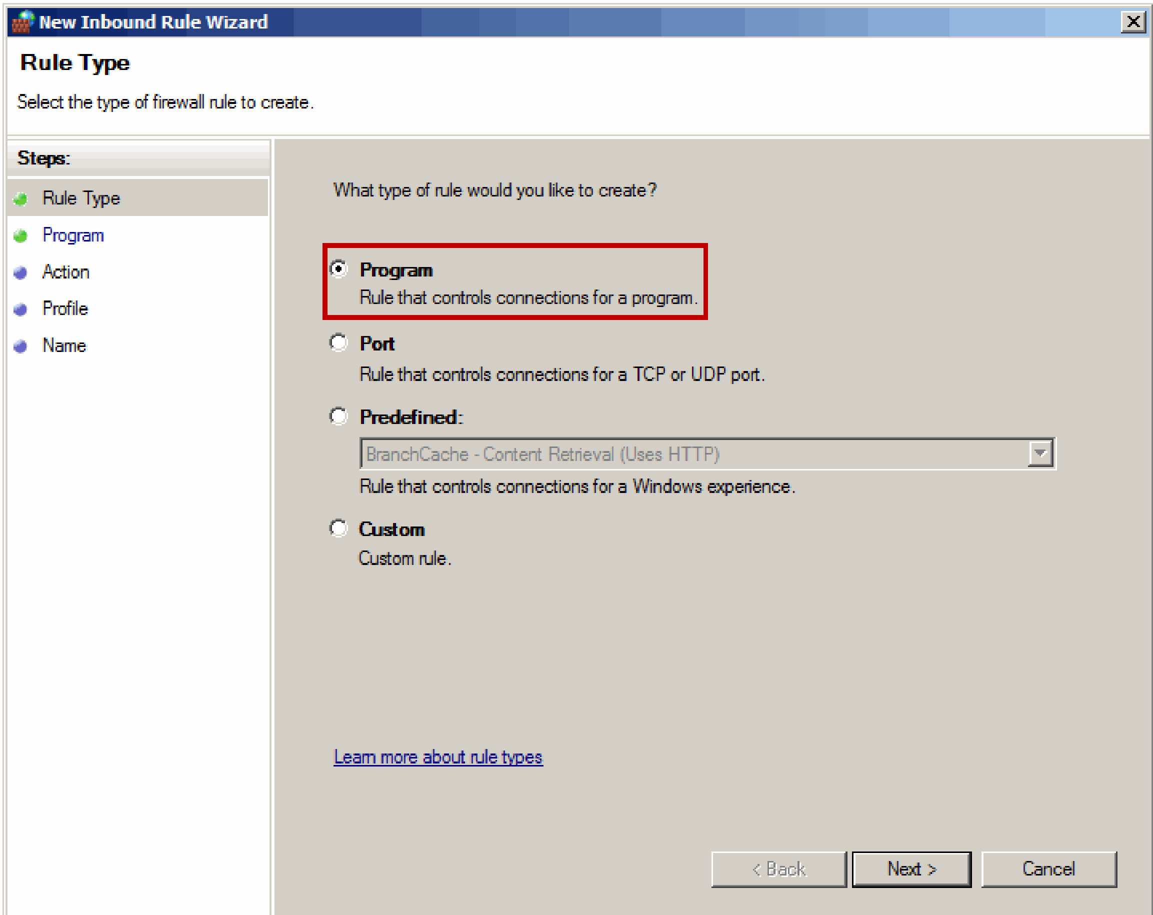


FIGURE 5-37. New Rule Wizard

5. Select **This program path**, browse to the multicast application, and press **Next**.

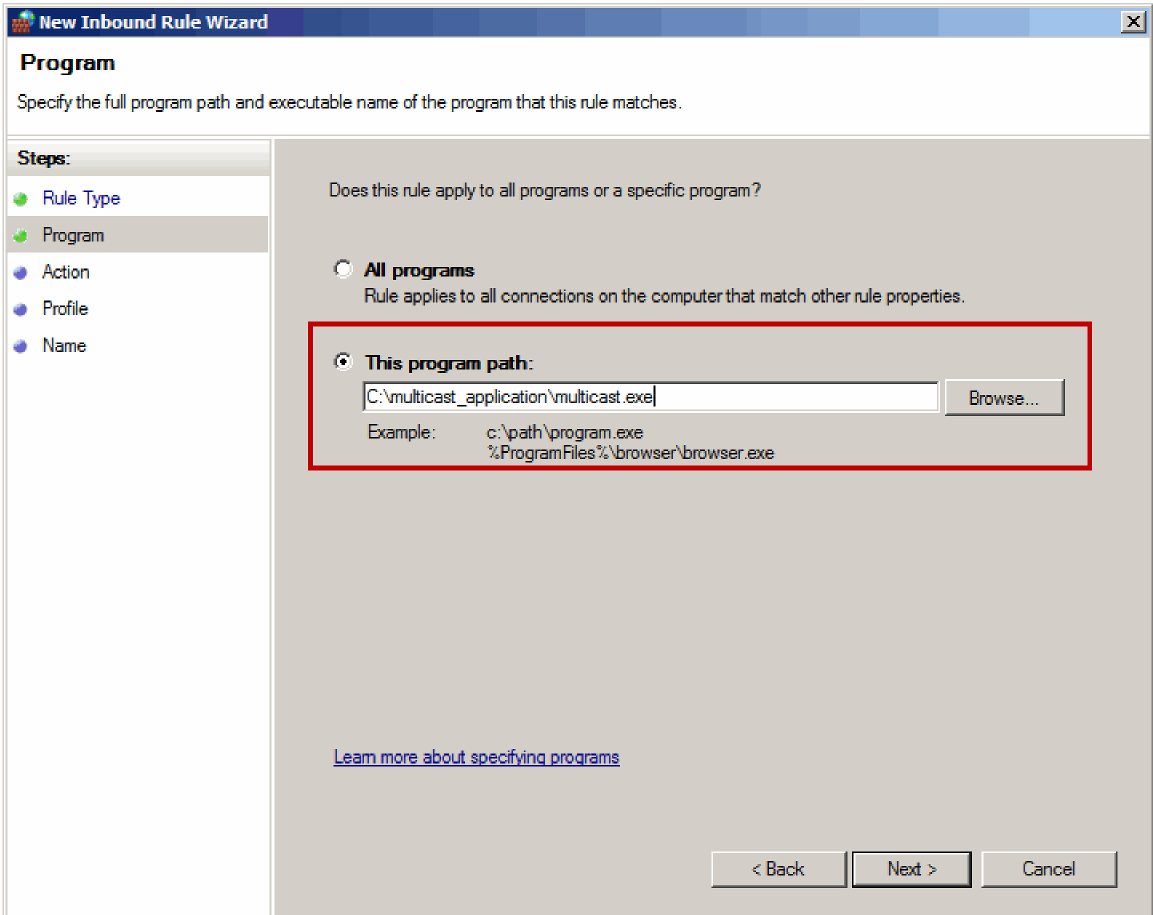


FIGURE 5-38. Specify Multicast Application

6. In the next window, select **Allow the connection** and click **Next**.

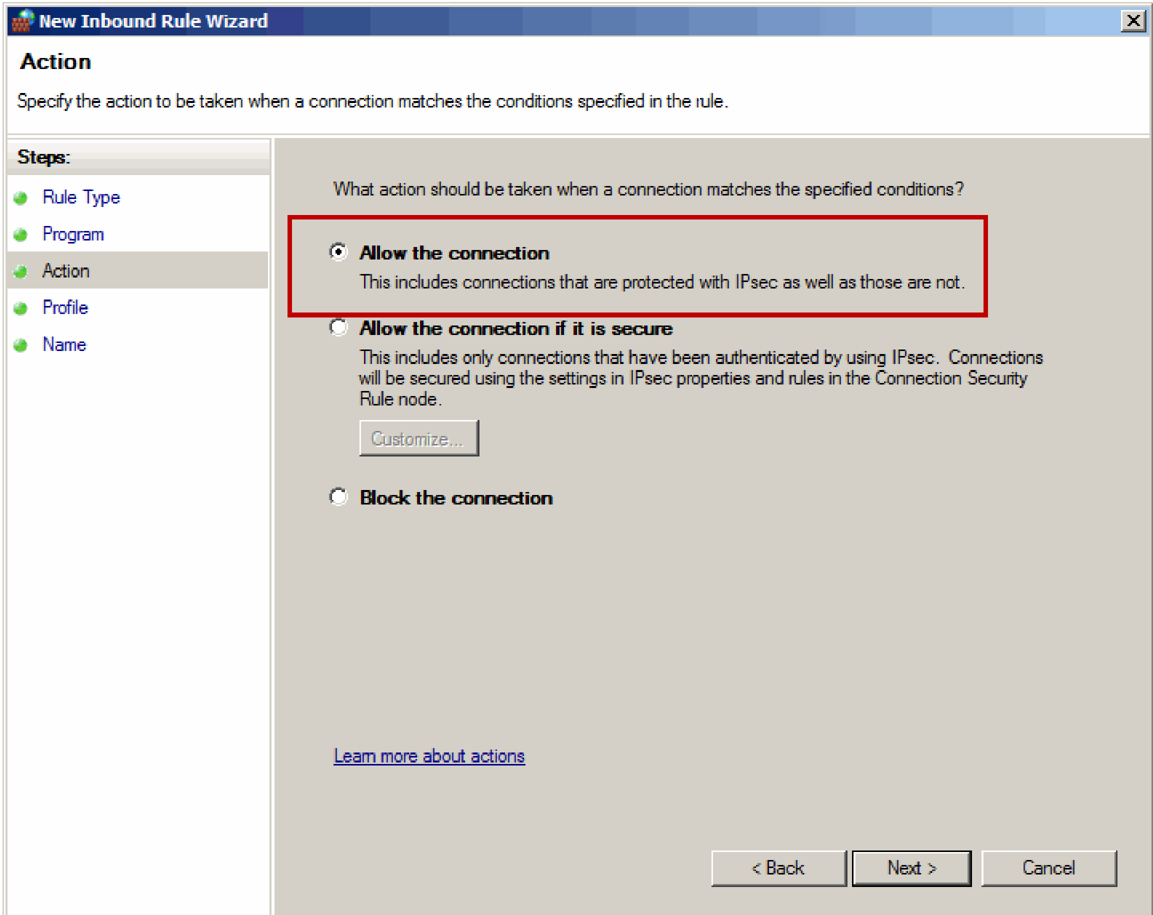


FIGURE 5-39. Select Action for Rule

7. In the next window, select the profiles and click **Next**.

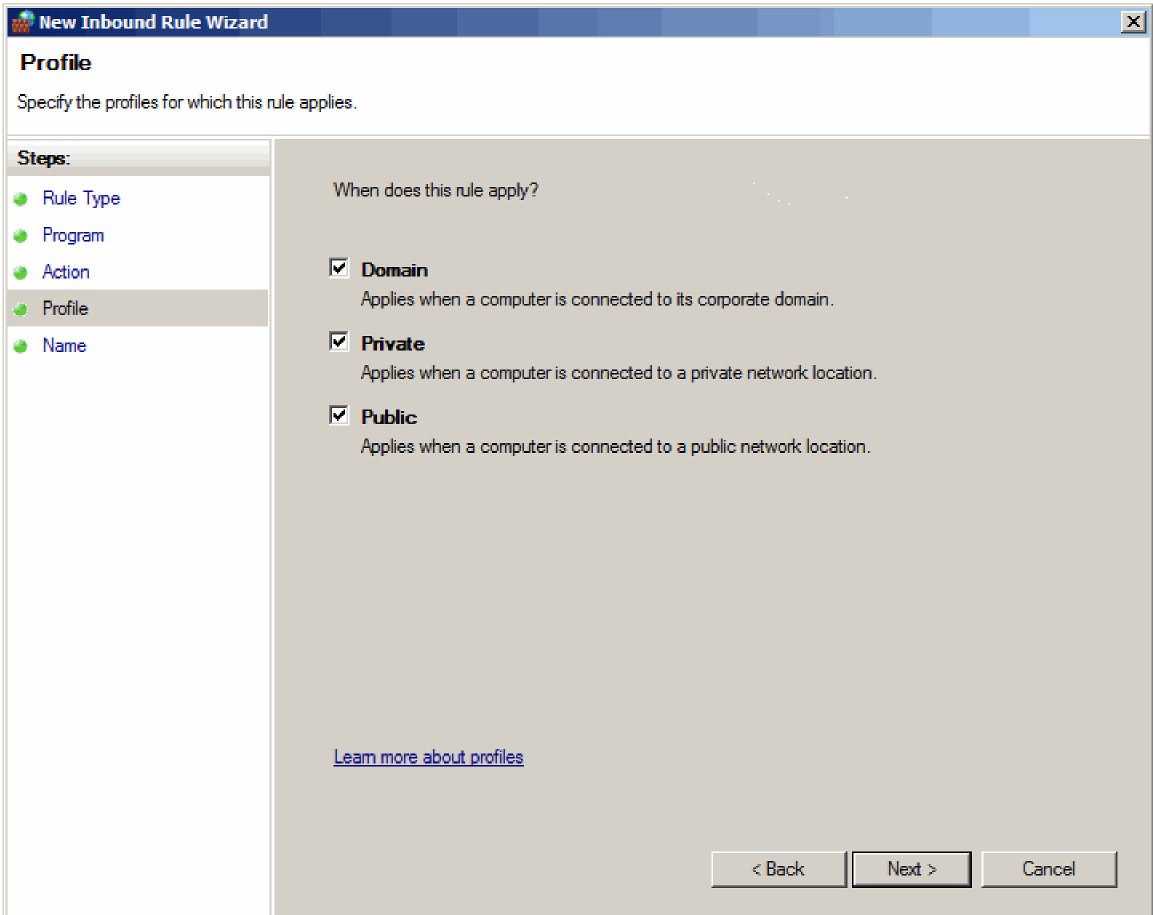


FIGURE 5-40. Select Profiles for Rule

8. In the next window, enter the name and description of the rule and click **Finish**.

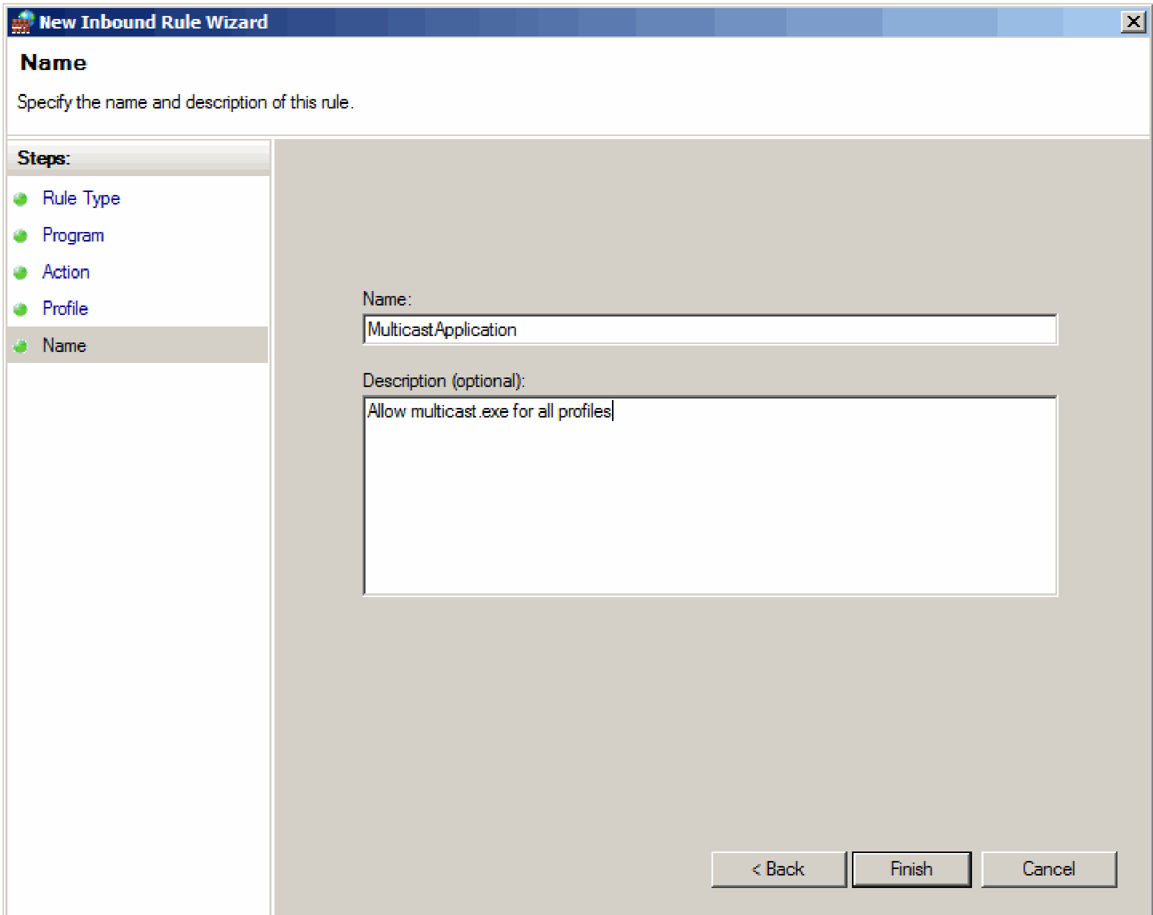


FIGURE 5-41. Enter Name of Rule

9. In the **Windows Firewall and Advanced Security Window**, select **Inbound Rules** in the left panel. Right click on the newly created rule and select **Properties**.

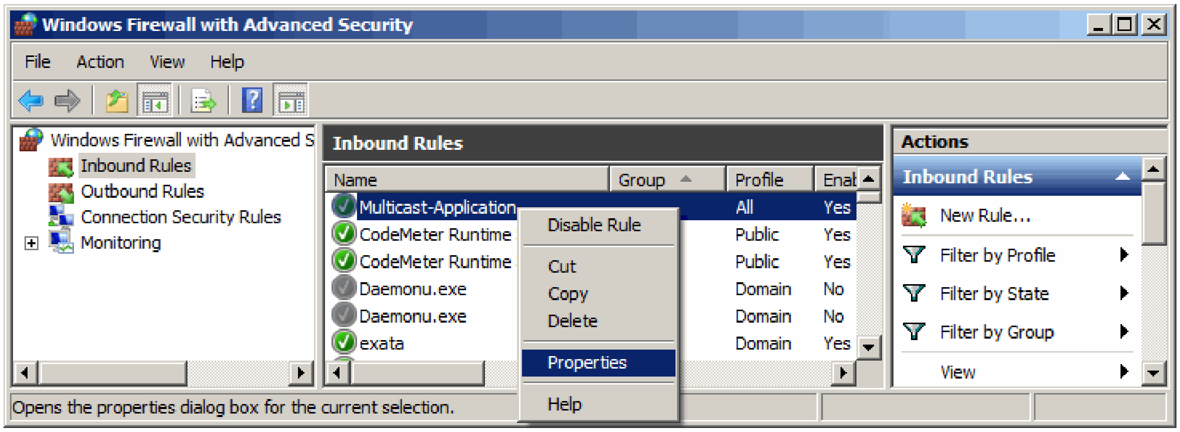


FIGURE 5-42. Set Properties of New Rule

10. In the properties wizard for the new rule, select the **Scope** tab. Under **Remote IP address**, select **These IP addresses** and click **Add**.

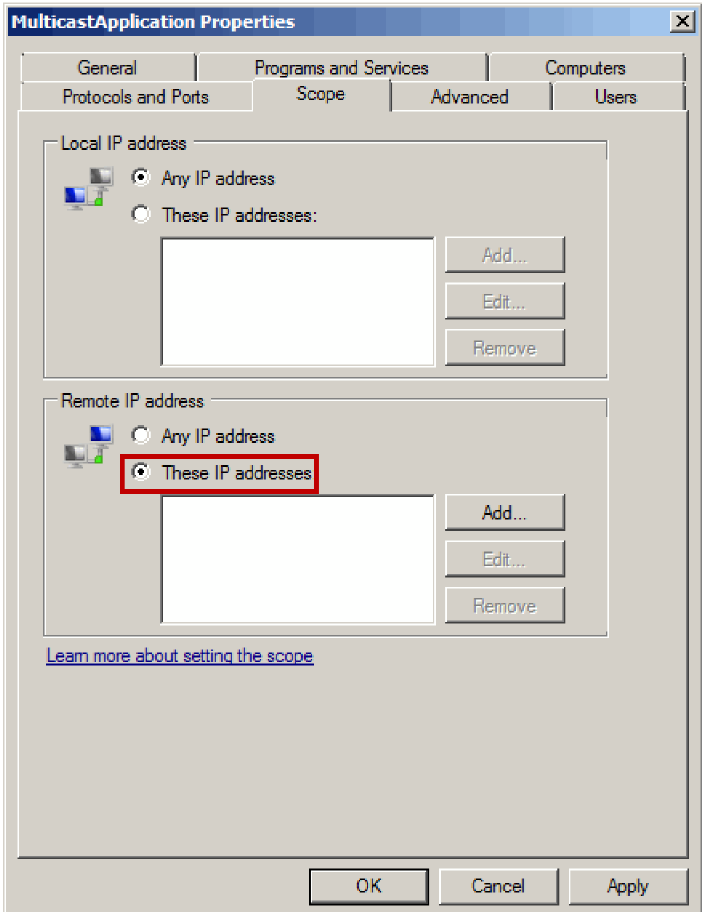


FIGURE 5-43. Specifying IP Addresses for Rule

11. In the **IP Address** dialog, select **This IP address or subnet**. Enter the IP address of the emulation server click **OK**. (See Section 5.2.3.3 for the procedure to find the IP address of a Windows or Linux machine.)

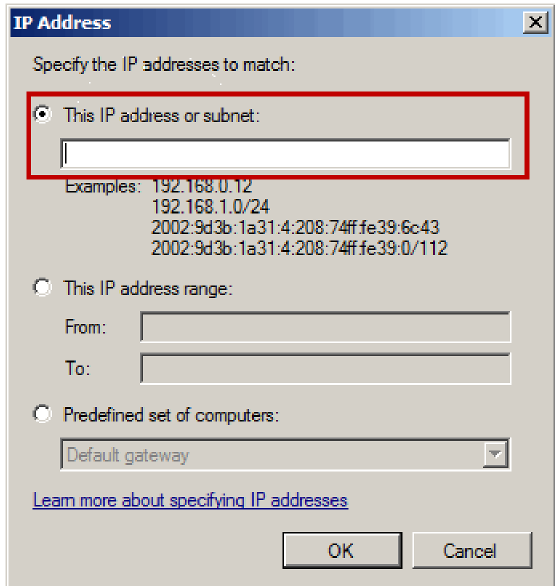


FIGURE 5-44. Specifying IP Address of Emulation Server

5.2.3.2 Configuring Firewalls on Linux

To configure a firewall on an operational host that blocks packets from the other operational hosts, perform the following steps:

1. Open a command window on the operational host where you are configuring the firewall.
2. If you are logged in as root, type the following command:

```
iptables -A INPUT -m mac --mac-source <HW-Address> -j DROP
```

where

`<HW-Address>`	Hardware address of the network interface card used to connect **the other operational host** to the emulation server. See Section 5.2.3.3.1 for the procedure to find the hardware address of a Windows machine. See Section 5.2.3.3.2 for the procedure to find the hardware address of a Linux machine.

If you are not logged in as root, use the following command:

```
sudo iptables -A INPUT -m mac --mac-source <HW-Address> -j DROP
```

3. Repeat the previous step for each of the other operational hosts.

5.2.3.3 Identifying IP and Hardware Addresses

This section describes how to identify the IP address and hardware (MAC) address for Windows machines (see Section 5.2.3.3.1) and Linux machines (see Section 5.2.3.3.2).

5.2.3.3.1 Identifying IP and Hardware Addresses for a Windows Machine

To identify the IP address and hardware (MAC) address of a the network interface card used to connect the operational host to the emulation server, perform the following steps:

1. Open a command window on the operational host.
2. Type the following command and press enter:

```
ipconfig /all
```

3. Information about all interfaces on the machine is displayed in the command window. Locate the network interface that is used to connect to the emulation server.

   As an example, the following is the information displayed for an ethernet interface:

```
Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : snt.loc
   Description . . . . . . . . . . . : Intel(R) 82579LM Gigabit ...
   Physical Address. . . . . . . . . : AB-CD-EF-GH-IJ-KL
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : #::#:#:#:#
   Link-local IPv6 Address . . . . . : #::#:#:#:#
   IPv4 Address. . . . . . . . . . . : #.#.#.#
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Tuesday, ...
   Lease Expires . . . . . . . . . . : Wednesday, ...
   Default Gateway . . . . . . . . . : #.#.#.#
   DHCP Server . . . . . . . . . . . : #.#.#.#
   DHCPv6 IAID . . . . . . . . . . . : 123456789
   DHCPv6 Client DUID. . . . . . . . : AB-CD-EF-GH-IJ...
   DNS Servers . . . . . . . . . . . : #.#.#.#
   Primary WINS Server . . . . . . . : #.#.#.#
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

4. The hardware address of an interface is displayed after `Physical Address`. The IPv4 address is displayed after `IPv4 Address`. The IPv6 address is displayed after `IPv6 Address`.

5.2.3.3.2 Identifying IP and Hardware Addresses for a Linux Machine

To identify the IP address and hardware (MAC) address of a the network interface card used to connect the operational host to the emulation server, perform the following steps:

1. Open a command window on the operational host.
2. Type the following command and press enter:

```
ifconfig
```

Note: If the following error message is displayed in the command window:

```
'ifconfig: command not found',
```

then use the following command:

```
/sbin/ifconfig
```

If the problem still persists, contact your system administrator for help with running the ifconfig command.

3. Information about all interfaces on the machine is displayed in the command window. Locate the name of the network interface that is used to connect to the emulation server.

As an example, the following is the information displayed for an ethernet interface **eth0**:

```
eth0      Link encap:Ethernet  HWaddr AB-CD-EF-GH-IJ-KL
          inet addr:#.#.#.#  Bcast:#.#.#.#  Mask:255.255.255.0
          inet6 addr:#::#:#:#:#  Scope:...
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:703762 errors:0 dropped:0 overruns:0 frame:0
          TX packets:364863 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:560600001 (534.6 MiB)
          TX bytes:137121730 (130.7 MiB)
          Interrupt:11 Base address:0x1400
```

4. The hardware address of an interface is displayed after **HWaddr** in the first line of the information displayed for the interface. The IPv4 address is displayed after **inet addr** on the second line. The IPv6 address is displayed after **inet6 addr** on the third line.

5.3 Connecting to Routers

Just as an emulated node in EXata can be mapped to an operational host running an application (as a External Node or an External Application Node), an emulated node can also be mapped to an operational host running a routing protocol. The operational host can be a router device, such as a Cisco router, or can be a computer implementing the full protocol stack. The emulated node that such an operational host is mapped to is called an *External Router Node*. This mapping allows the routing protocol running on the operational host to interact and share routing information with the routing protocol model running on the emulated nodes within EXata.

Note: An emulated node can not be mapped to an operational host running a routing protocol if the operational host is running IPv6.

EXata interoperates with the following routing protocols:

- Border Gateway Protocol v4 (BGPv4)
- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP), both with and without the Proxy mode
- Open Shortest Path First (OSPF version 2)
- Optimized Link State Routing (OLSR)
- Protocol Independent Multicast Protocol: Dense Mode (PIM-DM)
- Protocol Independent Multicast Protocol: Sparse Mode (PIM-SM)

EXata models of these routing protocols are high-fidelity and follow the industry standards and will interoperate with the routing protocols running on operational hosts as long as their implementation also follow the industry standards.

As an example, consider the configuration shown in Figure 5-45. The emulated scenario comprises six nodes. Node 1 is mapped to Operational Host 1 which is running an off-the-shelf implementation of OSPF. Node 2 is mapped to Operational Host 2 which is not running any routing protocol.

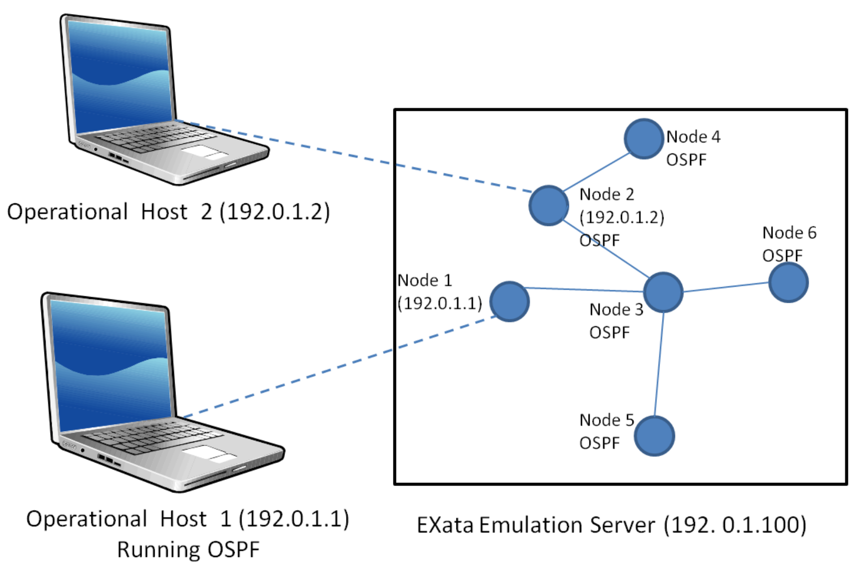


FIGURE 5-45. Mapping Routers to Emulated Nodes

OSPF packets sent by the Operational Host 1 are captured by the Emulation Server at node 1 and are forwarded to the emulated nodes in the scenario. Emulated nodes running OSPF update their routing tables upon receiving these packets. In turn, the OSPF control packets generated by the emulated nodes are also received by node 1. Node 1 is configured to not run a routing protocol and the OSPF control packets received by node 1 are forwarded to Operational Host 1.

To establish a connection between an operational host running a routing protocol and an emulated node (External Router Node), perform the following steps:

1. Connect the emulation server and operational host, as described in Section 5.1.
2. Create a mapping between the operational host and the External Router Node in the same manner as creating a mapping between an operational host and an External Node (see Section 5.2).

The following conditions should be met for the routing protocol running at the operational host and the routing protocol model running at the External Router Node to interoperate:

- The routing protocol should be OLSR, OSPFv2, or RIPv2
- The External Router Node should have only one interface
- The External Router Node should be configured to not run any routing protocol.
  - For configuration for the command line interface, the parameter `ROUTING-PROTOCOL` should be set to `NONE` for the interface of the External Router Node in the scenario configuration (.config) file. See Section 4.2.8.3.2.1 for details.
  - For configuration using the GUI, the parameter **Routing Protocol IPv4** should be set to *None* in the Interface Properties Editor for the interface of the External Router Node. See Section 4.2.8.3.2.2 for details.

To verify the interoperability, check the routing table at the operational host. For link state protocols such as OSPF that aggregate the entire topology information, the routing table will display all the nodes in the emulated scenario.

5.4 Connecting to the Internet

EXata allows Internet-based applications running on operational hosts to connect to the Internet via the emulated network. This allows applications like instant messengers, streaming video, VoIP, and others to run over EXata just like in the real world. This is achieved by designating one node in the emulated network as an *Internet Gateway Node* that serves as a boundary node between the emulated nodes within EXata and the Internet. Any traffic, originating from within the EXata scenario or from an operational host, destined to the Internet is routed to this gateway node and from there routed to the Internet. Similarly, any response from the Internet is received by the Internet Gateway node and subsequently forwarded to the intended emulated node.

Figure 5-46 illustrates the EXata Internet Gateway feature. On the emulation server, the EXata process is executing a scenario that models an emulated network. One of the nodes is connected with an operational host that is running an Internet-based application. A different node in the emulated network is designated as the Internet Gateway node that manages all the traffic flows between the emulated network and the Internet. EXata transparently manages the necessary Network Address Translation (NAT).

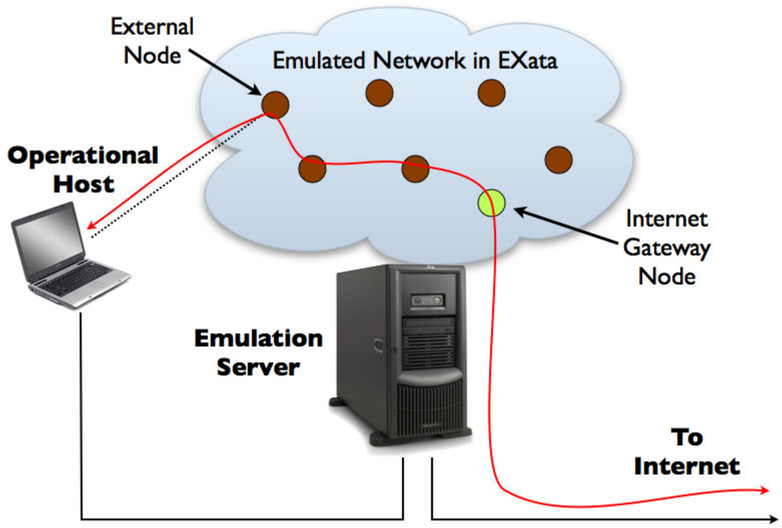


FIGURE 5-46. Illustration of the Internet Gateway

To configure an EXata Internet Gateway, the following conditions should be true:

- The emulation server should be connected to the Internet. This is can be verified by the opening a web browser and pointing it to any website. If the emulation server cannot reach the Internet, consult your network administrator.
- The network interface card on the emulation server that is used to connect to the Internet should not be shared with any operational host. Therefore, at least two network interface cards are required on the emulation server: one to connect to the Internet, and the remaining to connect to the operational hosts.
- The operational host should not be running any application (such as a routing protocol) that modifies the routing table on the operational host.

Section 5.4.1 describes how to configure the Internet Gateway in Windows. Section 5.4.2 describes how to configure the Internet Gateway in Linux. Section 5.4.3 describes how to verify the configuration.

5.4.1 Internet Gateway Configuration for Windows

To configure the EXata Internet Gateway on a Windows 7 or Windows 8 platform, perform the following steps:

1. Designate one node in the emulated network as the Internet Gateway Node. For configuring a gateway to an IPv4 network, the Internet Gateway Node must run IPv4 or Dual IP. For configuring a gateway to an IPv6 network, the Internet Gateway Node must run IPv6 or Dual IP. See Section 4.3.2 for details of configuring the Internet Gateway for the command line interface and the GUI.

If EXata GUI is running, the selected node is highlighted on the canvas by an orange triangle (see node 15 in Figure 5-47).

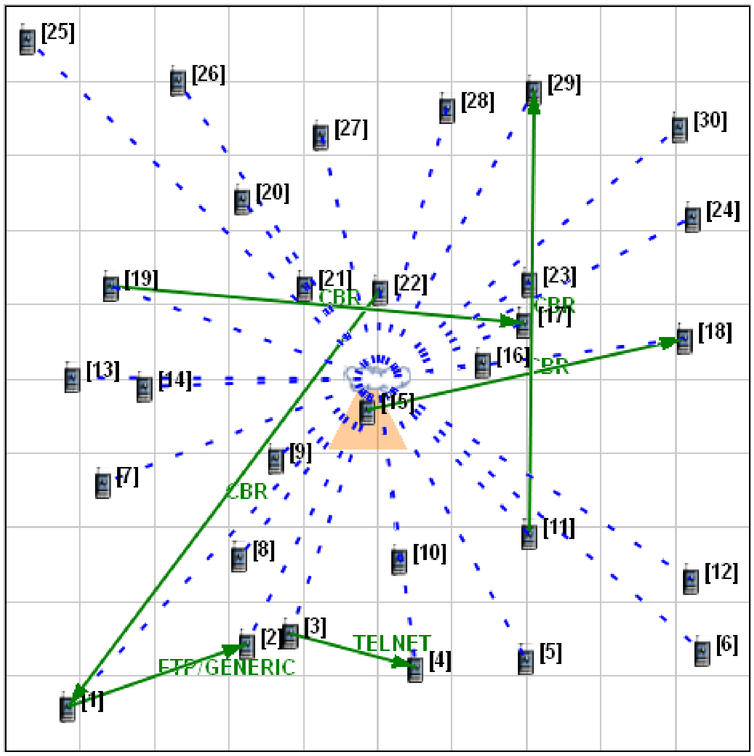


FIGURE 5-47. EXata GUI Displaying the Internet Gateway Node

2. Enable the firewall on the emulation server and configure it to drop any unsolicited packets from the outside. Configure the firewall to exempt the EXata process. To achieve this, do the following:

a. Select **Start > Control Panel > Windows Firewall**. Click on **Turn Windows Firewall on or off**.

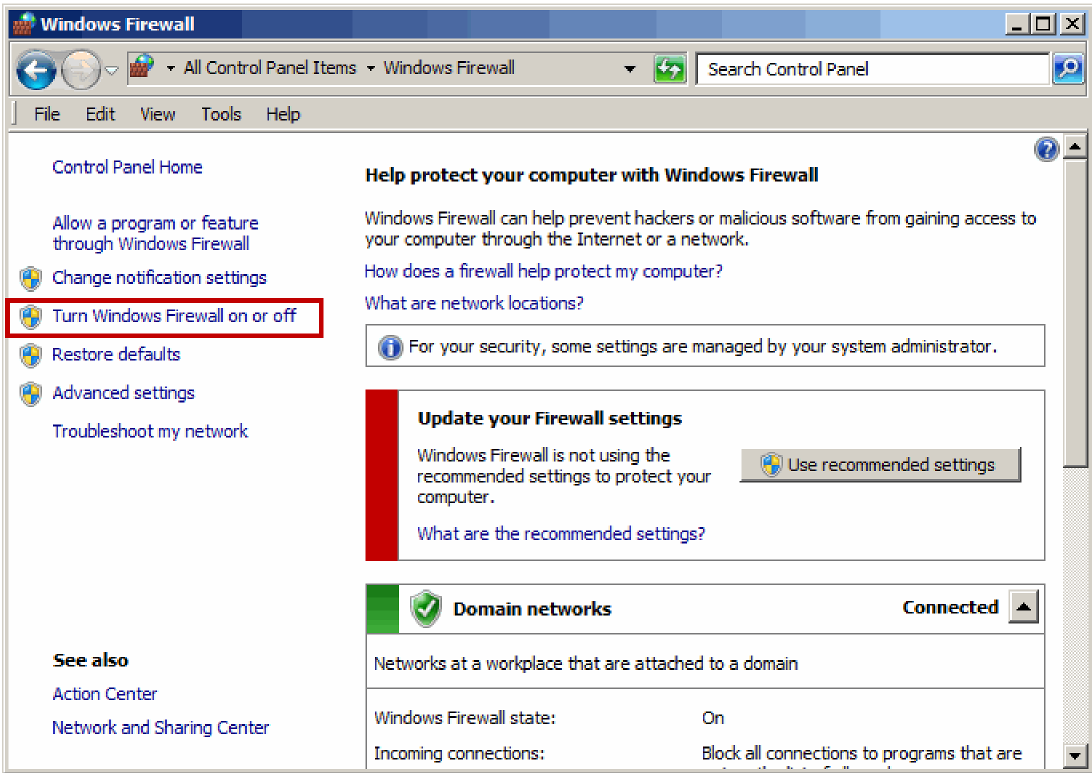


FIGURE 5-48. Controlling Windows Firewall

b. Turn on Windows firewall.

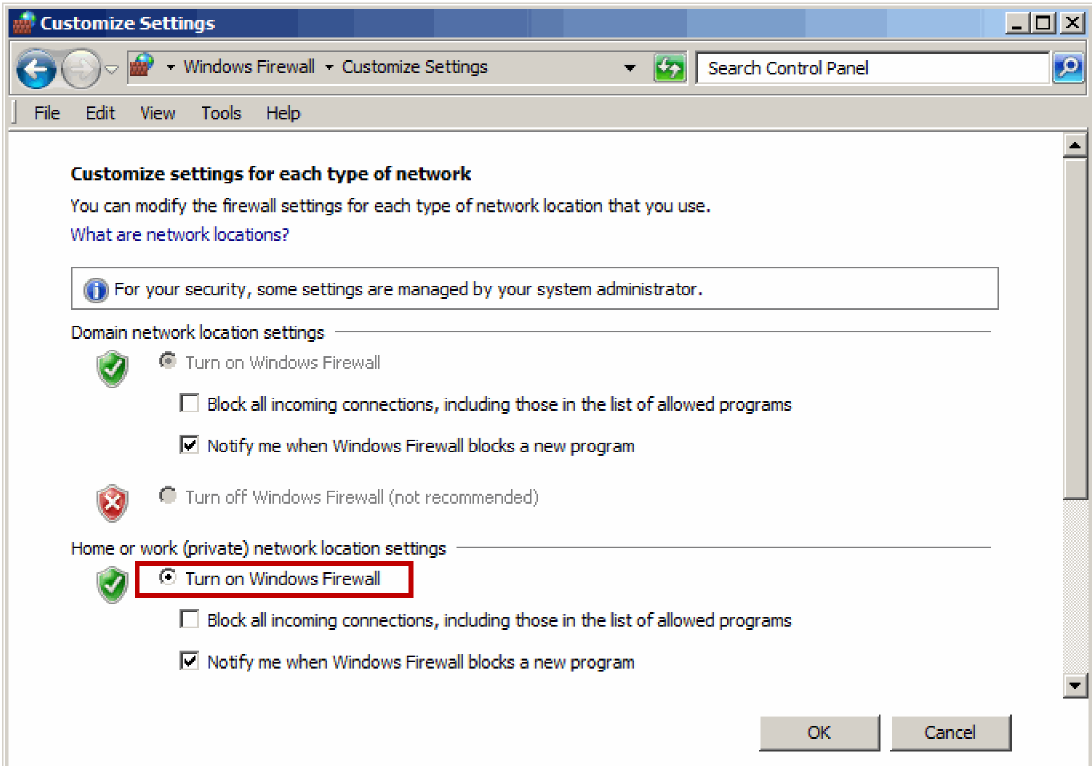


FIGURE 5-49. Turn Windows Firewall On

c. Navigate to the previous screen and calcite on **Allow a program or feature through Windows Firewall**.

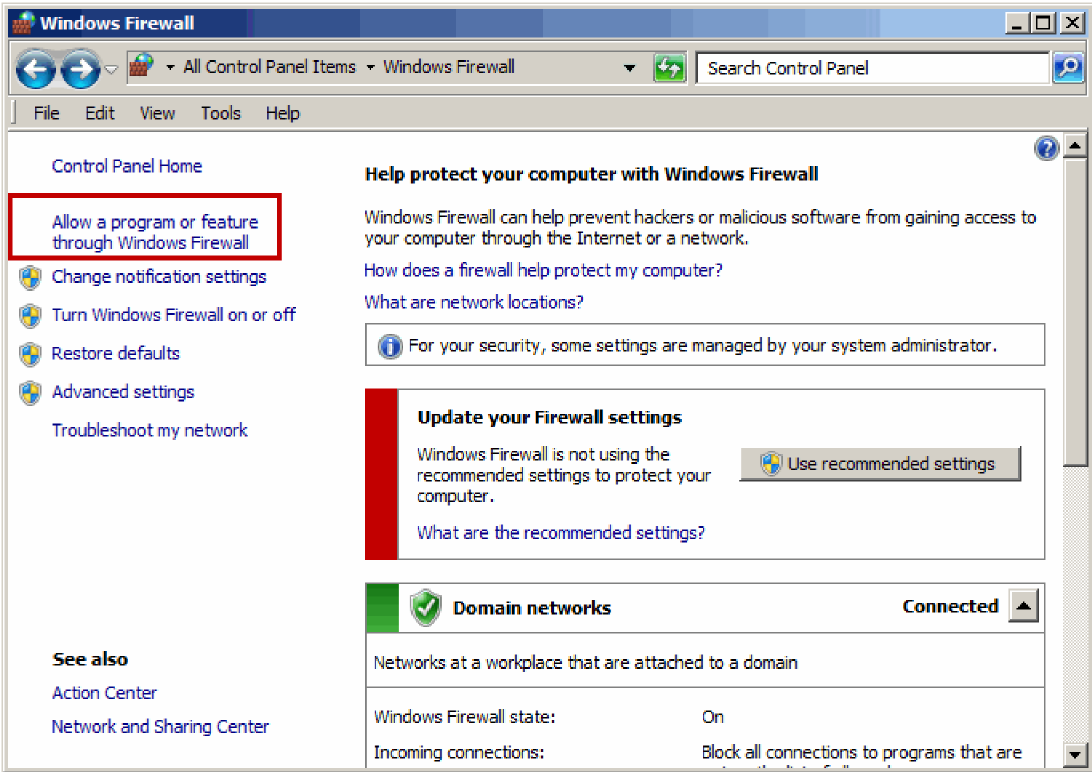


FIGURE 5-50. Allow a Feature through Firewall

d. In the next screen, examine the programs listed under **Allowed programs and features**. If exata.exe is listed, then select the access levels by checking the corresponding boxes and click **OK**.

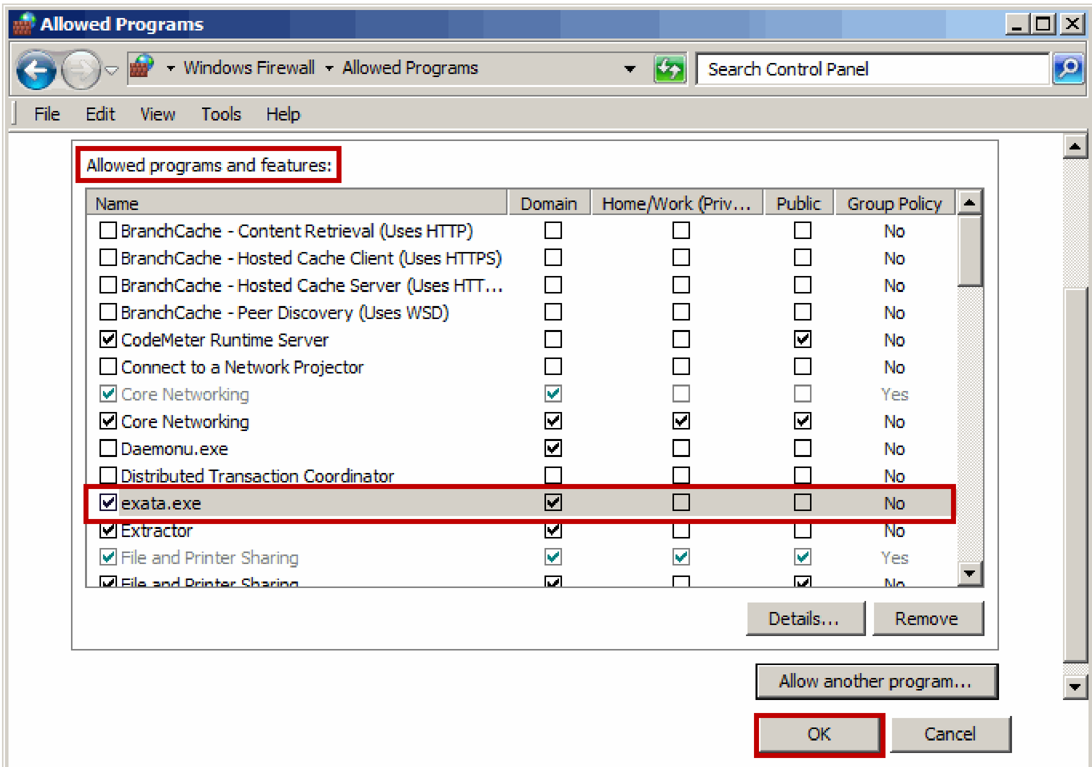


FIGURE 5-51. Allow EXata through Firewall

e. If exata.exe is not listed under **Allowed programs and features**, then click on **Allow another program**. In the next screen, select exata.exe (if it is listed under **Programs**) or browse to exata.exe and click on **Add**.

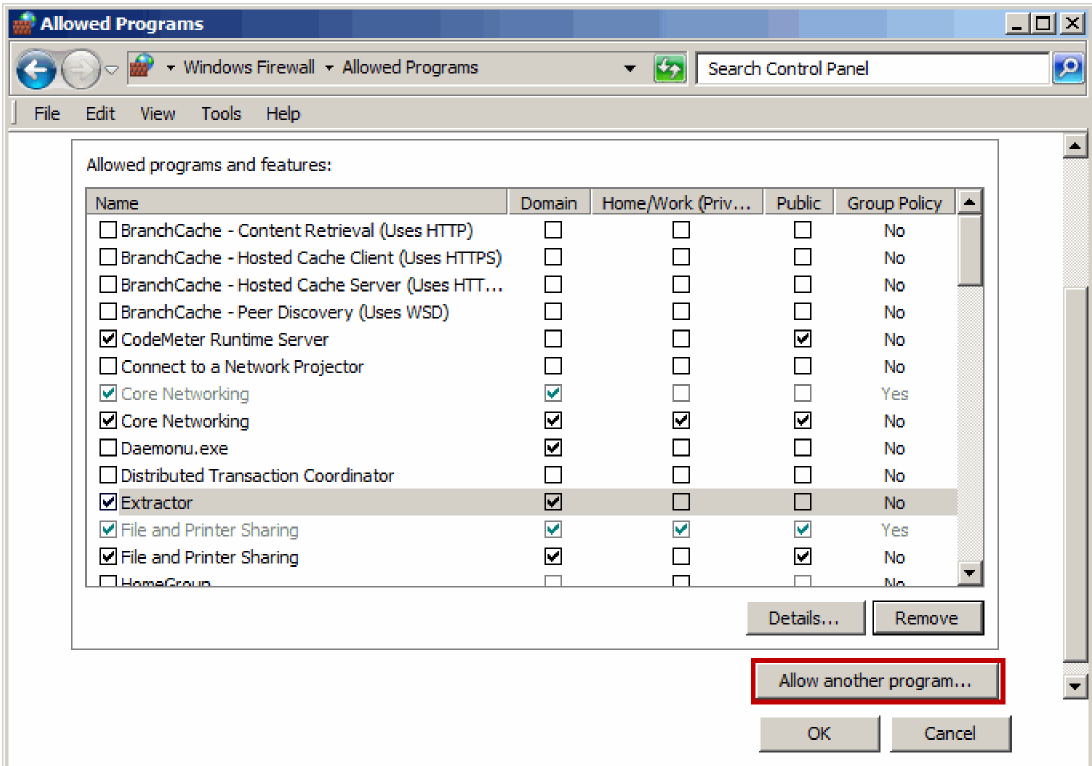


FIGURE 5-52. Allowing another Program through Firewall

Select the access levels by checking the corresponding boxes and click **OK**.

3. Determine the addresses of the Domain Name Servers (DNS) on the emulation server as follows:

a. Open a command window and type `ipconfig /all`.

b. Locate the network connection that is used to connect to the Internet and note the DNS entries. An example is shown in Figure 5-53.

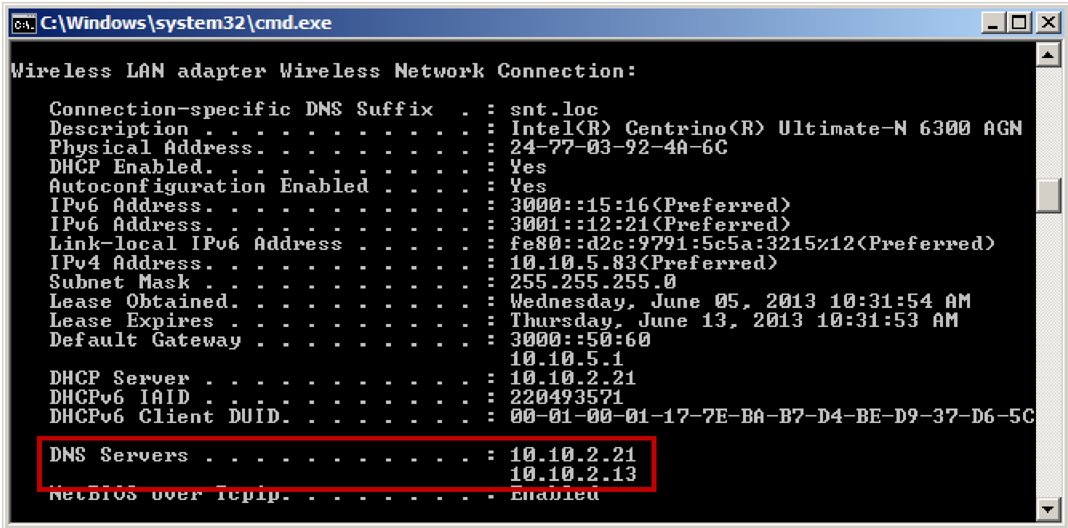


FIGURE 5-53. DNS Address in Windows

4. Apply the DNS address, obtained in the preceding step, on the operational host as follows:
   a. Open the network configuration editor from **Start > Control Panel > Network and Sharing Center > Change adapter settings**.
   b. Select the network interface card that is used to connect to the emulation server, right-click and select **Properties**.
   c. In the **Networking** tab, scroll down in the list to **Internet Protocol Version 4 (TCP/IPv4)** or **Internet Protocol Version 6 (TCP/IPv6)** and click the **Properties** button. Select **Use the following DNS server addresses**.
   d. Enter the DNS address obtained in previous step. An example using IPv4 addresses is shown in Figure 5-54.

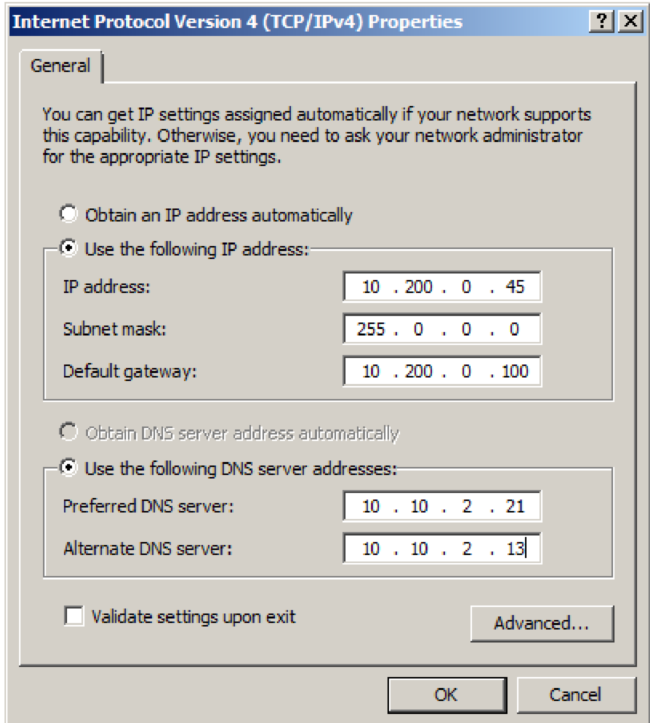


FIGURE 5-54. Setting the DNS Address in Windows Operational Host

Verify the configuration, as described in Section 5.4.3.

5.4.2 Internet Gateway Configuration for Linux

To configure the EXata Internet Gateway on Linux, perform the following steps:

1. Designate one node in the emulated network as the Internet Gateway Node. See Section 4.3.2 for details of configuring the Internet Gateway for the command line interface and the GUI.

   If EXata GUI is running, the selected node is highlighted on the canvas by an orange triangle (see node 15 in Figure 5-47).

2. Verify that the **iptables** program is installed. Open a terminal window and type:

   ```
   whereis iptables
   ```

   Note: The above command will print the location of this program, if it is installed, otherwise it will report it as missing. If iptables is not available on your computer, consult your system administrator on how to install this program.

3. Determine the addresses of the Domain Name Servers (DNS) on the emulation server as follows:

   a. Open a terminal window and type **`cat /etc/resolv.conf`**.

   b. Note the IP addresses next to the string `namespace`.

4. Apply the DNS address, obtained in the preceding step, on the operational host as follows:

a. Open the */etc/resolv.conf* file in your preferred editor. Comment out any existing lines (by typing # at the beginning of the line). Enter new lines in the following format:

```
nameserver <dns-address>
```

where `<dns-address>` is the DNS-address obtained in the preceding step. Enter each DNS address on a separate line.

Save and close the file.

b. Open a terminal window and type:

```
sudo service network restart
```

If the above command fails, type:

```
/etc/init.d/networking restart
```

Note: If you are logged in as *root* in the terminal window, you can omit the **`sudo`** keyword.

Verify the configuration, as described in Section 5.4.3.

5.4.3 Verifying Internet Gateway Configuration

To verify that the Internet Gateway is correctly configured, perform the following steps:

1. Verify that the Internet Gateway is reachable from the external node (mapped to the operational host). To do this, from the operational host, start a command window application on the external node and type

`ping <IPv4-address>` (on Windows or Linux)

or

`ping6 <IPv6-address>` (on Windows)

or

`ping -6 <IPv6-address>` (on Linux)

where

`<IPv4-address>`	IPv4 address of the Internet Gateway node assigned by EXata
`<IPv6-address>`	IPv6 address of the Internet Gateway node assigned by EXata

If there is no response to the ping command, then select a different emulated node as the external node that is reachable from the Internet Gateway.

2. From the operational host, start a web browser application to run on the external node. Enter the address of a web site in the browser. If the Internet Gateway configuration is correct, the specified web site will open in the browser. In addition, if EXata GUI is running, traffic (indicated by blue arrows) will flow between the external node and the Internet Gateway.

5.5 Simple Network Management Protocol (SNMP)

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.

To configure and use the SNMP model in EXata, refer to *Network Management Model Library*.

5.6 Using Packet Sniffing and Analysis Software

EXata supports sniffing of emulated network packets using popular third party software Wireshark (formerly Ethereal), Observer, or Microsoft Network Monitor. You can analyze the network packet by packet, protocol by protocol, and diagnose network problems and make use of the advanced capabilities such as statistics and reporting.

Packet analyzers can be used for:

- Analyzing network problems
- Protocol verification
- Monitoring network usage
- Gathering and reporting network statistics
- Filtering suspect content from network traffic
- Reverse engineering protocols used over the network
- Debugging client/server communications
- Debugging network protocol implementations
- Converting data to human readable format
- Fault analysis to discover problems in the network, such as lack of connectivity
- Performance analysis to discover network bottlenecks

Figure 5-55 shows the EXata Packet Sniffing Interface. The core feature is provided by the EXata Network Virtual Interface Card (ENETV) that acts as a bridge between the third party packet sniffing and analysis tools at one end, and the emulated network within EXata at the other. When the Packet Sniffing Interface feature is enabled, EXata configures this virtual network card on the emulation server and forwards all internal traffic within the emulated network to this virtual card. In addition, the Packet Sniffing Interface can correctly capture and show all the traffic that was generated from the operational hosts.

At the other end, the virtual network card is treated by the packet sniffing software as a physical network card which can be ‘tapped into’ to receive the packet streams. The high fidelity of the EXata protocol stack ensures that the packets written to this network card conform to the standards so that the packet sniffing tools can capture and parse these packets correctly.

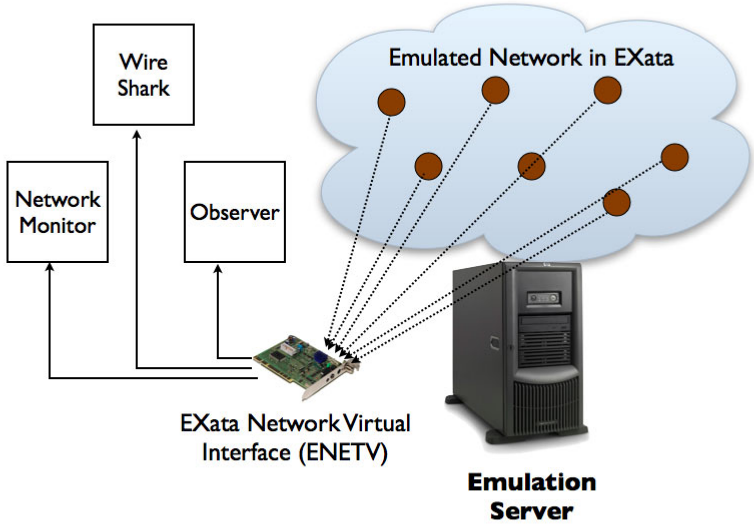


FIGURE 5-55. EXata Packet Sniffing Interface

The Packet Sniffing Interface in EXata is capable of displaying the packet contents and headers correctly for the transport layer (TCP and UDP), network layer (IPv4, IPv6, ICMP, and ICMPv6) and many routing protocols (including, OLSR, OSPF), applications (including FTP, HTTP), and MAC layer protocols.

By default, the ENETV displays all packets with *encapsulated Ethernet* MAC headers, i.e., the MAC headers of any custom or non-standard MAC protocol are converted into the familiar Ethernet MAC header format before being displayed by the ENETV.

EXata also supports the IEEE 802.11 MAC protocol and is capable of displaying the data, control and management frames of IEEE 802.11 MAC.

Note: Packet sniffing of the IEEE 802.11 MAC protocol is supported only on Linux emulation servers.

Installing ENETV on Windows

The ENETV driver can be installed along with the main EXata installation. Refer to *EXata Installation Guide* for details.

If the ENETV driver was not installed during the main installation, it can be separately installed by performing the following steps:

1. Open a command window.
2. Change the directory to EXATA_HOME/interfaces/pas/virtual_windows/<OS-version>/<system-arch>, where <OS-version> is the operating system version and <system-arch> is the architecture of the system on which ENETV driver needs to be installed.

3. Type the following command (all arguments must be entered on the same line):

For 32-bit systems:

```
%EXATA_HOME%\interfaces\pas\virtual_windows\devcon.exe install
enetv.inf "root\enetv"
```

For 64-bit systems:

```
%EXATA_HOME%\interfaces\pas\virtual_windows\devcon_amd64.exe install
enetv.inf "root\enetv"
```

Note: Alternatively, you can install the ENETV driver by using the batch file install-enetv.bat in directory the EXATA_HOME/interfaces/pas/virtual_windows.

To test that the EXata Network Virtual driver has been successfully installed, open the Networks Connections window from **Start > Control Panel > Network and Sharing Center > Change adapter settings**. Right-click in the window and click on **View > Tiles**. The EXata Device should be listed as shown in Figure 5-56.

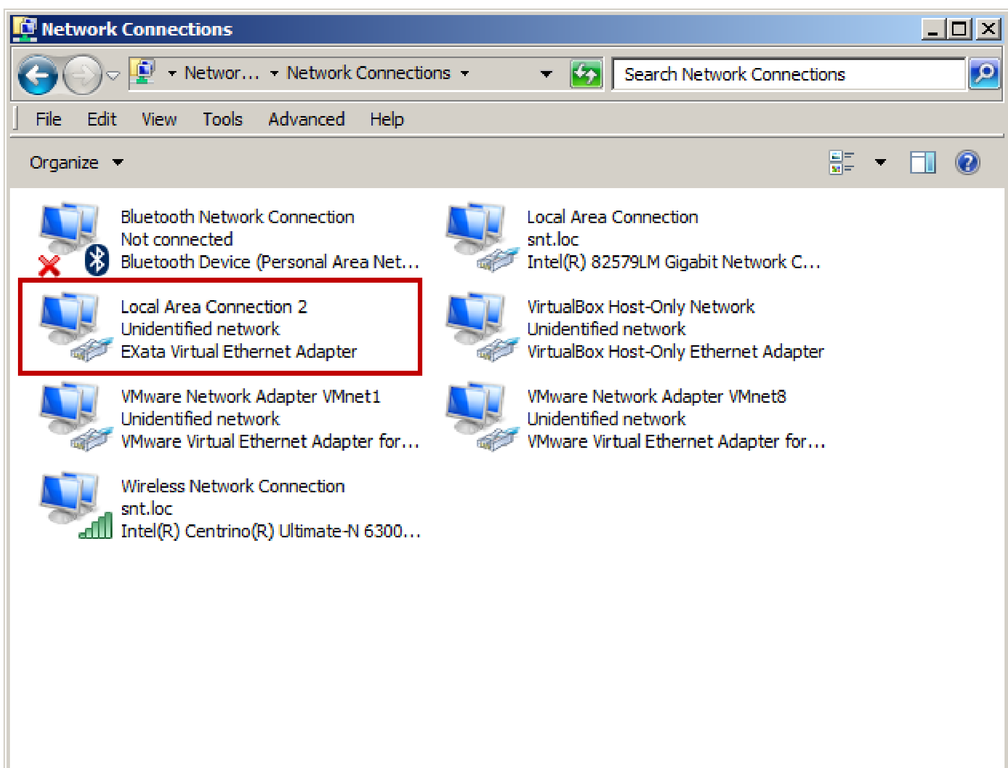


FIGURE 5-56. EXata Virtual Network Device

Uninstalling ENETV on Windows

To uinstall the ENETV driver, performing the following steps:

1. Open a command window.
2. Changing the directory to EXATA_HOME/interfaces/pas/virtual_windows/<OS-version>/<system-arch>, where <OS-version> is the operating system version and <system-arch> is the architecture of the system on which ENETV driver needs to be uninstalled.
3. Type the following command (all arguments must be entered on the same line):

   For 32-bit systems:

   ```
   %EXATA_HOME%\interfaces\pas\virtual_windows\devcon.exe remove
   enetv.inf "root\enetv"
   ```

   For 64-bit systems:

   ```
   %EXATA_HOME%\interfaces\pas\virtual_windows\devcon_amd64.exe remove
   enetv.inf "root\enetv"
   ```

Note: Alternatively, you can unnstall the ENETV driver by using the batch file uninstall-enetv.bat in directory the EXATA_HOME/interfaces/pas/virtual_windows.

Installing ENETV on Linux

On Linux, EXata automatically attempts to compile, load, and configure the ENETV driver each time the EXata process is executed. In order to use the EXata Packet Sniffer interface, Linux kernel source code or header files must be present. To verify if these are available on the emulation server, open a terminal window and do the following:

- For Debian or Ubuntu systems, type:

  ```
  dpkg -s kernel
  dpkg -s kernel-headers
  ```

- For other systems, type:

  ```
  rpm –q kernel
  rpm –q kernel-headers
  ```

If the above commands fail, consult your system administrator on how to install the kernel source or header files on the emulation server.

Using the Packet Sniffing Interface

To use the Packet Sniffing Interface, perform the following steps:

1. Enable the Packet Sniffing Interface and select node(s) for which the packets should be captured. See Section 4.3.3 for details on how to configure the Packet Sniffing Interface parameters for the command line interface and the GUI.

   If a specific node is selected for sniffing packets and the EXata GUI is running, the selected node is highlighted on the canvas by a brown triangle (see node 10 in Figure 5-57).

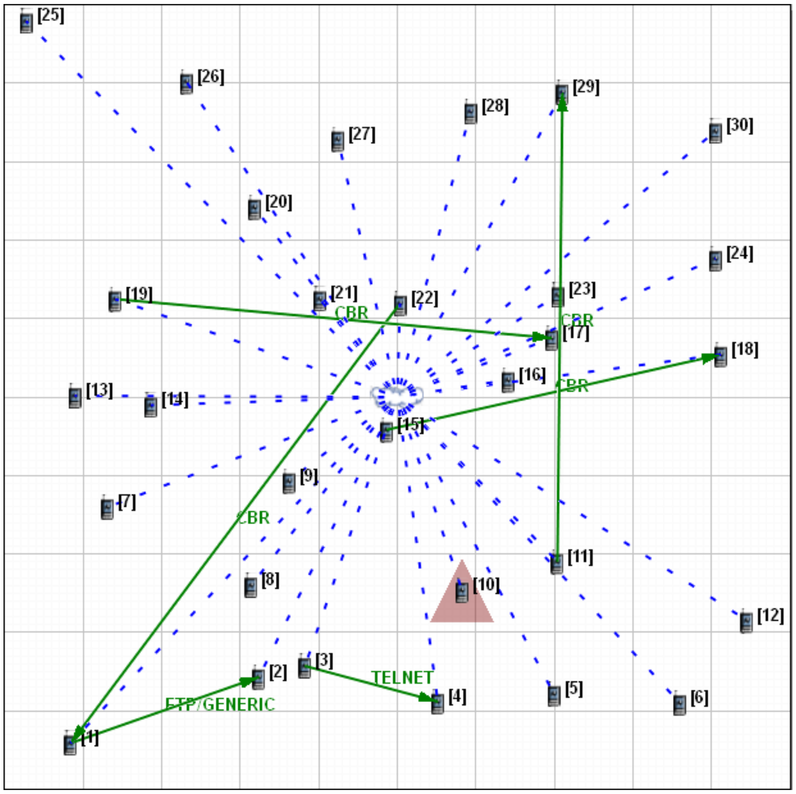


FIGURE 5-57. EXata GUI Displaying a Packet Sniffing Node

2. Open Wireshark, Observer or any of other packet sniffing tool. Locate the toolbar in this software that asks for the network device for which to capture packets on. Provide the EXata Virtual Network Device (SNT EXata Virtual Interface for Windows and exata or exata_dot11 for Linux) as input to this configuration toolbar. Figure 5-58 shows SNT EXata Virtual Interface in 'Capture Options' windows in Wireshark.

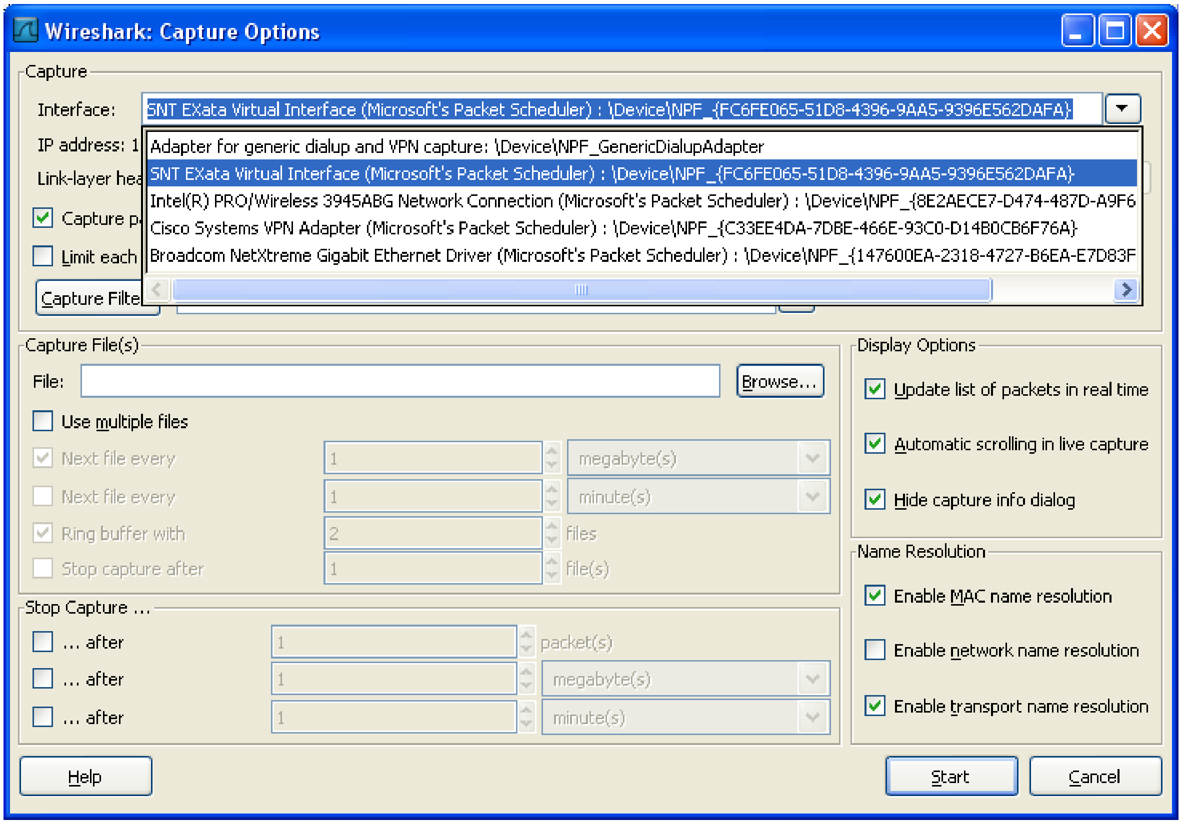


FIGURE 5-58. EXata Interface Selection in Wireshark

3. Start the packet sniffing software to begin capturing the packets.
4. Start the EXata process by pressing the **Play** button in GUI mode, or launching EXata in command line.

Verify that the packet sniffing software is able to capture and display the packets. All the transmitted and received packets for the node that was configured to capture packets should be displayed. Figure 5-59 show an example of packets displayed in Wireshark:

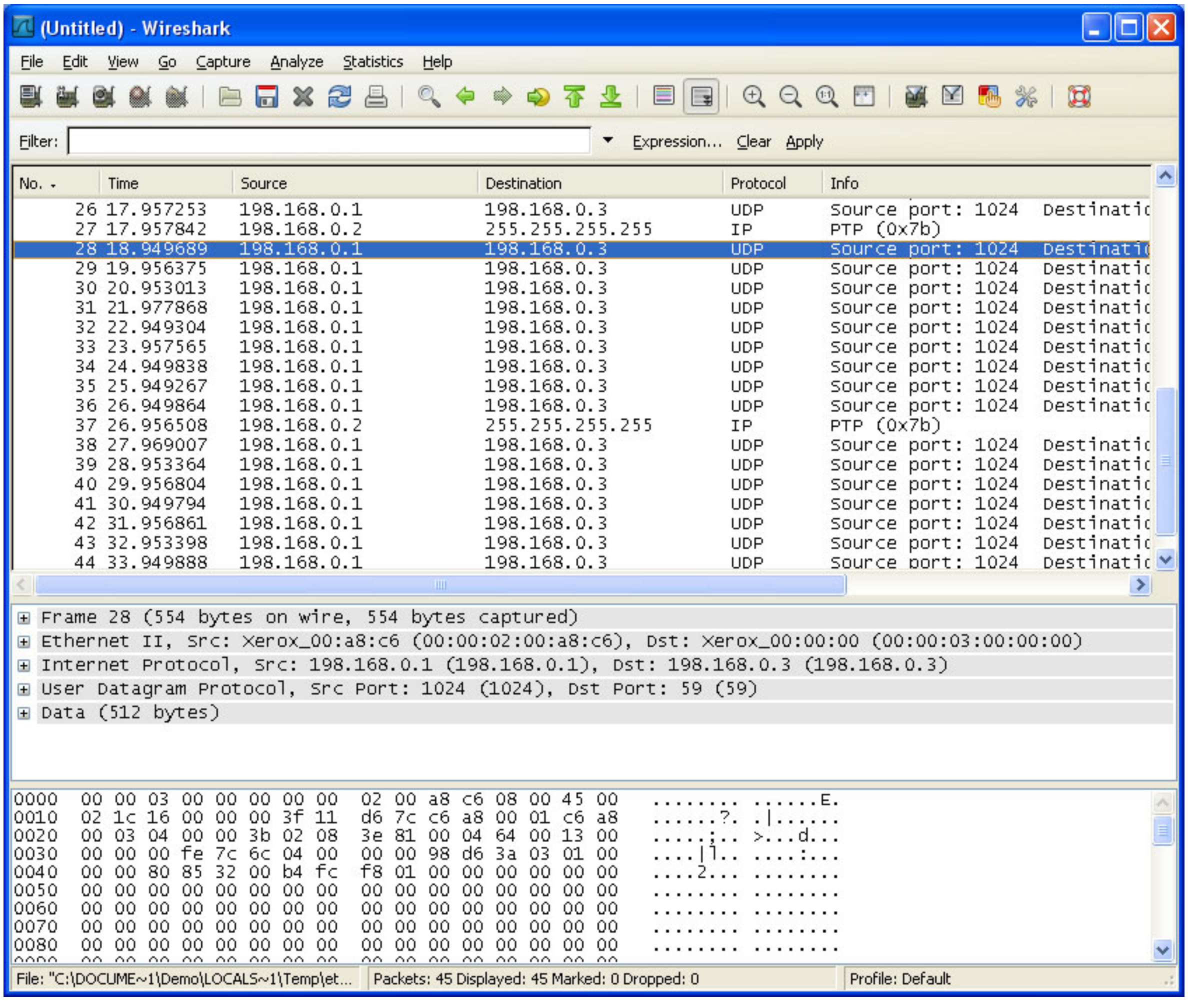


FIGURE 5-59. Captured Packet Contents in Wireshark

5.7 Configuring EXata for Parallel and Distributed Environments

EXata is a fully parallel network emulator with the simulation and emulation kernels designed from ground up for the parallel and distributed computing architectures. The parallel simulation kernel offers unprecedented execution speedups and scalability of emulated network scenarios. The parallel emulation kernel delivers scalability in the physical assets connected to the emulator.

In the shared memory Symmetric Multiprocessor (SMP) architectures, the simulation and emulation kernels transparently and seamlessly handle the parallelism and require no specific or additional configuration from the user.

Under distributed cluster-based architectures, EXata offers options to run the emulation kernel on one cluster node only or all cluster nodes.

When running EXata on one cluster node, the operational hosts connect to one cluster node only, and the entire emulation scenario is visible to these operational hosts via Connection Manager. Figure 5-60 illustrates an example where the emulated network scenario is spread across two cluster nodes. In this case, which is also the default case, the operational hosts can connect to the master cluster only. Connection Managers running on these cluster nodes will display all nodes in the emulated network, irrespective of the location of these nodes. It is, therefore, transparent to the user if EXata is running on a sequential, shared memory SMP, or distributed computing platform..

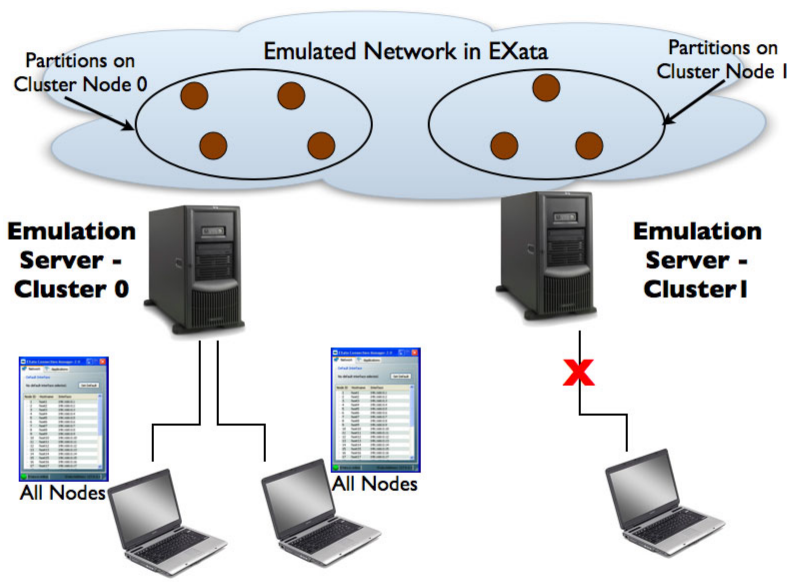


FIGURE 5-60. All Operational Hosts Connecting to Single Cluster Node

It may be desirable, for reasons such as distributing the traffic load from Operational hosts, to allow the Operational Hosts to connect to each of the cluster nodes. EXata offers a simple and seamless way to accomplishing this. Figure 5-61 illustrates, for the same example from above, the schematics for this mode of Operation. In this approach, the Operational Hosts can connect to any of the cluster nodes, thus distributing the traffic load between the Hosts and EXata.

It should be noted that in this case the Operational Hosts can connect to the EXata nodes only that reside on the cluster that they are connected to. Connection Manager makes this configuration easy, by displaying only those nodes in the *Network View* that belong the correct cluster node.

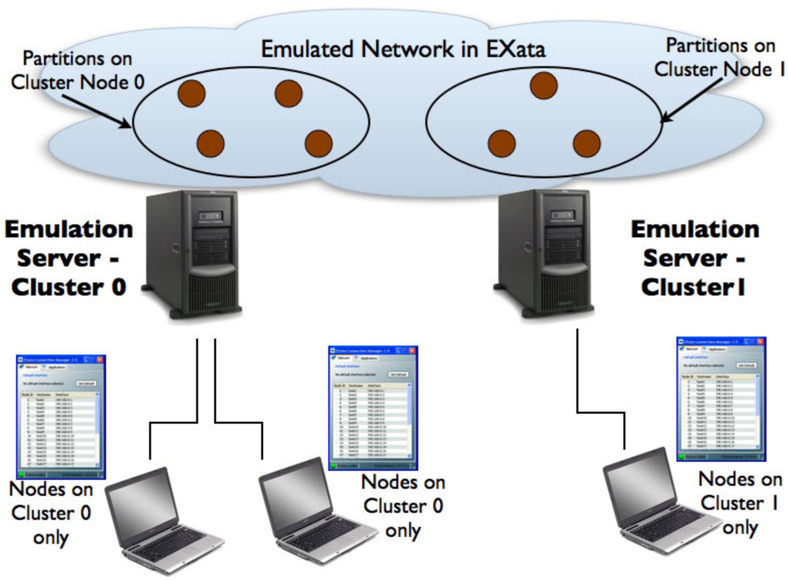


FIGURE 5-61. Operational Hosts Connecting to all Cluster Nodes

See Section 4.3.5.2 for details on how to configure cluster parameters for command line interface and the GUI.

6 EXata Architect: Visualize Mode

This chapter provides an overview of the various options, commands and functions in the Architect Visualize mode, which are used to run and animate experiments. Design mode of Architect is described in Chapter 3.

6.1 Components of Architect

This section provides an overview of the different components of Architect, but focuses on the features available in Visualize mode (see Figure 6-2).

Selecting Execution Mode

To visualize a scenario in Architect GUI, first select the scenario execution mode (simulation or emulation) from the **Emulation** toolbar.

FIGURE 6-1. Selecting Execution Mode in EXata GUI

Notes:

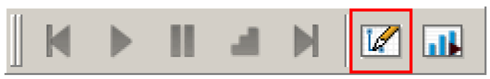
1. To run a scenario in emulation mode in Architect, EXata GUI must be launched from an administrator or root user account.
2. In EXata GUI, the execution mode (emulation or simulation) can be selected only in Design mode of Architect.
3. If emulation mode is enabled, then some emulation features (Section 4.3) can be configured in both Design and Visualize modes, while the other emulation features can be configured only in Design mode.

Switching between Design Mode and Visualize Mode

After creating a scenario or opening a saved scenario and selecting the execution mode in Design mode, press the **Run Simulation** button to initialize the scenario. This changes the mode of Architect from Design mode to Visualize mode. In Visualize mode you can run and animate the scenario and perform run-time analysis. This way of visualizing a scenario is called *Live Simulation Mode*.

You can also switch to Visualize mode from Design mode by opening a .ani file. This way of visualizing a scenario is called *Recorded Animation Mode*.

When a scenario is run in Live Simulation mode, after the simulation has completed or has been stopped, you can switch back to Design mode for the same scenario by clicking the **Switch to Design Mode** button.



Note: You can not switch back to Design mode for the same scenario by clicking **Switch to Design Mode** button if the simulation is run in Recorded Animation mode.

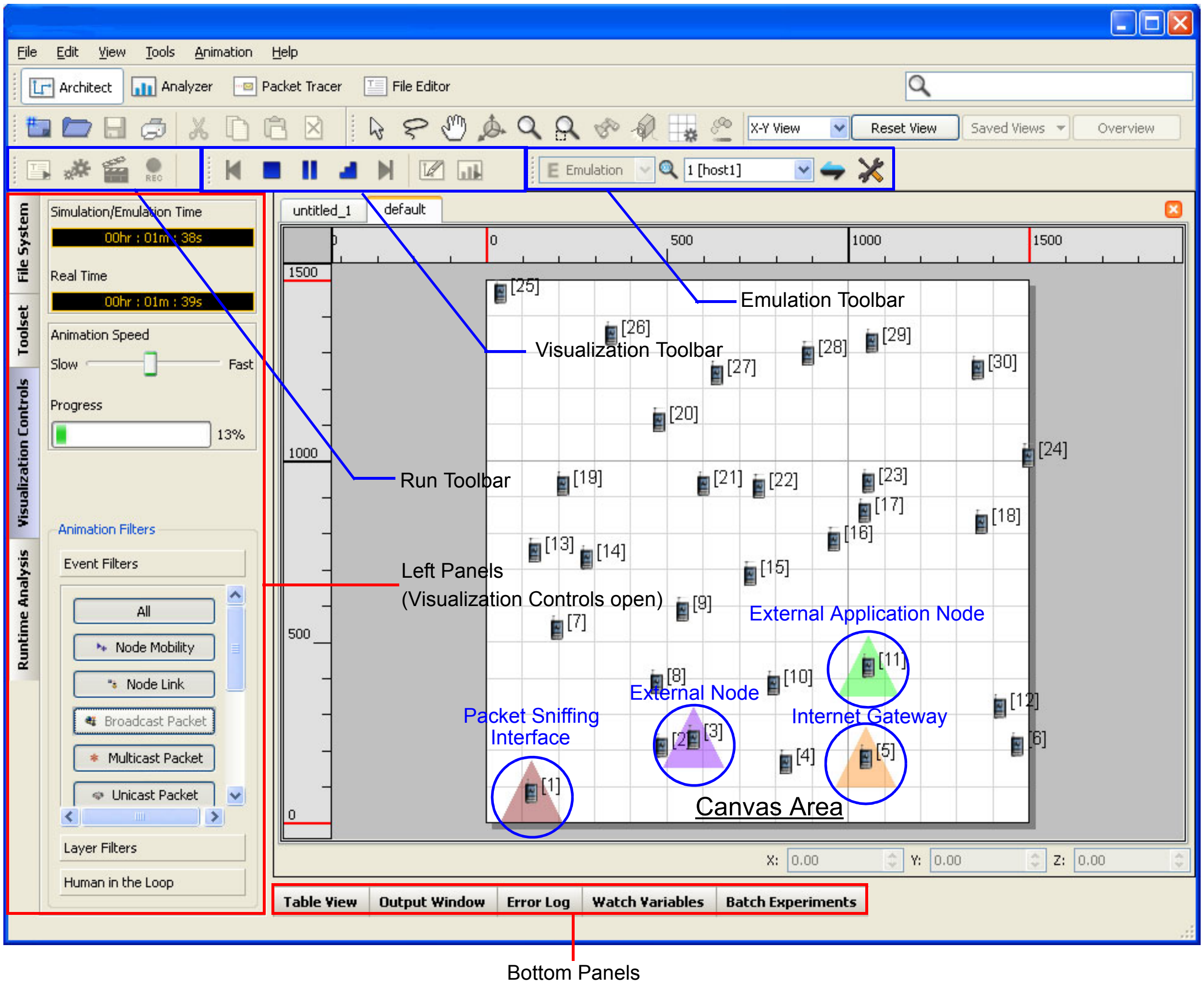


FIGURE 6-2. Architect Visualize Mode Layout

6.2 Canvas Symbols

Fault Symbols

When a scenario is run in Architect's Visualize mode, the following symbols appear dynamically next to nodes with one or more interface faults:

- A red circle with a diagonal indicates that all interfaces of the node have failed.
- An orange circle with a diagonal indicates that at least one of the interfaces of the node (but not all) has failed.

Interface faults may occur at a node in one of the following ways:

- Static or dynamic faults are configured at a node, interface, or point-to-point link. Refer to the Faults section of *Developer Model Library* for details.

- A node is deactivated by including a deactivate command in the Human in the Loop Configuration File (see Section 4.2.11.2) or by sending a deactivate command from the Human in the Loop Interface (see Section 4.2.11.2). When a node is deactivated, all its interfaces fail. When the node is subsequently activated, all interfaces of the node are repaired.
- A node or interface is deactivated/activated by an external program through an external interface.

Node Highlights

When a scenario is run in emulation mode in the GUI, some of the nodes are highlighted by colored triangles (see Figure 6-2).

- Purple triangle: Indicates an External Node, i.e., this node is mapped to an operational host (see Section 5.2.1.1.1).
- Green triangle: Indicates an External Application Node, i.e., an application is run on this node form an operational host (see Section 5.2.1.1.2).
- Orange triangle: Indicates the Internet Gateway Node, i.e., traffic between EXata nodes and operational hosts on the one end and the Internet on the other is routed through this node (see Section 5.4).
- Brown triangle: Indicates a Packet Sniffing Node, i.e., this node is selected for sniffing packets (see Section 5.6).

The size of the triangles in 3D view can be changed. See Section 4.2.4.2 for details.

6.3 Menus

This section describes the menus available from the **Menu** bar.

6.3.1 File Menu

See Section 3.1.1.1 for details of the **File** menu.

6.3.2 Edit Menu

See Section 3.1.1.2 for details of the **Edit** menu.

6.3.3 View Menu

See Section 3.1.1.3 for details of the **View** menu.

6.3.4 Tools Menu

See Section 3.1.1.4 for details of the **Tools** menu.

Note: The **Tools** menu is enabled only in the Design mode.

6.3.5 Animation Menu

The **Animation** menu provides commands to control various animation settings.

Note: The **Animation** menu is automatically enabled when the scenario is run in Live Simulation mode.

In Recorded Animation mode, the **Animation** menu gets enabled when the **Play** button is clicked. Some commands from the **Animation** menu are not available in Recorded Animation mode.

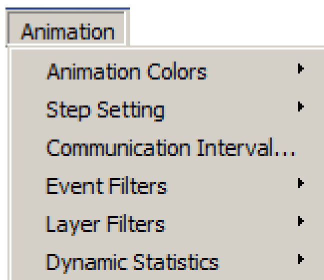


FIGURE 6-3. Animation Menu

6.3.5.1 Animation Colors Command

The **Animation Colors** command enables the user to select colors for events at different layers. Figure 6-4 shows the list of layers. Figure 6-5 shows the list of channel events for which a color can be selected. The same list of events is available for all layers. To set the color for an event, click on the event in the list. This opens a color palette from which a color can be selected.

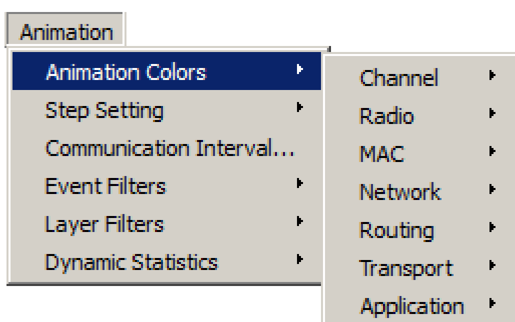


FIGURE 6-4. Animation Colors Command

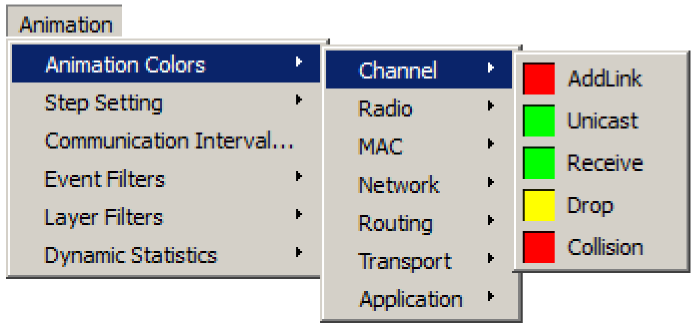


FIGURE 6-5. Setting Colors for Events

6.3.5.2 Step Setting Command

The **Step Setting** command determines the behavior of the **Step Forward** button of the **Animation** tool bar (see Section 6.4.4). When the **Step Forward** button is pressed, the simulation advances either by a pre-configured time interval or by a pre-configured number of animation commands. The **Step Setting** command lets the user choose between these two options and configure the step size (time interval or number of animation commands).

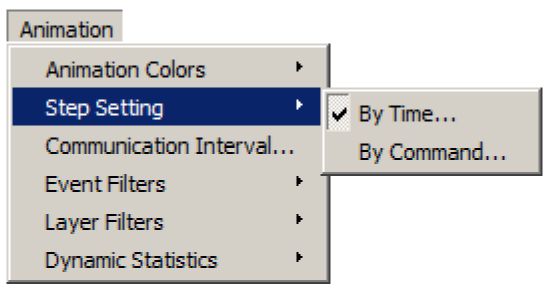


FIGURE 6-6. Step Setting Command

By Time

If this option is selected, the simulation will advance by a fixed time interval every time the **Step Forward** button is pressed. Selecting this command opens a **Step Interval Time** dialog box (Figure 6-7) in which the step size (as a time interval) can be specified.

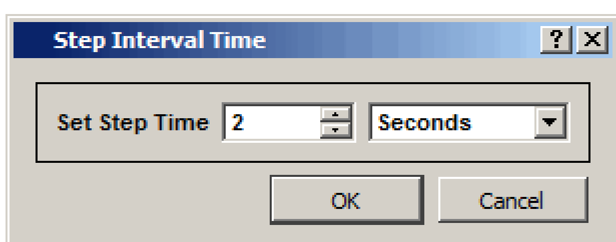


FIGURE 6-7. Step Interval Time Dialog Box

By Command

If this option is selected, the simulation will advance by a fixed number of simulation commands every time the **Step Forward** button is pressed. Selecting this command opens a **Step Command Interval** dialog box (see Figure 6-8) in which the step size (in number of commands) can be specified.

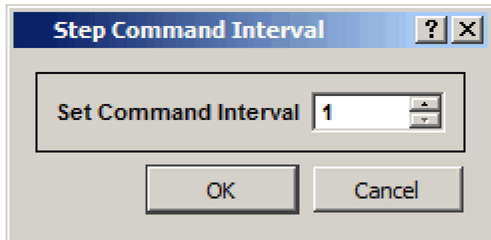


FIGURE 6-8. Step Command Interval Dialog Box

6.3.5.3 Communication Interval Command

The **Communication Interval** command is used to set the frequency of communication between the simulator and Architect (i.e., how often the simulator sends animation commands to Architect).

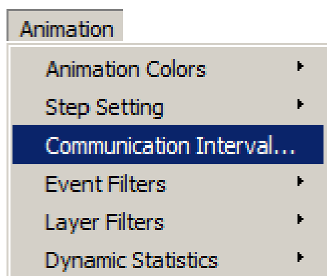


FIGURE 6-9. Communication Interval Command

Selecting this command opens the Communication Interval dialog box in which the communication interval can be specified (see Figure 6-10).

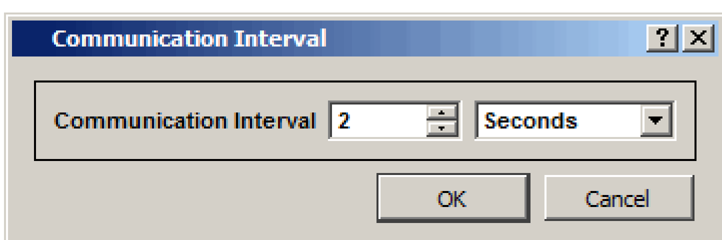


FIGURE 6-10. Communication Interval Dialog Box

Notes: **1.** This command is not available in Recorded Animation mode.

2. In general, setting the communication interval to a lower value may reduce the speed of the simulation and setting it to a higher value may increase the speed of the simulation. However, setting the communication interval to a very high value can cause the GUI buffer to overflow, and cause the simulation to run slow. It is not recommended to set the communication interval to a value much higher than the default value (2 seconds).

6.3.5.4 Event Filters Command

The **Event Filters** command is used to enable or disable animation of different events. The events to animate can be selected from the displayed list.

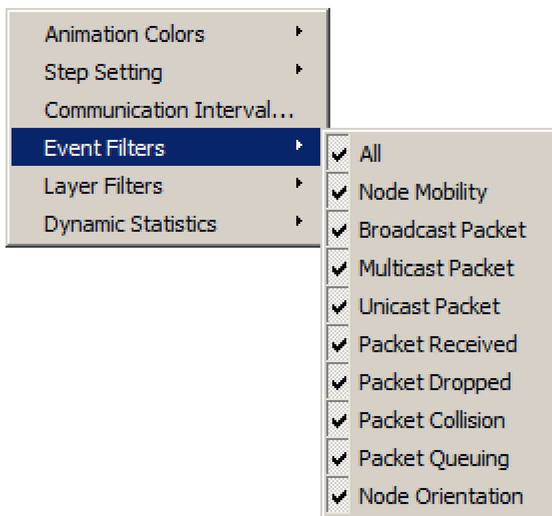


FIGURE 6-11. Event Filters Command

Note: You can also enable or disable animation of different events from the **Visualization Controls** panel (see Section 6.5.3.2). See Table 6-1 for a description of event filters.

6.3.5.5 Layer Filters Command

The **Layer Filters** command is used to enable or disable animation at different layers. The layers can be selected from the list that is displayed.

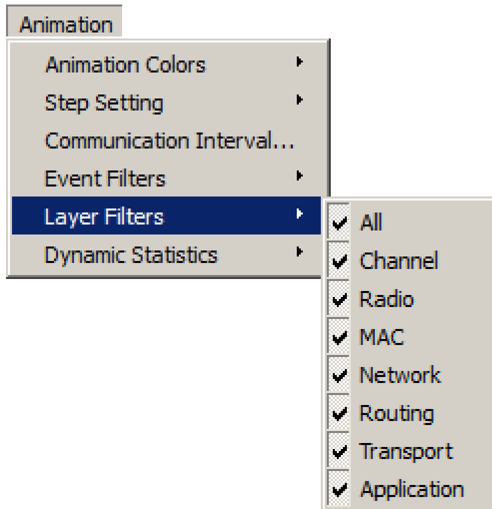


FIGURE 6-12. Layer Filters Command

Note: You can also enable or disable animation at different layers from the **Visualization Controls** panel (see Section 6.5.3.2). See Table 6-2 for a description of layer filters.

6.3.5.6 Dynamic Statistics Command

The **Dynamic Statistics** command is used to configure graph properties and dynamic statistics at the scenario level.

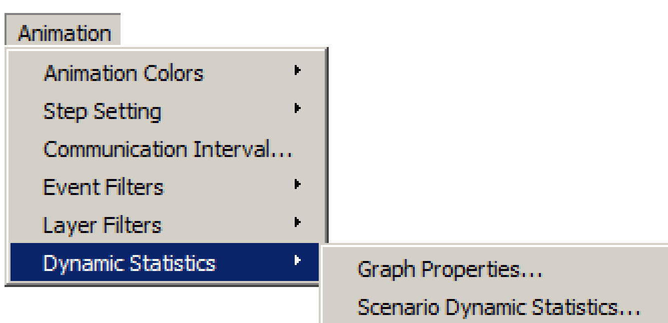


FIGURE 6-13. Dynamic Statistics Command

6.3.5.6.1 Graph Properties

Selecting this option opens the **Graph Properties** dialog box which is used to customize the properties of graphs that display dynamic statistics.

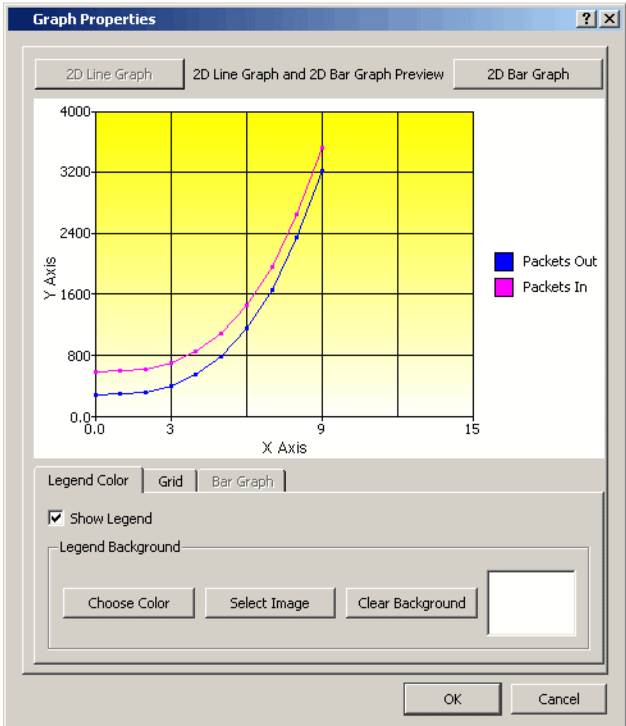


FIGURE 6-14. Graph Properties: Legend Color Tab

You can customize properties for line graphs and bar graphs by clicking the **2D Line Graph** button and **2D Bar Graph** button, respectively.

Legend Color Tab

The legend display can be customized using this tab (see Figure 6-14). To display legends on graphs, check the **Show Legends** box.

If the **Show Legends** box is checked, you can customize the legend display by using the following options:

- **Choose Color**: You can select a background color by clicking the **Choose Color** button and selecting a color from the color palette that opens.
- **Select Image**: You can select a background image for the legend by clicking the **Select Image** button and selecting an image file from the file selector that opens.
- **Clear Background**: You can clear the legend background by clicking the **Clear Background** button.

Grid Tab

The grid display can be customized using this tab.

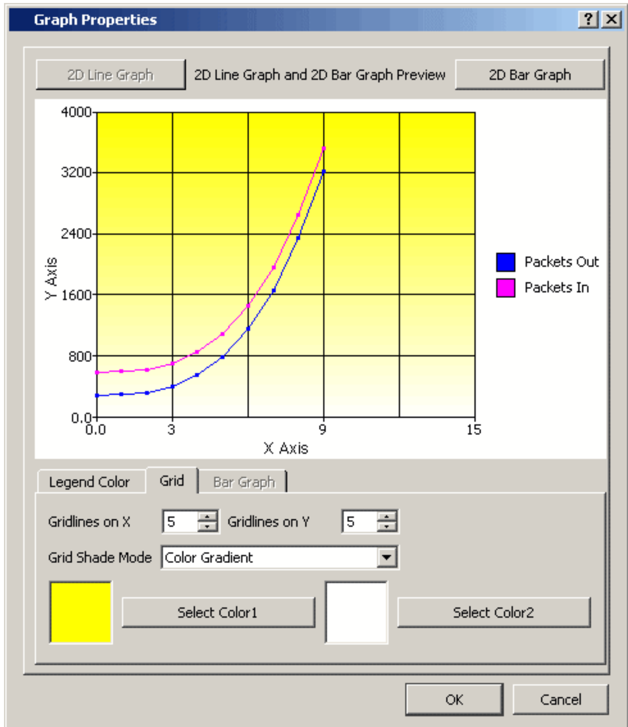


FIGURE 6-15. Graph Properties: Grid Tab

You can customize the grid display by using the following fields:

- **Grids on X**: Enter the number of vertical lines in the grid in this field.
- **Grids on Y**: Enter the number of horizontal lines in the grid in this field.
- **Grid Shade Mode**: This field is used to set the shading of the grid. The following options are available from the pull-down menu:
  - **No Background**: Select this option to display a white background for the grid.
  - **Constant Color**: Select this option to display a solid color in the grid background. The color can be set by clicking the **Select Color1** button.
  - **Color Gradient**: Select this option for a background that is a continuous transition from one color to another. The two colors for the gradient can be set by clicking the **Select Color1** and **Select Color2** buttons.

Bar Graph Tab

This tab is used to customize display of bars in bar graphs.

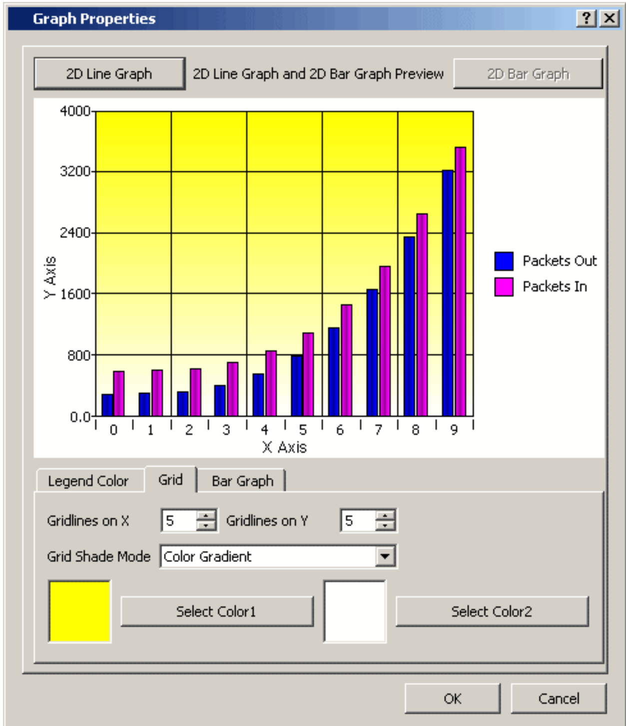


FIGURE 6-16. Graph Properties: Bar Graph Tab

Note: The Bar Graph tab is enabled only when the **2D Bar Graph** button is clicked.

To enable color shades in bars, check the **Enable Bar Shades** box. If bar shades are not enabled, the bars appear in a solid color. If bar shades are enabled, you can select the shade mode by selecting the appropriate button in the Bar Shade Mode area.

6.3.5.6.2 Scenario Statistics

Selecting this option opens a dialog box that is used to configure scenario level dynamic statistics.

Note: See Section 6.5.4.1.1.2 for configuring node level dynamic statistics.

To configure scenario level dynamic statistics, perform the following steps:

1. Click **Animation > Dynamic Statistics > Scenario Statistics**. This opens the dialog box shown in Figure 6-17.

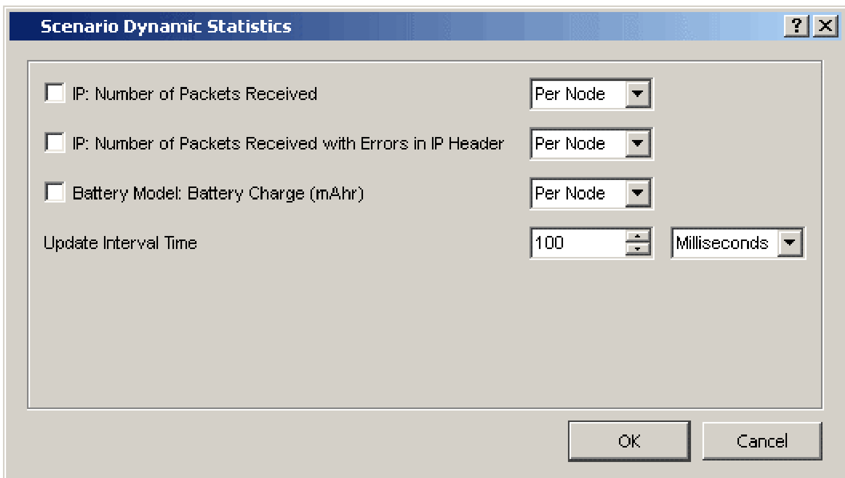


FIGURE 6-17. Configuring Scenario Level Dynamic Statistics

2. Select the statistics to display by clicking its check box. The available statistics depend on the protocols in the scenario.
3. For each selected statistic, select the statistic type (per-node total or system aggregate) from the pull-down list.
4. Set the statistic interval time. This is the interval at which the statistics are updated by the simulator.
5. Click **OK**.

Graphs for the selected statistics are displayed in the workspace under the scenario. Per-node statistics are shown as bar graphs and aggregate statistics are displayed as line graphs.

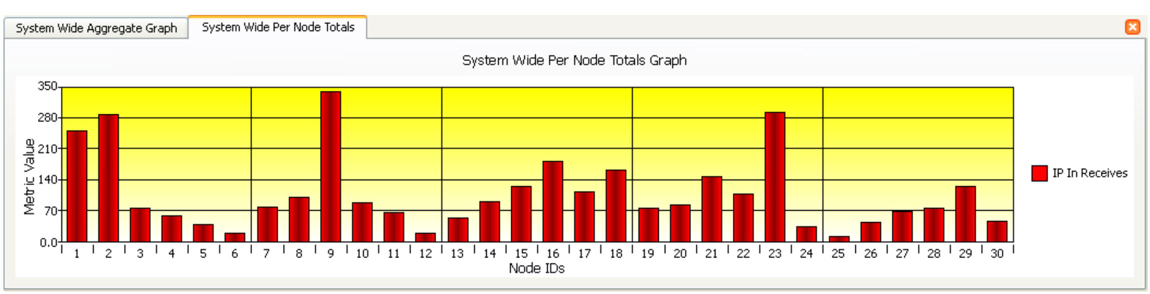


FIGURE 6-18. Bar Graph Showing System Wide Per Node Dynamic Statistics

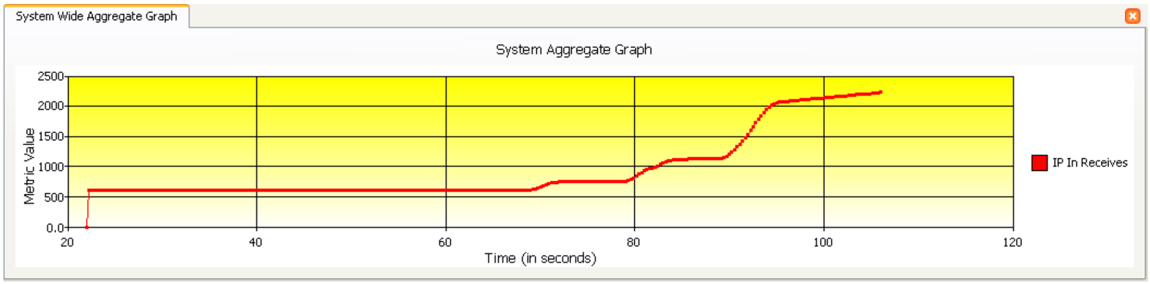


FIGURE 6-19. Line Graph Showing System Wide Aggregate Dynamic Statistics

For bar graphs (per-node statistics), the X-axis corresponds to node IDs and the Y-axis corresponds to the metric value. For line graphs (aggregate statistics), the X-axis corresponds to time and Y-axis corresponds to the metric value.

If both per-node and aggregate statistics are selected, they are displayed in separate tabs. When the simulation is run by clicking the **Play** or **Step Forward** buttons, the statistics graphs are continuously updated as the simulation proceeds.

6.4 Toolbars

This section describes the toolbars available in Architect.

6.4.1 Standard Toolbar

See Section 3.1.2.1 for details of the **Standard** toolbar.

6.4.2 View Toolbar

See Section 3.1.2.2 for details of the **View** toolbar.

6.4.3 Run Toolbar

See Section 3.1.2.3 for details of the **Run** toolbar.

6.4.4 Visualization Toolbar

The **Visualization** toolbar contains visualization control buttons.

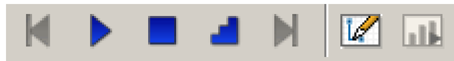
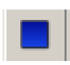
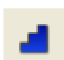


FIGURE 6-20. Visualization Toolbar

Button	Function	Description
	Rewind to Start	Used to rewind the animation to the beginning. **Note:** This option is available only in Recorded Animation mode.
 	Play/Pause	Used to start, pause, and resume the animation. This button toggles between **Play** and **Pause**.
	Stop	Used to stop the animation.
	Step Forward	Used to run the animation in step mode. Clicking on this button advances the simulation by a pre-configured length of simulation time or number of simulation commands.
	Forward to End	Used to advance the animation to the end. **Note:** This option is available only in Recorded Animation mode.
	Switch to Design Mode	Used to switch to Design mode of Architect with the current scenario. **Note:** This option is available only in Live Simulation mode and is enabled only after the simulation has completed or has been stopped.
	Analyze Statistics	Used to switch to Analyzer where statistics for the current scenario can be viewed. **Note:** This option is available only in Live Simulation mode and is enabled only after the simulation has completed.

6.4.5 Emulation Toolbar

See Section 3.1.2.5 for a description of the Emulation Toolbar.

6.5 Left Panels

The following panels are available to the left of the canvas:

- File System
- Toolset
- Visualization Controls
- Runtime Analysis

Note: These four panels occupy the same space and at most one of them can be open at any time. By default, the **Toolset** panel is open.

6.5.1 File System Panel

See Section 3.1.3.1 for a description of the **File System** panel.

6.5.2 Toolset Panel

The toolset panel is enabled only in Design mode. See Section 3.1.3.2 for a description of the **Toolset** panel.

6.5.3 Visualization Controls

The **Visualization Controls** panel provides various controls for monitoring and controlling the scenario execution. The top part of the **Visualization Controls** is for displaying scenario execution status, while the bottom part is for configuring animation filters and Human-In-The-Loop (HITL) interactions.

6.5.3.1 Status Display

This panels displays the simulation/emulation time, real time, animation speed, and progress of the scenario execution (see Figure 6-21). It also displays the Real-time Execution Status indicator if the display is enabled in Display Settings (see Section 3.1.1.3.1).

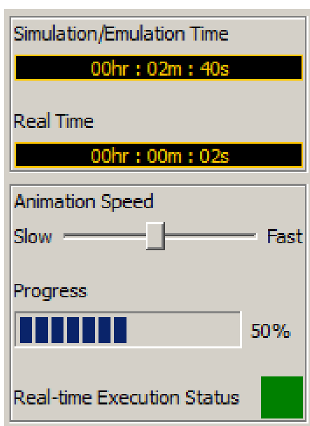


FIGURE 6-21. Status Display

The Real-time Execution Status indicator shows whether real time and emulation time are in synch. The indicator is green if real time and emulation time are in synch. The indicator turns yellow if the difference between real time and emulation time is more than x seconds but less than y seconds. The indicator turns red If the difference between real time and emulation time is y seconds or greater.

Simulation/Emulation Time	The **Simulation/Emulation Time** display shows the current simulation/emulation time.
Real Time	The **Real Time** display shows the actual time that has elapsed since the start of scenario execution.
Animation Speed	The **Animation Speed** control is a slide bar which can be used to control the speed of simulation. You can decrease or increase the speed of scenario execution by moving the slider to the left or right, respectively.
Progress	The scenario execution progress is displayed in the form of a progress bar and shows how much of the execution is complete as a percentage.
Real-time Execution Status	The Real-time Execution Status indicator shows whether real time and emulation time are in synch. The indicator is green if real time and emulation time are in synch. The indicator turns red If the difference between real time and emulation time is 1 second or greater. **Note:** The Real-time Execution Status indicator is displayed only if the option is selected in Display Settings (see Section 3.1.1.3.1).

6.5.3.2 Animation Filters

The **Animation Filters** area displays the event filters, layer filters, and human-in-the-loop interface. You can expand any filter list by clicking on the corresponding button. Figure 6-22 shows the animation filters.

Note:
1. Only one of the filters lists can be displayed at one time.
2. Clicking on **All** in the event or layer filter list selects or deselects all filters in the list.

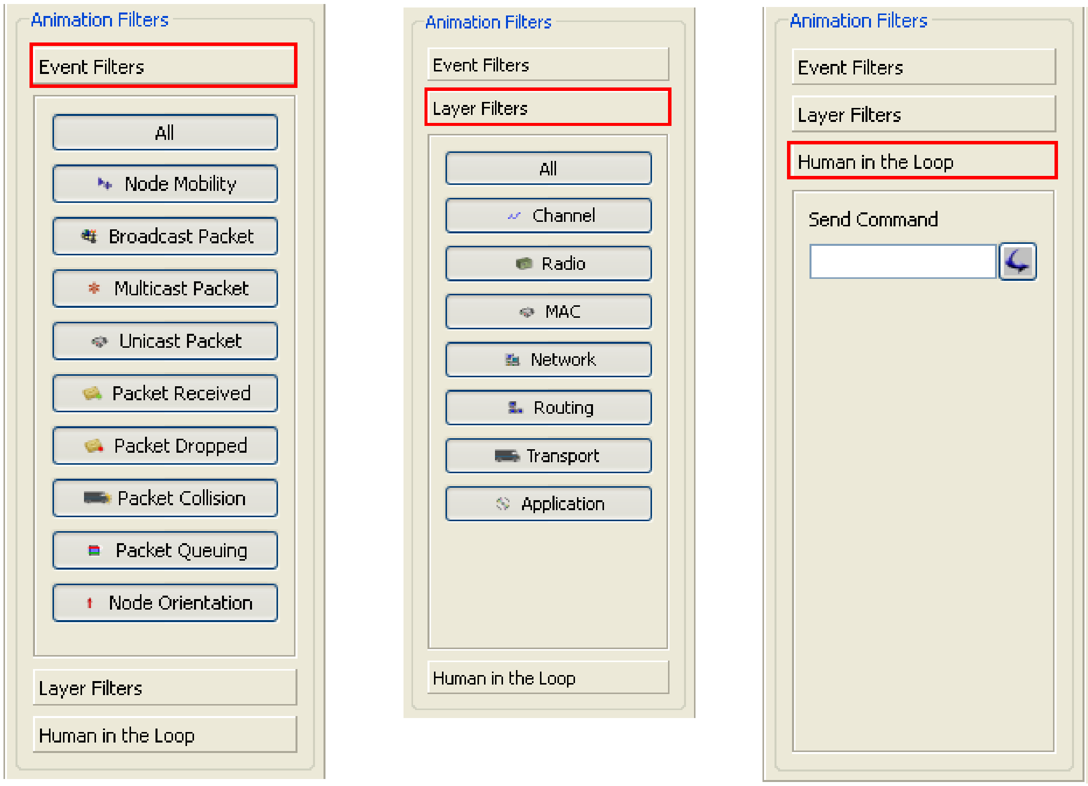


FIGURE 6-22. Animation Filters

You can select the colors for animation of events at different layers by using the Animation Colors command (see Section 6.3.5.1).

Note: Simulation speed is affected by both event and layer filters. In general, fewer the events and layers enabled for animation, faster the simulation speed. However, disabling event animations does not affect the simulation speed substantially, but disabling layer animations can significantly speed up the simulation.

6.5.3.2.1 Event Filters

The event filters in the **Visualization Controls** panel are used to enable or disable animation of events at the global level (i.e., for all nodes). Table 6-1 describes the animation effects controlled by the event filters.

Note: Global level event filters apply to all nodes in the scenario. However, animation of some events can be selectively disabled for specific nodes by applying node level event filters (see Section 6.5.4.1.1). Event animation at a node can not be enabled unless it is enabled at the global level.

TABLE 6-1. Event Filters

<table>
<tr><th>Filter</th><th>Description</th><th>Animation Effect</th></tr>
<tr><td>All</td><td>Enables or disables all event filters.</td><td></td></tr>
<tr><td>Node Mobility</td><td>Enables or disables the animation of node mobility.</td><td>Node position is updated as the node moves.</td></tr>
<tr><td>Broadcast Packet</td><td>Enables or disables the animation of broadcast events.</td><td rowspan="2">For Wired Subnets:
Lines are drawn from the subnet containing the source node to all nodes in the subnet.
For Wireless Subnets:
A sphere is drawn with the source node as the center.</td></tr>
<tr><td>Multicast Packet</td><td>Enables or disables the animation of multicast events.</td></tr>
<tr><td>Unicast Packet</td><td>Enables or disables the animation of unicast events.</td><td rowspan="2">For Wired Subnets:
The packet is represented by a an arrowhead which travels from the source to the destination.
For Wireless Subnets:
A directional line is drawn for a short time between the source and destination.
For both wired and wireless subnets, if the source and destination are in the same subnet, then the packet transmission is shown via the subnet.
Blue color is used for animation effects of packet events for traffic from external sources while green is used for simulated traffic.</td></tr>
<tr><td>Packet Received</td><td>Enables or disables the animation of packet received events.</td></tr>
<tr><td>Packet Dropped</td><td>Enables or disables the animation of packet dropped events.</td><td rowspan="2">A line is drawn from the source part of the way to the destination. The line ends in a “burst” effect.</td></tr>
<tr><td>Packet Collision</td><td>Enables or disables the animation of packet collision events.</td></tr>
</table>

TABLE 6-1. Event Filters (Continued)

Filter	Description	Animation Effect
Packet Queuing	Enables or disables the animation of packet queuing events. **Note:** All packet queuing events (joining a queue, leaving a queue, and being dropped from a queue) are controlled by this filter.	Size of the red bar representing the queue increases. When a packet is dropped from the queue due to overflow, a red box is displayed next to the base of the queue.
Node Orientation	Enables or disables the animation of node orientation changes.	Node icon is rotated in the X-Y view according to the new orientation.

6.5.3.2.2 Layer Filters

The layer filters in the **Visualization Controls** panel are used to control the animation of simulation events at the different layers. Table 6-2 describe the animation effects controlled by the layer filters.

TABLE 6-2. Layer Filters

Filter	Description
All	Enables or disables all layer filters. If all layers are disabled then only mobility events are animated.
Channel	Enables or disables the Channel layer animation.
Radio	Enables or disables the Radio layer animation.
MAC	Enables or disables the MAC layer animation.
Network	Enables or disables the Network layer animation.
Routing	Enables or disables the Routing layer animation.
Transport	Enables or disables the Transport layer animation.
Application	Enables or disables the Application layer animation.

6.5.3.3 Human-in-the-Loop Interface

The **Human-In-The-Loop** (HITL) interface is used to send commands to the simulator over the socket while the scenario is running.

Note: The HITL interface is enabled only for live simulations.

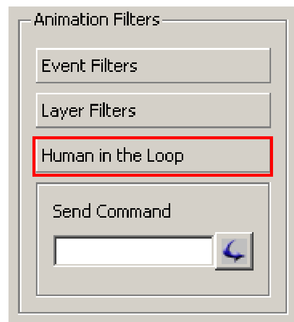


FIGURE 6-23. Human-in-the-Loop Interface

The text box is used to enter the command to send to the simulator. Pressing the  button sends the command to the simulator. The commands that can be used in HTIL scenarios are described in Table 6-3.

Note: Some model libraries support additional HITL commands. These are described in the model library documentation.

TABLE 6-3. Human-In-The-Loop Commands

Command	Description
D <node ID>	This command deactivates the node with specified node ID. A deactivated node can not communicate with other nodes. **Example**: D 11 The example above deactivates the node with ID 11. This command may also be executed by right-clicking on a node in Visualize mode, and selecting **Deactivate Node(s)**. **Note:** A red circle with a diagonal appears next to a deactivated node indicating that all interfaces of the node have failed.
A <node ID>	This command activates the node with the specified node ID. An activated node can communicate with other nodes again. **Example**: A 11 The example above activates the node with ID 11. This command may also be executed by right-clicking on a node in Visualize mode, and selecting **Activate Node(s)**.

TABLE 6-3. Human-In-The-Loop Commands (Continued)

Command	Description
`P <priority>`	This command changes the precedence of all CBR sessions in the scenario to the specified priority. `<priority>` should be an integer in t he range 0 ≤ `<priority>` ≤ 7. See *Developer Model Library* for details of CBR. **Example**: `P 3` The above example changes the precedence of all CBR sessions to 3.
`T <interval>`	This command changes the rate of all CBR sessions in the scenario by changing the CBR inter-packet interval. The new inter-packet interval is equal to `<interval>` milliseconds. `<interval>` should be an integer or real value. **Example**: The command `T 30` changes the inter-packet interval for all CBR sessions to 30 milliseconds.
`L <rate-factor>`	This command changes the rate of all CBR sessions in the scenario by changing the CBR inter-packet interval. The new inter-packet interval is equal to the product of the current inter-packet interval and `<rate-factor>`. **Example**: If the current interval is 0.1 second, the command `L 0.1` changes the interval to 0.01 second (= current interval * `<rate-factor>`).

6.5.4 Runtime Analysis Panel

The **Runtime Analysis** panel is used to observe properties of scenario components during the scenario execution. Node-level event filters can also be configured from the **Runtime Analysis** panel.

If dynamic parameters are enabled for the scenario (see Section 4.2.11), then values of dynamic parameters can also be viewed and set from the **Runtime Analysis** panel during runtime.

6.5.4.1 Scenario Component Properties and Animation Controls

The **Runtime Analysis** panel displays the scenario components (nodes, queues, and subnets) in a tree view. Expanding the **Nodes** group lists all nodes in the scenario. Expanding a node lists all interfaces of the node and expanding an **Interface** lists all queues at the interface. Similarly, expanding the **Subnets** group lists all subnets in the scenario.

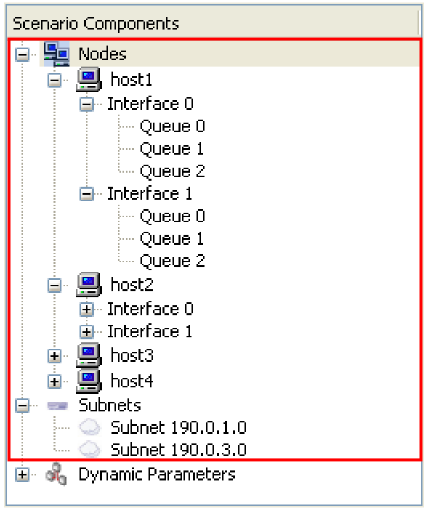


FIGURE 6-24. Scenario Components in Runtime Analysis Panel

6.5.4.1.1 Node Properties and Filters

To observe and configure properties of a node and set event filters for a node, expand the **Nodes** group and navigate to the node.

Selecting a node in the **Runtime Analysis** panel displays the event filters that can be applied to the node (see Section 6.5.4.1.1.1). The following commands are available if you right-click on the node name in the **Runtime Analysis** panel:

- **Dynamic Statistics**: Used for configuring node-level dynamic statistics (see Section 6.5.4.1.1.2)
- **Locate on Canvas**: Used to zoom to the node on the canvas

6.5.4.1.1.1 Event Filters

Event filters can be set for each node individually to control the granularity of animation (see Figure 6-25).

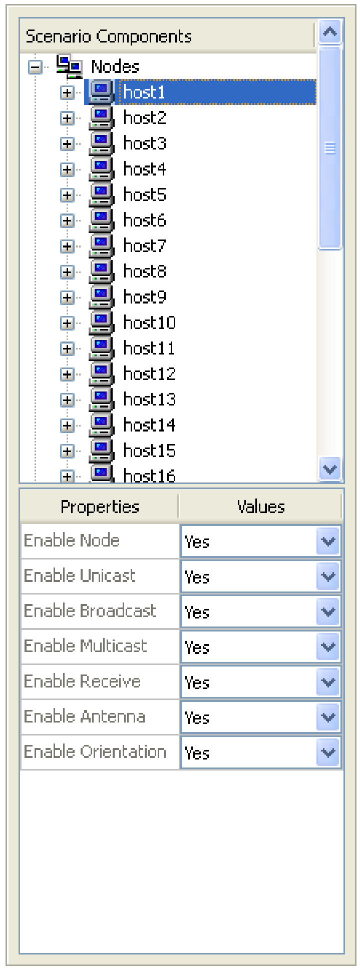


FIGURE 6-25. Animation Filters for Nodes

Table 6-4 lists the event filters that can be set for a node.

Note: Animation of some events can be selectively disabled for specific nodes by applying node level event filters. Event animation at a node cannot be enabled unless it is enabled at the global level (see Section 6.5.3.2).

TABLE 6-4. Node-level Event Filters

Filter	Description
Enable Node	Enables or disables animation for the node.
Enable Unicast	Enables or disables animation of unicast events for the node.
Enable Broadcast	Enables or disables animation of broadcast events for the node.
Enable Multicast	Enables or disables animation of multicast events for the node.
Enable Receive	Enables or disables animation of packet receive events for this node.
Enable Antenna	Enables or disables display of antenna patterns for the node.
Enable Orientation	Enables or disables animation of orientation events for this node.

6.5.4.1.1.2 Node-level Dynamic Statistics

To configure node level dynamic statistics, perform the following steps:

1. In the **Runtime Analysis** panel, expand the **Nodes** group and navigate to the desired node. Right-click on the node and select **Dynamic Statistics**.

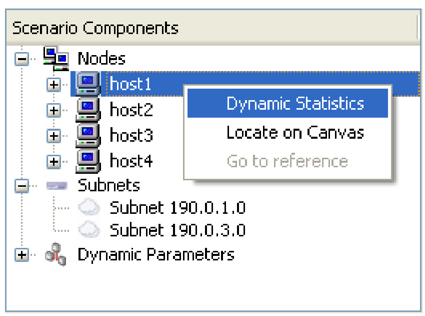


FIGURE 6-26. Configuring Dynamic Statistics for a Node

This opens the **Per Node Dynamic Statistics** dialog box (see Figure 6-27).

2. Select the statistics to display by clicking its check box. The available statistics depend on the protocols running at the node in the scenario.

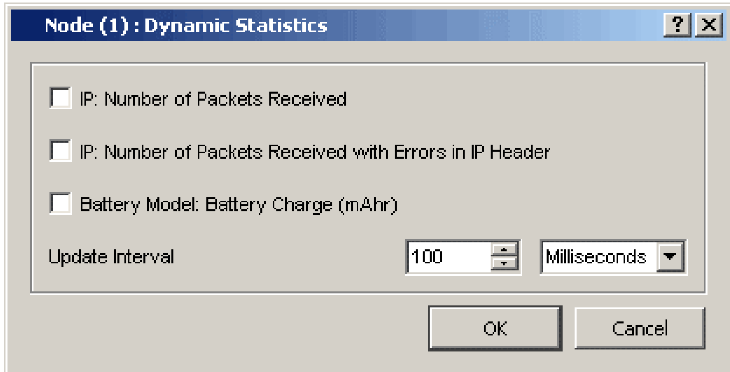


FIGURE 6-27. Per Node Dynamic Statistics Dialog Box

3. Set the statistic interval time. This is the interval at which the statistics updated by the simulator.
4. Click **OK**.

Per node dynamic statistics are displayed below the scenario in the central workspace. Figure 6-28 shows a sample statistics graph for a node.

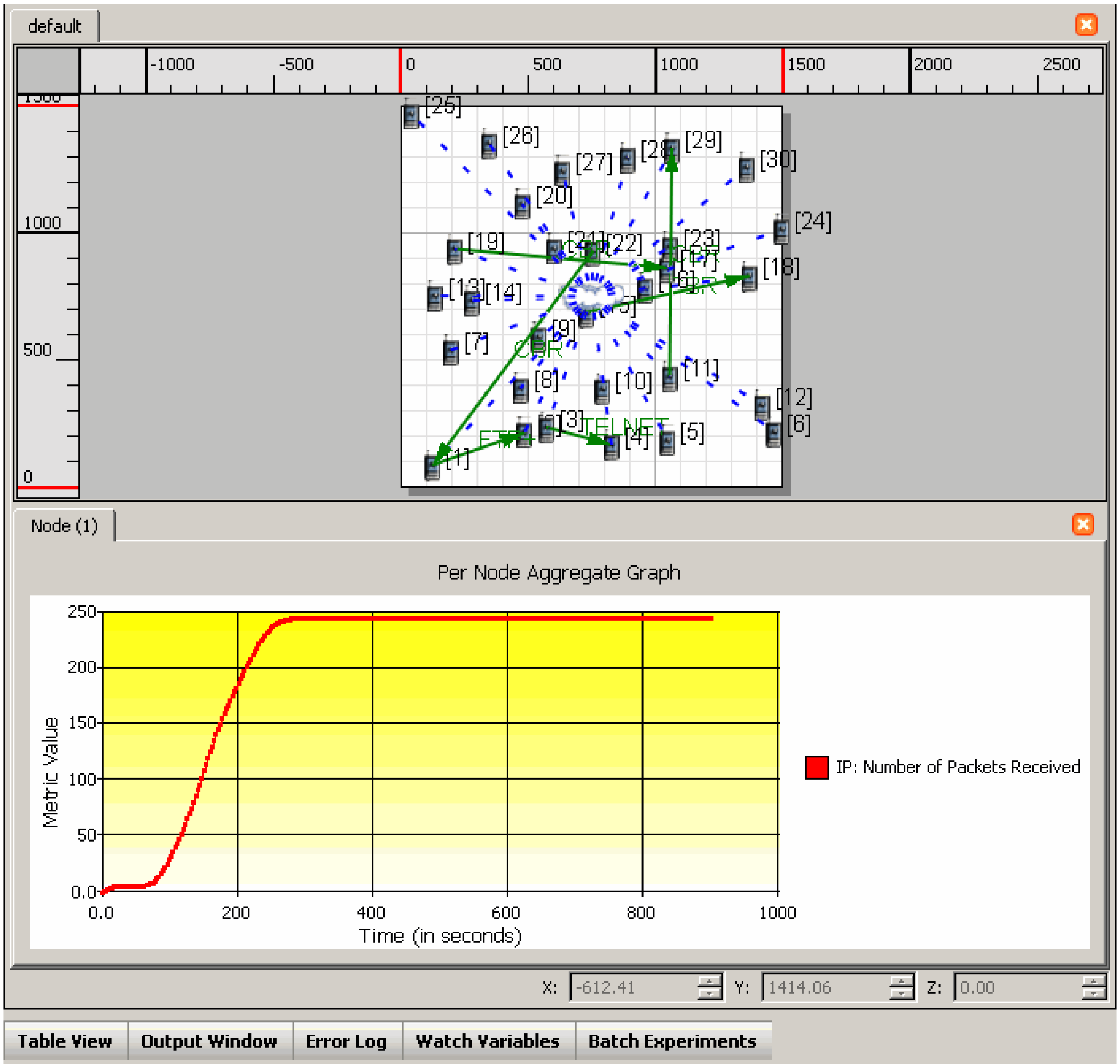


FIGURE 6-28. Dynamic Statistics for a Node

6.5.4.1.2 Queue Properties

To display the properties of a queue, select the desired queue in the **Runtime Analysis** panel. The bottom part of the **Runtime Analysis** panel displays the queue properties (see Figure 6-29). These properties are read-only and cannot be configured.

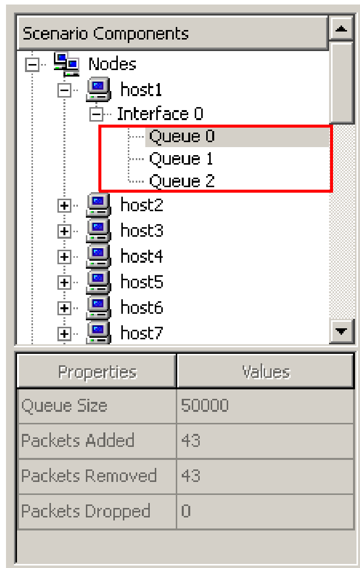


FIGURE 6-29. Queue Properties

Table 6-5 lists the queue properties that are displayed.

TABLE 6-5. Queue Properties

Property	Description
Queue Size	Maximum size of the queue, in bytes.
Packets Added	Total number of packets added to the queue.
Packets Removed	Total number of packets removed from the queue.
Packets Dropped	Total number of packets dropped from the queue.

6.5.4.1.3 Subnet Properties

To display the properties of a subnet, expand the **Subnets** group in the **Runtime Analysis** panel and navigate to the subnet. Selecting a subnet displays the subnet properties (see Figure 6-30).

The following commands are available if you right-click on the subnet in the **Runtime Analysis** panel:

- **Locate on Canvas**: Used to zoom to the subnet on the canvas

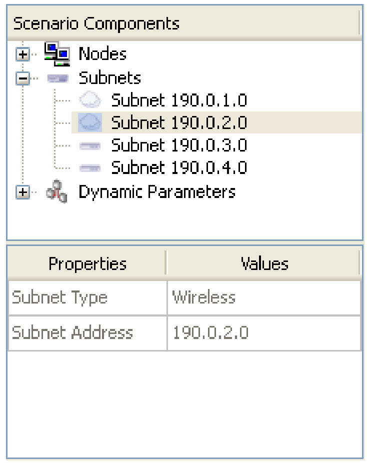


FIGURE 6-30. Subnet Properties

Table 6-6 lists the subnet properties that are displayed.

TABLE 6-6. Subnet Properties

Property	Description
Subnet Type	Type of the subnet (wired or wireless).
Subnet Address	IP address of the subnet.

6.5.4.2 Dynamic Parameters

Some models in EXata implement *dynamic* parameters to enable interaction with the simulator during run time. Values of dynamic parameters can be observed and/or modified during the simulation. Some dynamic parameters are read-only while other dynamic parameters can be modified by the user during the simulation. Changing the value of a parameter affects the behavior of the corresponding protocol model depending on how the parameter is used in that protocol model.

Values of dynamic parameters can be observed and/or modified from the **Runtime Analysis** panel.

Note: Dynamic parameters can be viewed in the **Runtime Analysis** panel only if dynamic parameters are enabled for the scenario (see Section 4.2.11.1).

Dynamic parameters are arranged in the **Runtime Analysis** panel in a tree view (see Figure 6-31) with the following three groups:

- **node:** All nodes in the scenario are listed in this group. Expanding the list view for a node lists all protocols/models running at the node that have dynamic parameters. The list also contains a group called **interface**.
  - Expanding the list view for a protocol/model lists all dynamic parameters for that protocol/model.
  - Expanding the list view for the **interface** group lists all interfaces for the node. Expanding the list view for a specific interface lists the interface-level properties and protocols/models running at the interface. Each protocol/model can be further expanded to show the dynamic parameters for that protocol/model.

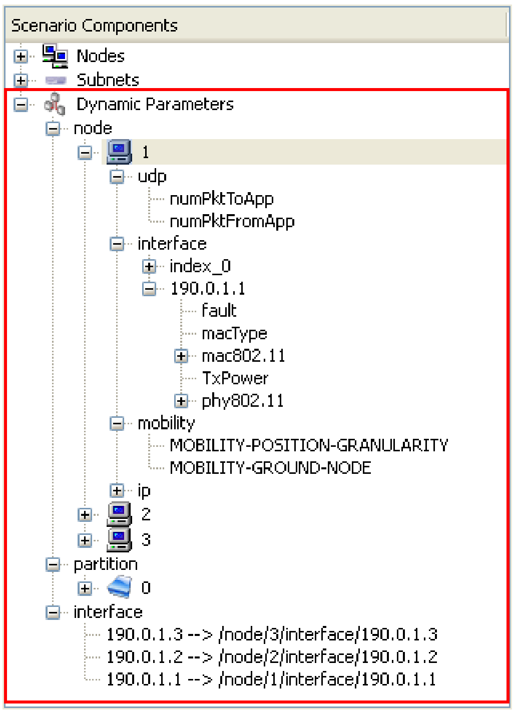


FIGURE 6-31. Dynamic Parameters

- **partition:** All partitions in the scenario are listed in this group. Expanding the list view for a partition lists the partition-level dynamic parameters for that partition.
- **interface**: All interfaces in the scenario are listed in this group. Each item in the list is a reference to an interface entry under a node. To access an interface's properties, right-click on an interface and select **Go to reference**. This expands the list view and highlights the referenced interface from which its properties can be accessed.

Selecting a dynamic parameter from the tree view displays the parameter and its value in the bottom part of the panel (see Figure 6-32). The options available for a parameter depend on the permissions (read, write, execute) associated with it. The current value of the parameter is displayed in the **Current Value** field. If write permission is associated with the parameter, then a **New Value** field is also displayed. To assign a new value to the parameter, enter the new value in the **New Value** field and click the  button.

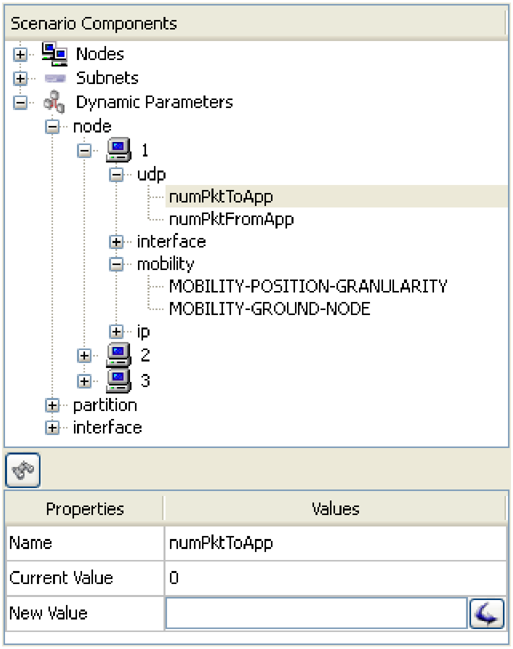


FIGURE 6-32. Dynamic Parameter with Write Permission

If execute permission is associated with the parameter, then an **Execute String** field is displayed. To execute a string, enter the string in the **Execute String** field and click the button. The result of the string execution is displayed in the **Execute Result** field (see Figure 6-33).

Note: The execute string feature is an advanced feature meant primarily for debugging.

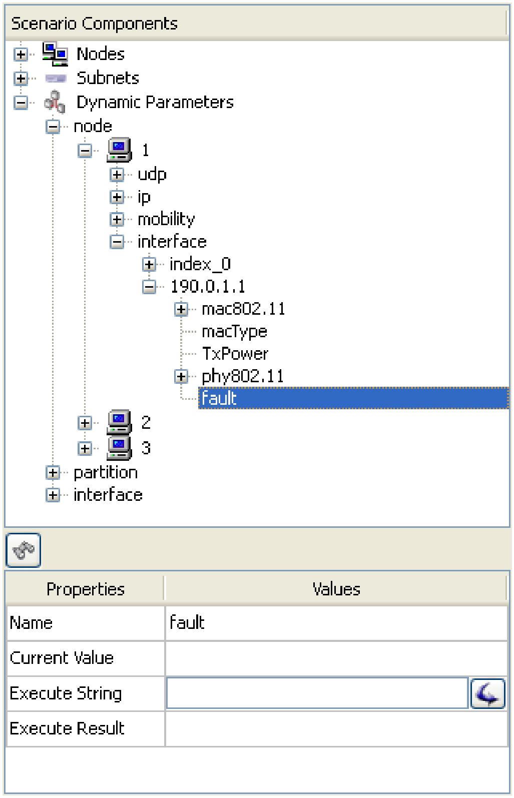


FIGURE 6-33. Dynamic Parameter with Execute Permission

You can also select one or more dynamic parameters and observe their values during the simulation in the **Watch Variables** panel. See Section 6.6.4 for details.

6.6 Bottom Panels

The following panels are available below the canvas:

- Table View
- Output Window
- Error Log
- Watch Variables
- Batch Experiments

Note: These panels occupy the same space and only one of them can be opened at any time. Any panel can be opened by clicking on it; by default all panels are closed.

6.6.1 Table View Panel

See Section 3.1.5.1 for details of the **Table View** panel.

6.6.2 Output Window Panel

Any message that simulator prints to the standard output is redirected to the output window (see Figure 6-34).

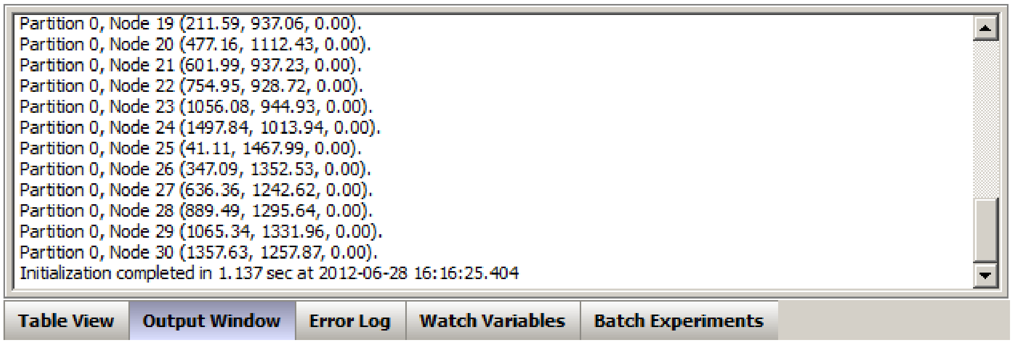


FIGURE 6-34. Output Window

6.6.3 Error Log Panel

See Section 3.1.5.3 for details of the **Error Log** panel.

6.6.4 Watch Variables Panel

The **Watch Variables** panel is used to watch values of selected dynamic parameters during a simulation. The selected dynamic parameters (watch variables) are displayed in a tabular format in the **Watch Variables** panel. The first column shows the path to the watch variable in the dynamic hierarchy, the second column displays the name of the variable, and the third column shows its current value.

Watch variables can be added to the table from the **Dynamic Parameters** tree in the **Runtime Analysis** panel (see Section 6.5.4.2). To add a watch variable to the table, do the following:

1. In the **Runtime Analysis** panel, expand the **Dynamic Parameters** tree and select the parameter to add to the table, as described in Section 6.5.4.2.
2. Click the **Add to Watch** button. This adds the parameter information (path, name, and current value) to the table in the **Watch Variables** panel.

The values of watch variables are continually updated as the simulation progresses.

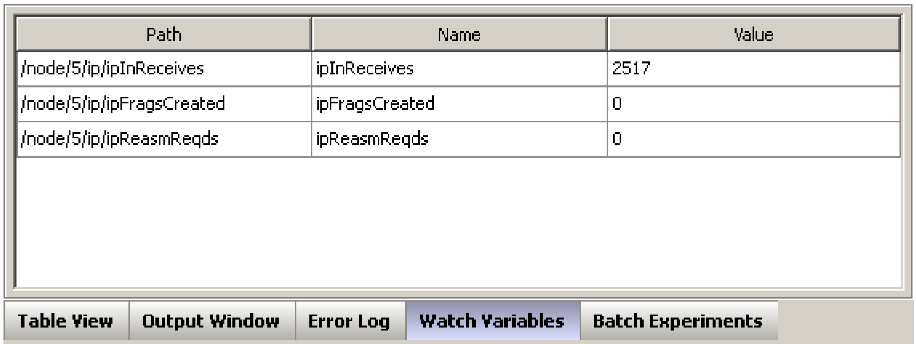


FIGURE 6-35. Watch Variables Panel

To remove a watch variable from the table, right-click on the variable in the table and select **Remove Selected from the menu**. You can also remove all variables from the table by selecting **Remove All**.

6.6.5 Batch Experiments Panel

See Section 3.1.5.5 for details of the **Batch Experiments** panel.

7 EXata Analyzer

Analyzer is a graphical tool for analyzing statistics generated by a simulation. When a simulation is run from the command line or from Architect, a statistics (.stat) file is created containing the simulation results. The statistics file is a text file which can be opened in any text editor. (See Section 2.3 for the syntax of the statistics file.) Analyzer parses the contents of a statistics file and presents the information graphically. This provides a convenient way to analyze simulation results. While the statistics file contains only per-node and per-interface statistics, Analyzer can also aggregate these metrics and show statistics for a node aggregated over all interfaces of the node and system-wide (scenario-level) statistics. In addition, statistics from two or more statistics files can be analyzed together to compare results from different experiments.

This chapter provides an overview of the various options, commands, and functionalities of the Analyzer component of EXata GUI. It also provides examples of analyzing statistics in EXata Analyzer.

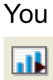
Analyzer features are described in the following sections:

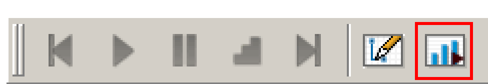
- Components of Analyzer: provides an overview of the components of Analyzer.
- Using Analyzer: describes how to analyze statistics using Analyzer.

How to Get to Analyzer

To switch to the Analyzer component of EXata GUI, click the 

button in the **Components** toolbar. No statistics files are opened automatically, but you can open and analyze the statistics file for any scenario.

You can also switch to Analyzer after running a scenario in Architect by clicking the **Analyze Statistics**  button. In this case, the statistics file for the scenario that was run in Architect is automatically opened in Analyzer.



7.1 Components of Analyzer

This section provides an overview of the different components of Analyzer.

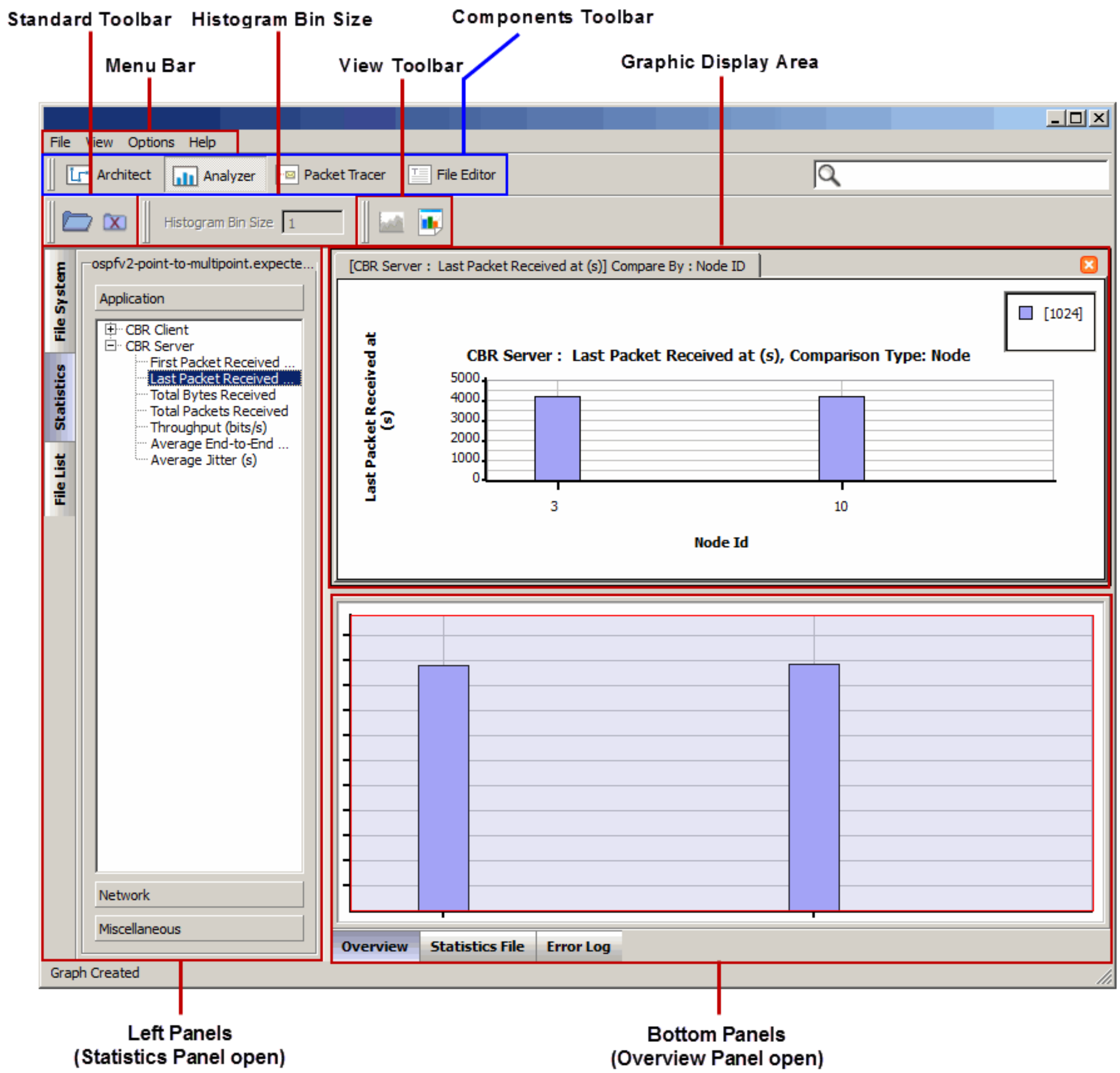


FIGURE 7-1. Analyzer Layout

7.1.1 Menu Bar

This section explains the commands available from the **Menu** bar.

7.1.1.1 File Menu

The **File** menu provides the following commands for file operations.

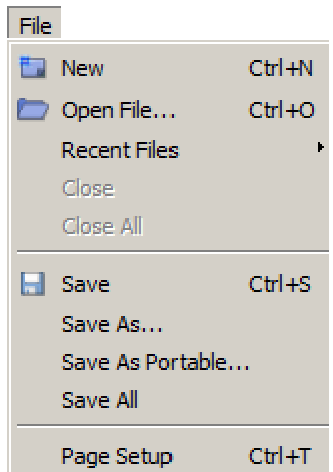


FIGURE 7-2. File Menu

Command	Description
Open	Opens a file in Analyzer. The file is added to the list of open files in the **File List** panel. If the file contains any statistics, they are listed in the **Statistics** panel.
Recent Files	Displays a list of recently opened statistics files. Selecting a file from this list opens it in Analyzer.
Close Graph	Displays a list of all graphs that are open in the current statistics file. Selecting a graph from the list closes it. Selecting **Close All Graphs** closes all graphs in the current statistics file.
Close	Closes the current statistics file.
Close All	Closes all open statistics files.
Export	Exports the graph data to a text file or saves the graph as an image file. The options are: • **Current Graph to Delimited Text**: Saves the data from the selected graph to a delimited text file. • **All Opened Graphs to Delimited Text**: Saves the data from all open graphs to a delimited text file. • **All Graphs to Delimited Text**: Saves the data from all metrics (whether or not they are displayed as graphs) of the selected protocol to a text file. (A protocol is selected by clicking on the name of the protocol or any metric of the protocol in the **Statistics** panel.) • **As Image**: Saves the selected graph as an image file (in png, jpg, bmp, ppm, tif, xbm, orxpm format). When saving data to a text file, selecting **Row Wise** exports data in a row-wise alignment format, and selecting **Column Wise** exports a data in a column-wise alignment format. **Note:** The **Export** option is enabled only if at least one graph is displayed.

Command	Description
Page Setup	Opens a dialog box to set printing options.
Print	Prints a hard copy of the displayed graph.
Exit	Exits from the EXata application. If there are any unsaved changes in any of the open scenarios, the user is prompted to save them.

7.1.1.2 View Menu

The **View** menu provides the commands to set the display settings.

FIGURE 7-3. View Menu

Command	Description
Graph Visibility	Option to plot a single graph or multiple graphs. • **Single Graph**: Only one graph is opened at a time. When you open another graph, the existing graph will be closed and the new graph will be opened. • **Multiple Graph**: More than one graph can be opened. Each graph is opened in a separate tab.
Node Name **Node ID**	Option to display node names or node IDs on the X-axis. By default, node ID's are displayed on the X-axis. The **Node Name** option is enabled only if host statistics are enabled for any node in the scenario (see Section 4.2.9). If **Node Name** option is enabled and is selected, then the option changes to **Node ID**. This menu option performs the same function as the **Node Name** button in the **View** toolbar. See Section 7.1.2.2.
Graph Properties	Displays the **Graph Properties Dialog** box where you can configure the following display properties: • **Graph Background**: Provides options to change the background color and shading of a graph. • **View Options**: When selected, shows the grid and legend on the graph, and provides options to set the bar shading and log scale. • **Legend Options**: Change the legend background and the bar colors from this tab.

7.1.1.3 Options Menu

The **Options** menu allows you to select the graph type, compare statistics by IP address or node ID, and set the graph display properties.

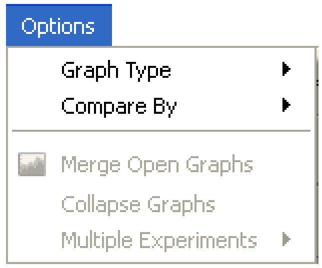


FIGURE 7-4. Options Menu

Command	Description
Graph Type	Select the graph type: • **Bar Graph**: Shows a bar graph comparing per-node or per-interface statistics. • **Histogram**: Shows a summary of system-wide statistics falling within various ranges. **Note:** Changing the graph type does not change the graphs that are already plotted. Graphs plotted after the change are drawn with the new option.
Compare By	Select the type of statistics (node level or interface level) to be plotted. • **Node ID**: Node-level statistics are plotted. If a node has multiple interfaces, the sum of the metric for all interfaces at the node is plotted. • **IP Address**: Interface-level statistics are plotted. If a node has multiple interfaces, the metric for each interface is plotted separately. **Note:** Changing this option closes all open graphs. Graphs plotted after the change are drawn with the new option.
Merge Open Graphs	Merge all open graphs into a single graph and show the merged graph in a new tab. If the graphs are bar graphs, the generated merged graph is a line graph. If the source graphs are histograms, the generated merged graph is a histogram. This option is enabled only if all open graphs are of the same type (bar or histogram). This menu option performs the same function as the **Merge Graph** button in the **View** toolbar. See Section 7.1.2.2.
Collapse Graphs	Display only metrics that have a non-zero value. This is useful to hide metric values that are zero for a large number of nodes. This option is enabled only for bar graphs.
Multiple Experiments	Compare metrics from different statistics files. • **Display Metric Sum**: Display the sum of the values of the selected metric for each node or interface for all files. • **Display Metric Average**: Displays the average value of the selected metric for each node or interface for all files.

7.1.1.4 Help Menu

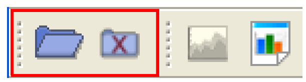
This is the same as the **Help** menu described in the components of Architect. See Section 3.1.1.6 for a description of the **Help** menu.

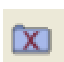
7.1.2 Toolbars

This section describes the toolbars available in Analyzer.

7.1.2.1 Standard Toolbar

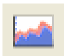
The **Standard** toolbar is used to open and close files. The following table describes the buttons of the **Standard** toolbar.



Button	Function	Description
	**Open**	Opens a file in Analyzer. The file is added to the list of open files in the **File List** panel. If the file contains any statistics, they are listed in the **Statistics** panel. This button performs the same function as the **File > Open** command. See Section 7.1.1.1.
	**Close**	Closes the current statistics file. This button performs the same function as the **File > Close** command. See Section 7.1.1.1.

7.1.2.2 View Toolbar

The **View** toolbar is used to merge graphs and display node names. The following table describes the buttons of the **View** toolbar.

Button	Function	Description
	**Merge Open Graphs**	Merges all open graphs into a new composite (merged) graph. If the individual graphs are bar graphs, the merged graph is a line graph. If the individual graphs are histograms, the merged graph is a histogram. This button is enabled only when multiple graphs are open and all graphs are of the same type (either bar graph or histogram). This functionality is also available from the **Options > Merge Open Graphs** menu option. See Section 7.1.1.2.
	**Graph Properties**	Displays the **Graph Properties Dialog** box where you can configure the following display properties: • **Graph Background**: Provides options to change the background color and shading of a graph. • **View Options**: When selected, shows the grid and legend on the graph, and provides options to set the bar shading and log scale. • **Legend Options**: Change the legend background and the bar colors from this tab. **Note:** This function is also available from **View > Graph Properties** menu option. See Section 7.2.5.

7.1.2.3 Histogram Bin Size Toolbar

The **Histogram Bin Size** toolbar allows you enter a new bin size. This option is enabled only if the active graph is opened as a histogram (i.e., **Options > Graph Type > Histogram** is selected before plotting the graph). To apply a new bin width, enter the new value and press the **Return** key.

FIGURE 7-5. Histogram Bin Size Toolbar

7.1.3 Left Panels

The following panels are available to the left of the Graph Display:

- File System
- Statistics
- File List

Note: These three panels occupy the same space and at most one of them can be open at any time.

7.1.3.1 File System Panel

This is the same as the **File System** panel of Architect. See Section 3.1.3.1 for a description of the **File System** panel.

7.1.3.2 Statistics Panel

The **Statistics** panel lists all statistics in the selected statistics (.stat) file.

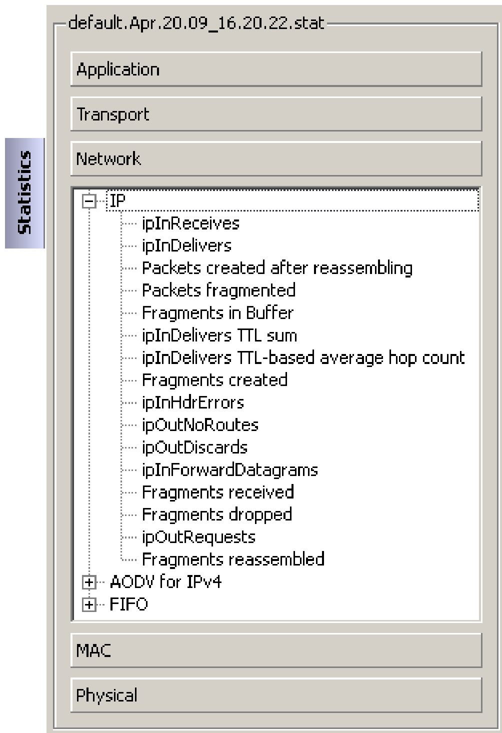


FIGURE 7-6. Statistics Panel

The statistics are grouped together by layers. Clicking on a layer button (**Application**, **Transport**, **Network**, **MAC**, or **Physical**) displays the list of all protocols at that layer for which there are statistics in the statistics (.stat) file. The statistics for a protocol are displayed as a list by expanding the plus sign next to the protocol's name.

7.1.3.3 File List Panel

The **File List** panel lists all statistics (.stat) files that are open in Analyzer.

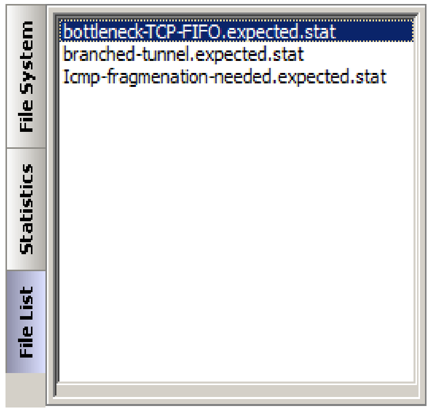


FIGURE 7-7. File List Panel

To list the statistics in a file, double-click on the file name in the **File List** panel. This opens the **Statistics** panel, which lists the statistics of the selected file.

To close a statistics file, right-click on the file name in the **File List** panel and select **Close File**. (You can also close a file from the **File** menu.)

7.1.4 Bottom Panels

The following panels are available below the Graph Display area:

- Overview
- Statistics File
- Error Log

Note: These three panels occupy the same space and at most one of them can be open at any time.

7.1.4.1 Overview Panel

The **Overview** panel displays a graph for the selected statistic for the entire scenario. You can zoom into an area by resizing and moving the red rectangle over the desired area. The selected area of the graph is displayed in the Graph Display area above.

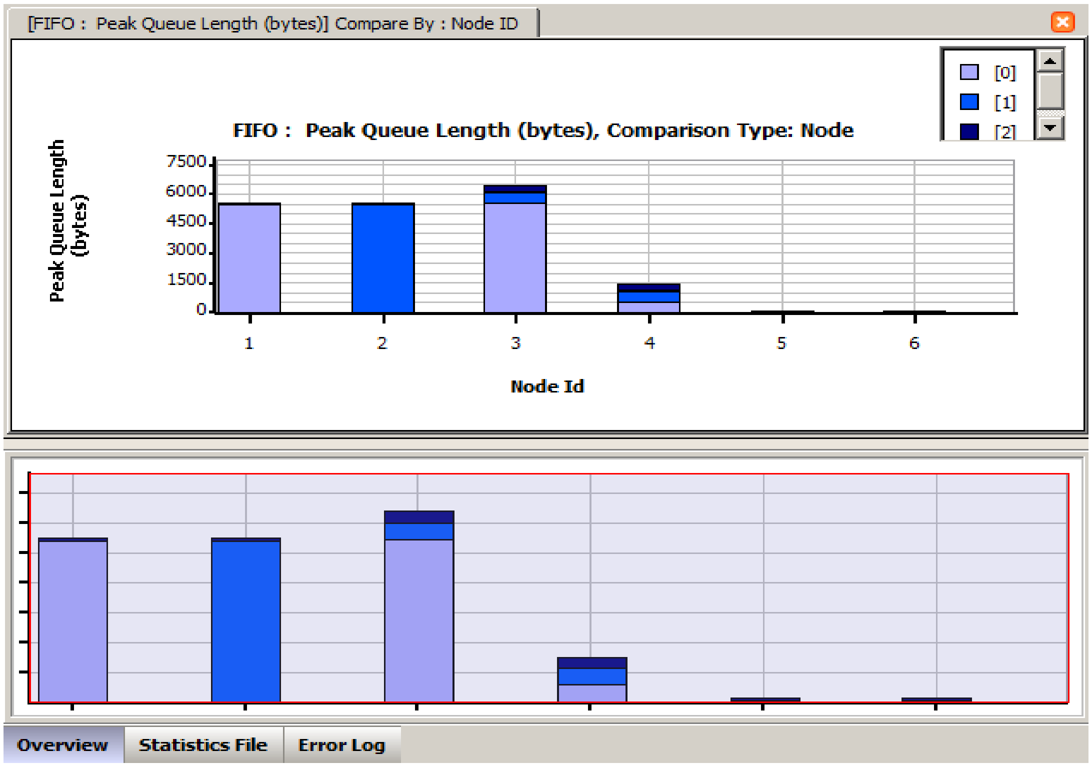


FIGURE 7-8. Overview Panel

7.1.4.2 Statistics File Panel

The **Statistics File** panel displays the selected statistics (.stat) file as a text file.

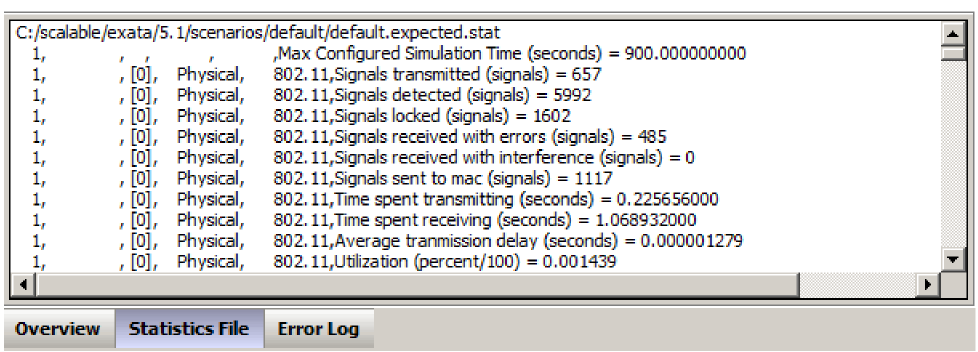


FIGURE 7-9. Statistics File Panel

7.1.4.3 Error Log Panel

The **Error Log** panel displays any errors encountered in reading the selected statistics (.stat) file.

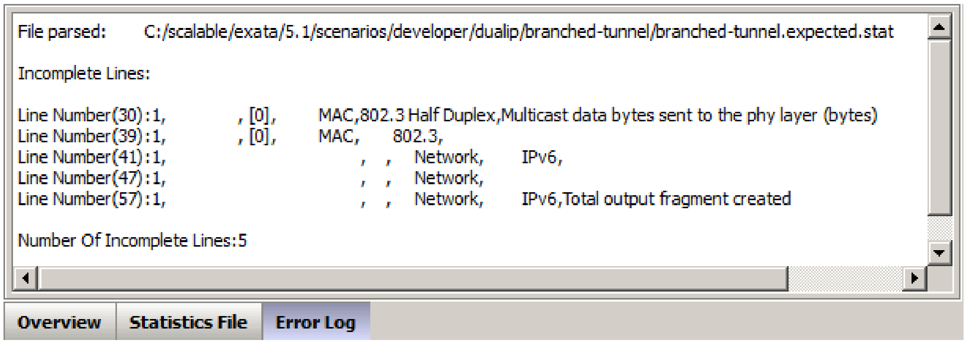


FIGURE 7-10. Error Log Panel

7.2 Using Analyzer

This section describes how to analyze simulation results in Analyzer. It is assumed that one or more statistics(.stat) files have been generated by running a scenario from the command line or in Architect.

- Section 7.2.1 describes how to analyze per-node statistics for a single experiment.
- Section 7.2.2 describes how to analyze system-wide statistics for a single experiment.
- Section 7.2.3 describes how to analyze statistics for a single experiment but viewing statistics for interfaces separately instead of aggregating them at the node level.
- Section 7.2.4 describes how to compare statistics from multiple experiments.
- Section 7.2.5 describes how to change the appearance of a graph.

To illustrate the use of Analyzer, we will use two statistics files that are included in the EXata distribution. These files can be found in EXATA_HOME/scenarios/developer/tcp/bottleneck-TCP. This folder contains two scenario configuration files (bottleneck-TCP-FIFO.config and bottleneck-TCP-RED.config) and the statistics files generated by running these scenarios (bottleneck-TCP-FIFO-expected.stat and bottleneck-TCP-RED-expected.stat, respectively). These scenarios are identical except that one uses FIFO queues and the other uses RED queues. The README file in this folder describes the scenario.

7.2.1 Analyzing Per-node Statistics for a Single Experiment

To analyze per-node statistics from a single experiment, perform the following steps:

1. Open the statistics file to analyze. You can open a statistics file from the **File** menu, **Standard** Toolbar or **File System** panel. (To open a statistics file from the **File System** panel, double-click on the file name or right click on the file name and select **Analyze**.)

For this example, open the file bottleneck-TCP-FIFO-expected.stat in the directory EXATA_HOME/ scenarios/developer/tcp/bottleneck-TCP.

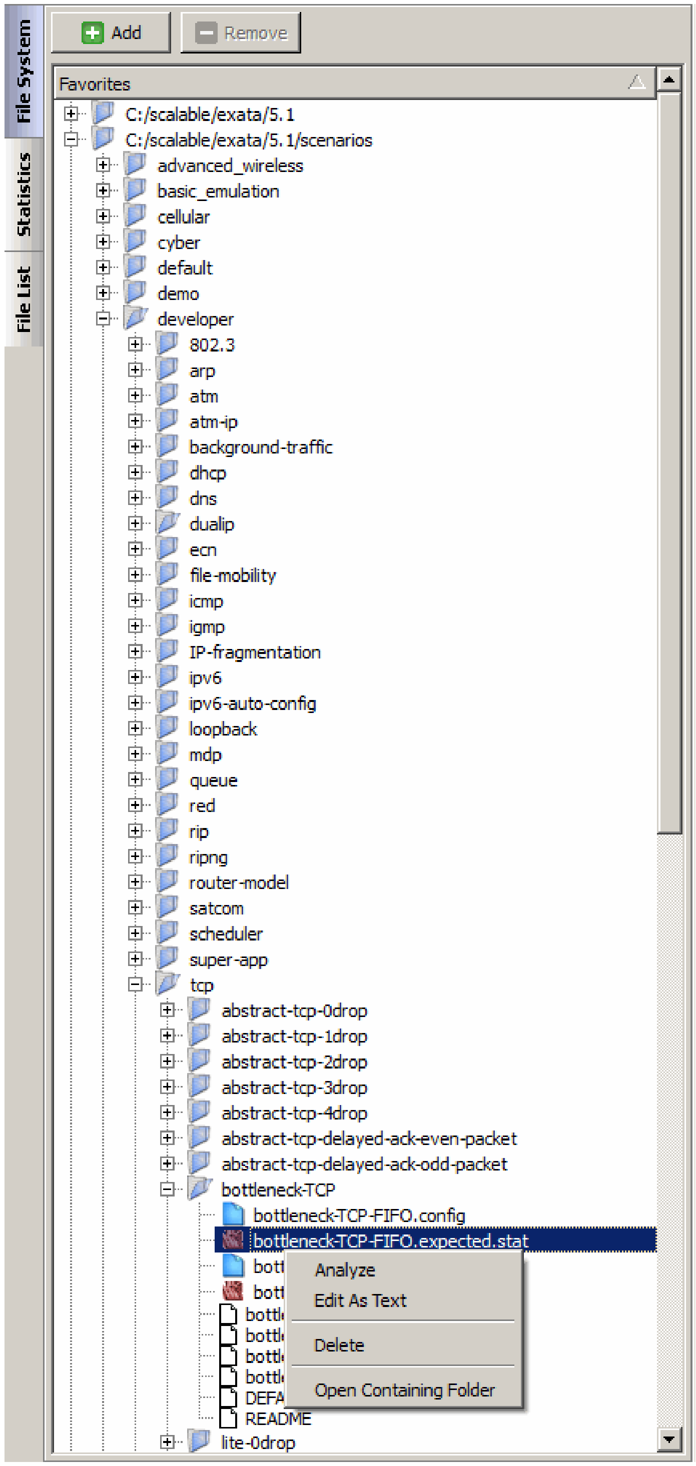


FIGURE 7-11. Opening a Statistics File from File System

2. Per-node statistics are displayed as bar graphs. Go to **Options > Graph Type** and select **Bar Graph**.

3. The layers for which there are statistics in the file are displayed in the **Statistics** panel. For this example, the file contains statistics for the Application, Transport, Network, and MAC layers. There are no Physical layer statistics in this file.

   Click the **Network** button. This displays all protocols/models at the Network layer for which there are statistics in the file. Click on the plus sign before “FIFO”. This displays all statistics collected for the FIFO model (see Figure 7-12).

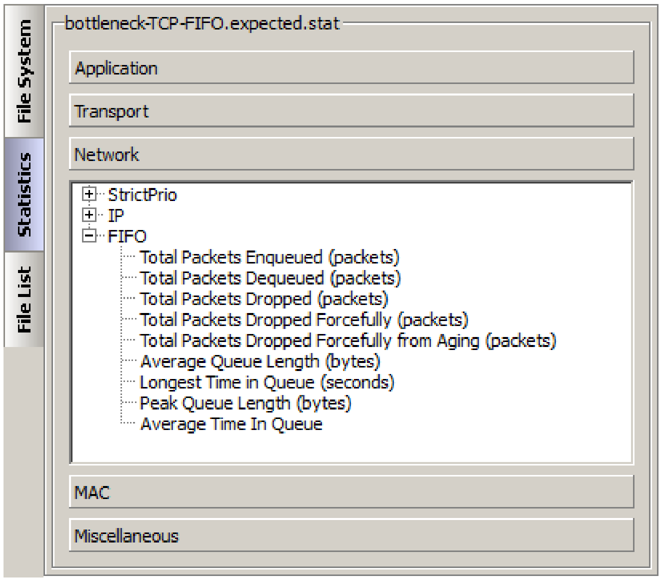


FIGURE 7-12. Statistics for a Specific Model

4. To plot the graph for a statistic, click on the statistic name in the **Statistics** panel. A bar graph is displayed in the Graph Display area as well as in the **Overview** panel.

   You can close the **Overview** panel to increase the size of the graph in the Graph Display area.

   You can also select an area of the graph to zoom in by resizing and moving the red rectangle in the **Overview** panel. Only the area of the graph covered by the red rectangle is displayed in the Graph Display area. This allows you to examine a region of the graph in more detail.

   Figure 7-13 shows the bar graph for FIFO: Peak Queue Length. In the **Overview** panel, statistics for all nodes are displayed. A region of the graph corresponding to nodes 4, 5, and 6 has been selected by the red rectangle. This selected region is displayed in the Graph Display area.

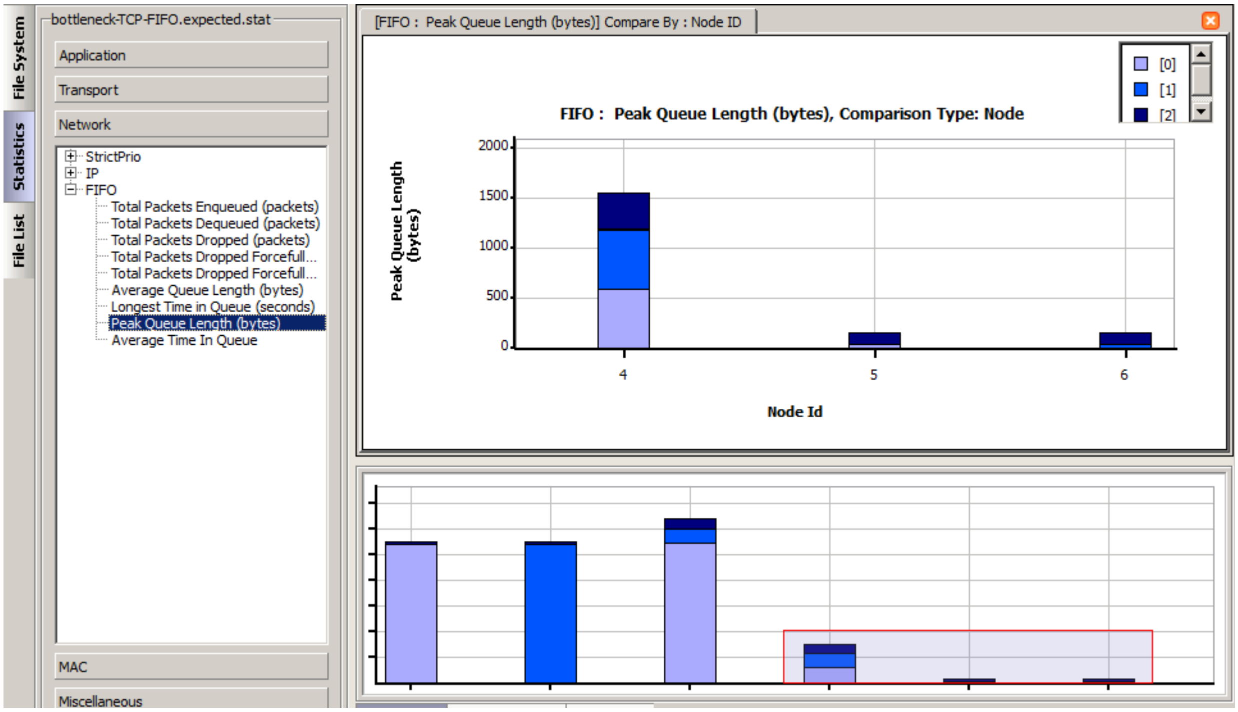


FIGURE 7-13. Bar Graph for a Single Statistic

The bar graph shows the metric for each instance (in this case, each instance corresponds to a queue). In this example, there are three queues at each node interface. The metrics for a node are plotted as a single bar with three colored sections. The different colors in a bar correspond to the metric value for different instances. The legend in the top right corner of the graph shows the color used for each instance.

Note: The value plotted is the sum of the metric at all interfaces of the node. To plot the metric for each interface separately, plot the graph by IP addresses, as explained in Section 7.2.3.

Placing the mouse over a section of a bar displays the metric value for that section (i.e., the metric value of the instance corresponding to that section) and the total of the metrics for all instances at the node. For example, Figure 7-13 shows that the peak queue size for queue 1 of node 4 is 596 bytes and the total of peak queue sizes for all queues of node 4 is 1552 bytes.

By default, only one graph can be displayed at a time. If you select another statistic from the **Statistics** panel, the currently open graph is closed and the graph for the selected statistic is displayed.

5. To plot another statistic without closing the current graph, go to **View > Graph Visibility** and select **Multiple Graph**. This opens each selected statistic in a separate tab in the Graph Display area.

   Figure 7-14 shows the tabs for three statistics. The name of the model and the statistic are displayed in the tab.

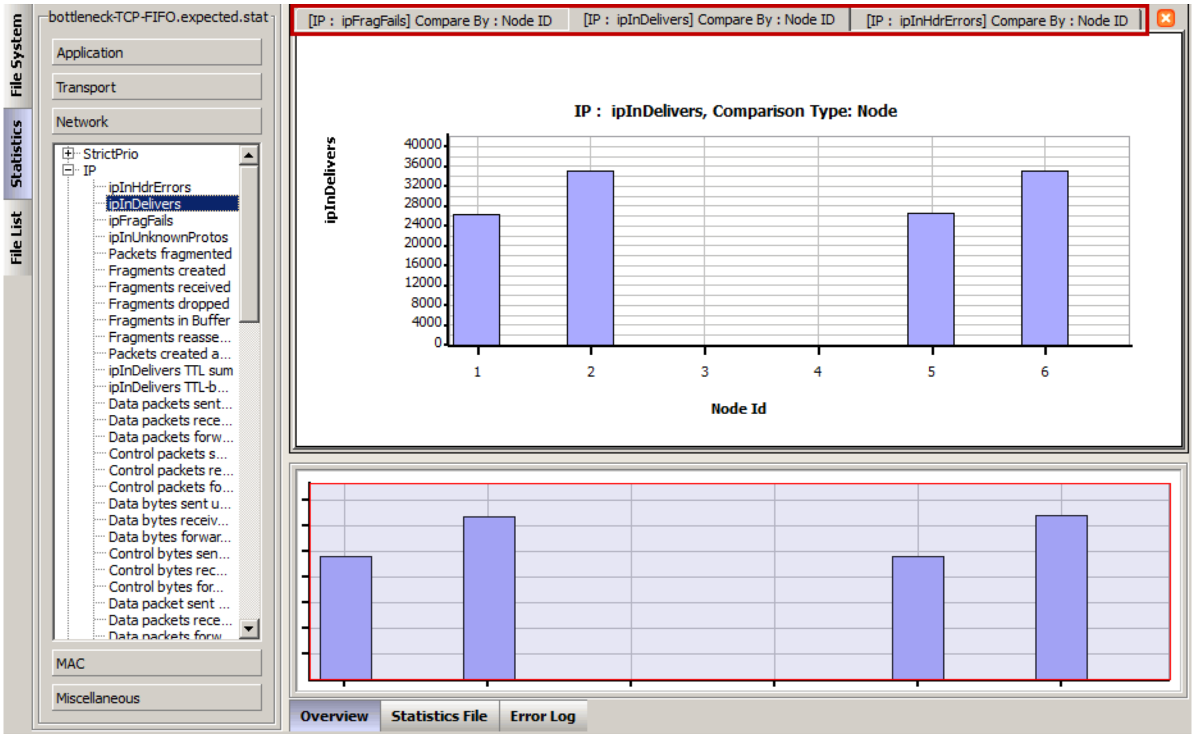


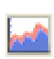
FIGURE 7-14. Multiple Graphs in Separate Tabs

6. You can plot multiple statistics in a single graph by merging the graphs for individual statistics as follows:

   a. Close all open graphs.

   a. Enable the multiple graph option by selecting **View > Graph Visibility > Multiple Graph**.

   b. Select the statistics to plot one at a time. For example, select FIFO: Peak Queue Length and FIFO: Average Queue Length. The graphs for the selected statistics are displayed in separate tabs in the Graph Display area.

   c. Click the **Merged Graph**  button in the **View** toolbar or select **Options > Merge Open Graphs**. This creates a composite graph in a new tab. The statistics from all open graphs are plotted in the composite (merged) graph.

Notes: **1.** Graphs can be merged only if all open graphs are of the same type (bar graph or histogram). If the individual graphs are bar graphs, the merged graph is a line graph. If the individual graphs are histograms, the merged graph is a histogram.

2. The total of the metric values for all instances are plotted in the merged graph. Per-instance values are not plotted.

Figure 7-14 shows the merged graph for FIFO: Peak Queue Size, and FIFO: Average Queue Length.

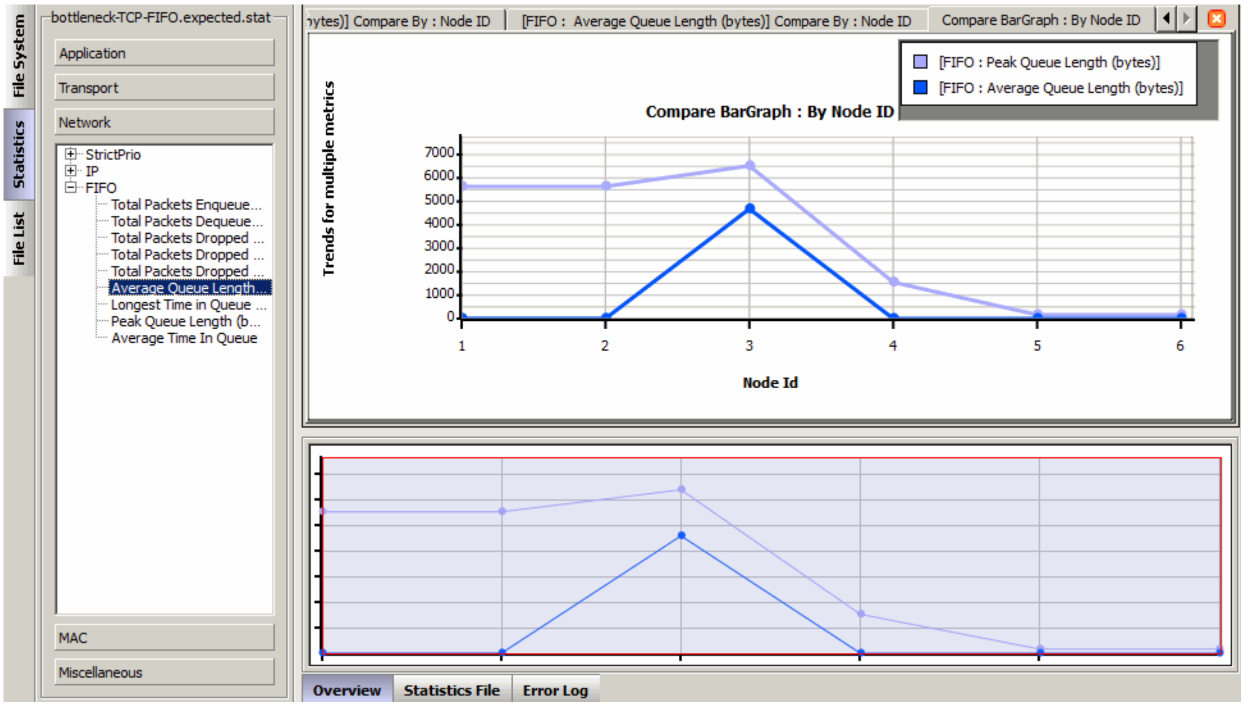


FIGURE 7-15. Merged Graph Showing Multiple Statistics

7.2.2 Analyzing Scenario Statistics for a Single Experiment

Scenario (system-wide) statistics are plotted as histograms. Scenario statistics can be viewed in the same way as per-node statistics (see Section 7.2.1), except that the graph type must be histogram.

In the following example, we will first show the per-node graph for a statistic (from a single experiment) and then plot the scenario graph for the same statistic.

1. Open the file bottleneck-TCP-FIFO-expected.stat, as described in Section 7.2.1.
2. Go to **Options > Graph Type** and select **Bar Graph**.
3. In the **Statistics** panel, click the **Transport** button and select TCP: Unicast average delay at the transport layer. Figure 7-16 shows the graph that is displayed.

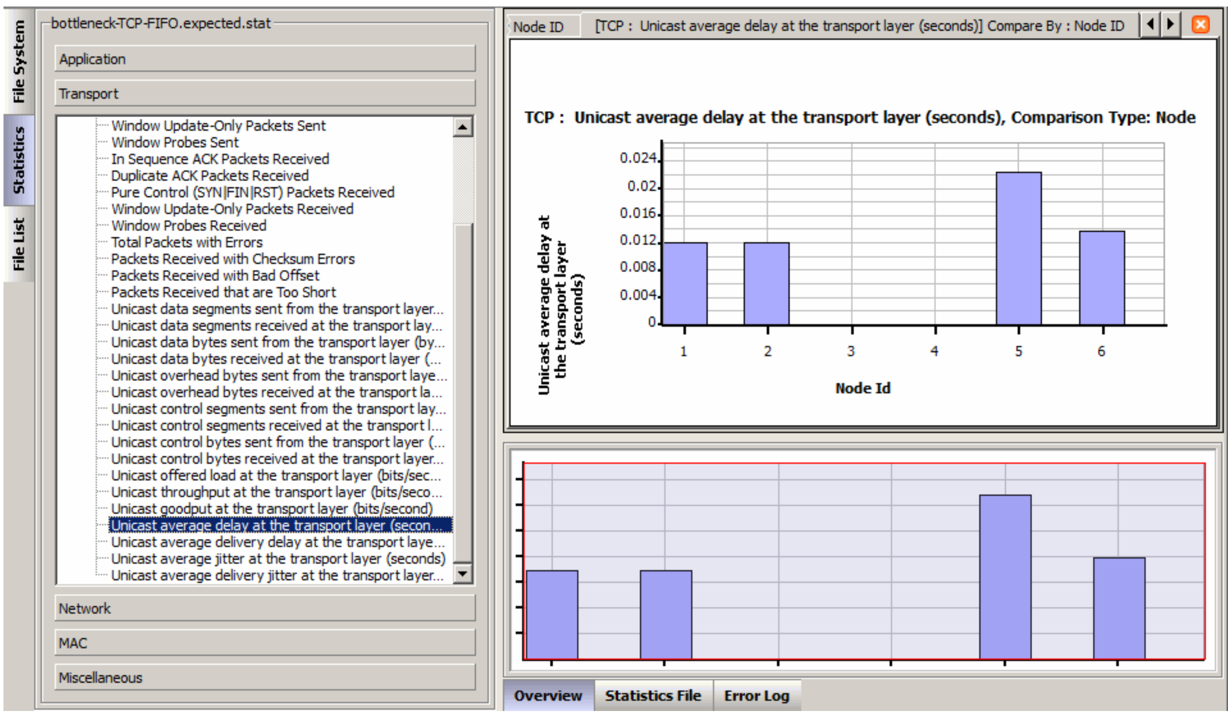


FIGURE 7-16. Per-node TCP Statistics

This graph plots the TCP statistic for each node.

4. Go to **Options > Graph Type** and select **Histogram**.
5. In the **Statistics** panel, click the **Transport** button and select TCP: Unicast average delay at the transport layer.
6. Change the histogram bin size to an appropriate value. To change the histogram bin size, enter a value in **Histogram Bin Size** and press the **Return** key. Figure 7-17 shows the graph that is displayed for a bin size of 0.015.

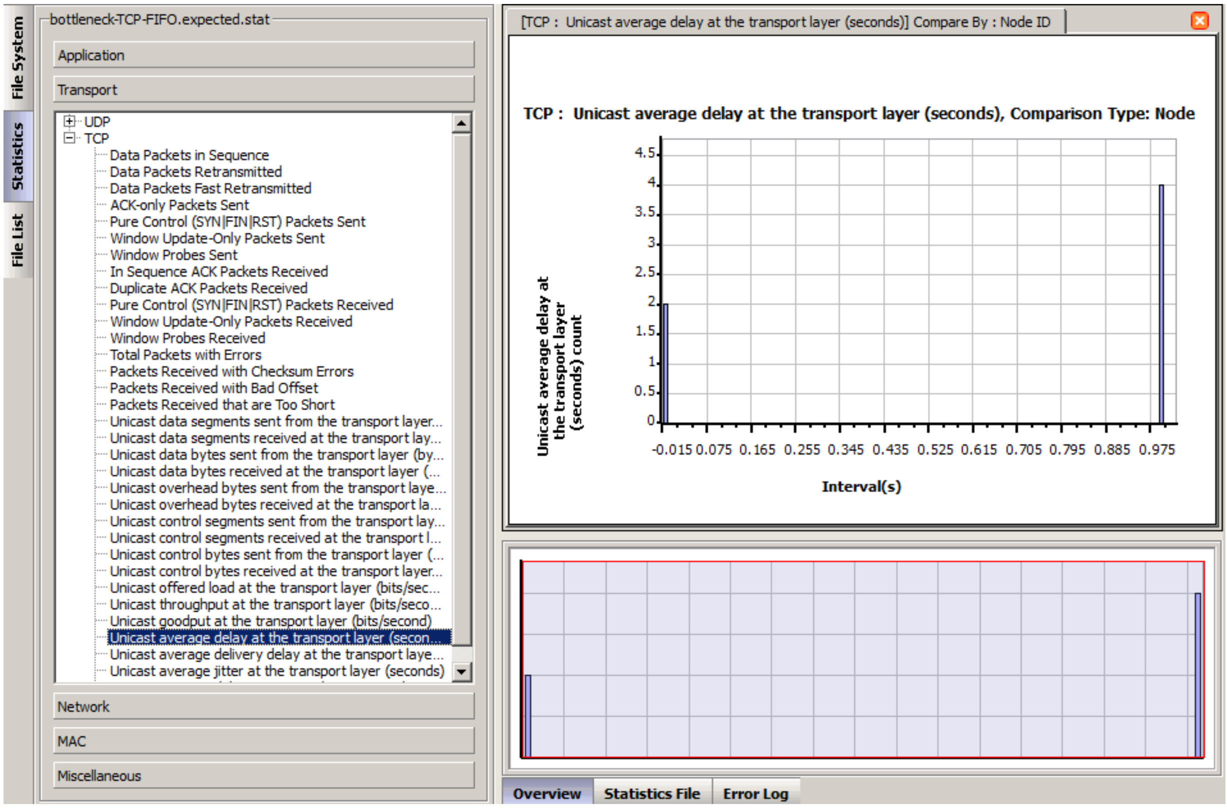


FIGURE 7-17. System-wide TCP Statistics

This graph plots the TCP statistic for the entire scenario.

7.2.3 Analyzing Statistics by Interface

When statistics are analyzed by node (by selecting the **Options > Compare By > Node** option), the sum of the metric value for all interfaces of the node is plotted. When statistics are analyzed by interface (by selecting the **Options > Compare By > IP Address** option), metric values for interfaces are plotted separately rather than as a sum.

In the following example, we will first plot a statistic by node (aggregated for all interfaces at a node) and then plot the same statistic separately for each interface.

1. Open the file bottleneck-TCP-FIFO-expected.stat, as described in Section 7.2.1.
2. Go to **Options > Graph Type** and select **Bar Graph**.
3. Go to **Options > Compare By** and select **Node ID**.

   All statistics in the statistics file are listed in the **Statistics** panel.
4. In the **Statistics** panel, click the **Network** button and select FIFO: Total Packets Enqueued. Figure 7-18 shows the graph that is displayed.

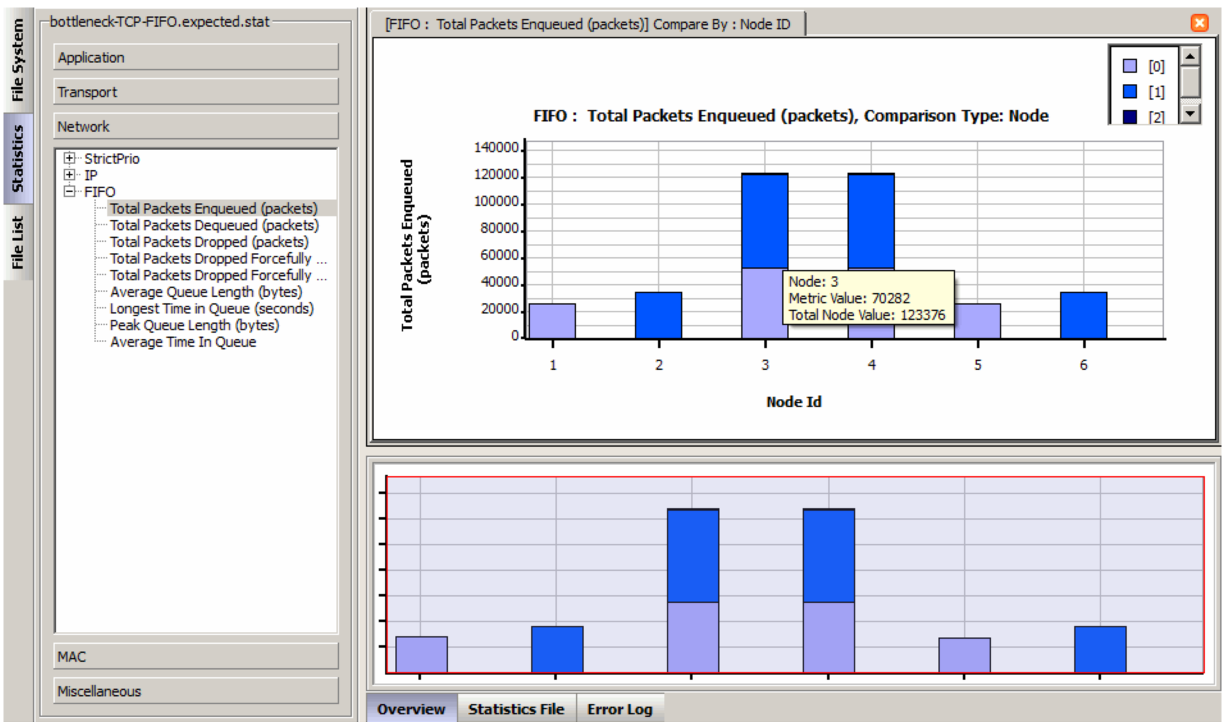


FIGURE 7-18. Queue Statistics by Node

For each node, the graph plots (for each queue instance) the sum of Total Packets Enqueued at all interfaces of the node. For example, for node 3, the sum of Total Packets Enqueued for queue 1 at all interfaces is 70282, and the sum of Total Packets Enqueued for all queues at all interfaces is123376. The node IDs are displayed along the X-axis.

5. Go to **Options > Compare By** and select **IP Address**.

   Now only those statistics are listed in the **Statistics** panel that have an IP address associated with them in the statistics file (see Section 2.3). Note that only the **Network** button is visible and within the Network layer, IP statistics are not listed any more.

6. In the **Statistics** panel, click the **Network** button and select FIFO: Total Packets Enqueued. Figure 7-19 shows the graph that is displayed.

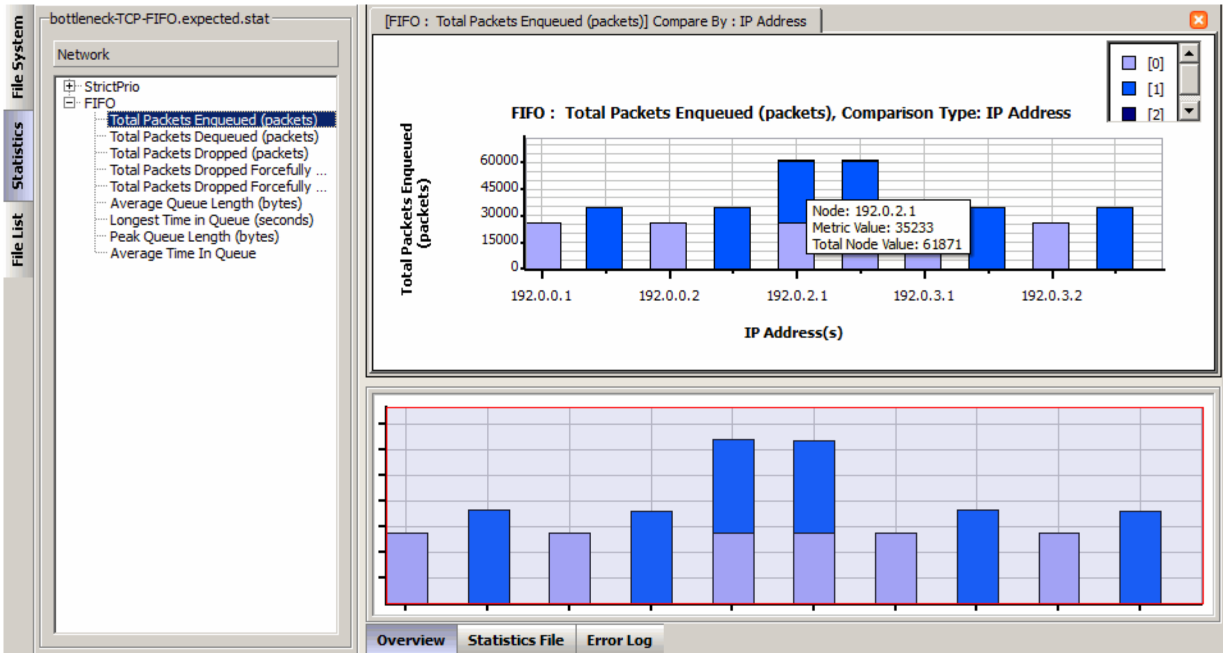


FIGURE 7-19. Queue Statistics by Interface

The graph plots (for each queue instance) Total Packets Queued at each interface. The IP addresses of the interfaces are displayed along the X-axis. In this example, the IP addresses of the interfaces at each node are:

Node 1: 192.0.0.1

Node 2: 192.0.1.1

Node 3: 192.0.0.2, 192.0.1.2, and 192.0.2.1

Node 4: 192.0.2.1, 192.0.3.1, and 192.0.4.1

Node 5: 192.0.3.2

Node 6: 192.0.4.2

For the third interface of node 3 (with IP address 192.0.2.1), Total Packets Queued for queue 1 is 35233 and the sum of Total Packets Queued for all three queues is 61871.

Total Packets Queued for queue 1 at the first interface of node 3 (with IP address 192.0.0.2) is 0, and Total Packets Queued for queue 1 at the second interface of node 3 (with IP address 192.0.1.2) is 35049. (Thus, the sum of Total Packets Queued for queue 1 at all three interfaces of node 3 is 70282, as plotted in Figure 7-18.)

7.2.4 Analyzing Statistics from Multiple Experiments

To analyze statistics from multiple experiments simultaneously, perform the following steps:

1. In the **File System** panel, select the files to analyze, right click and select **Analyze**. (To select multiple files, press the **Ctrl** key and click on the file names).

   For this example, open the files bottleneck-TCP-FIFO-expected.stat and bottleneck-TCP-RED-expected.stat in the directory EXATA_HOME/scenarios/developer/tcp/bottleneck-TCP.

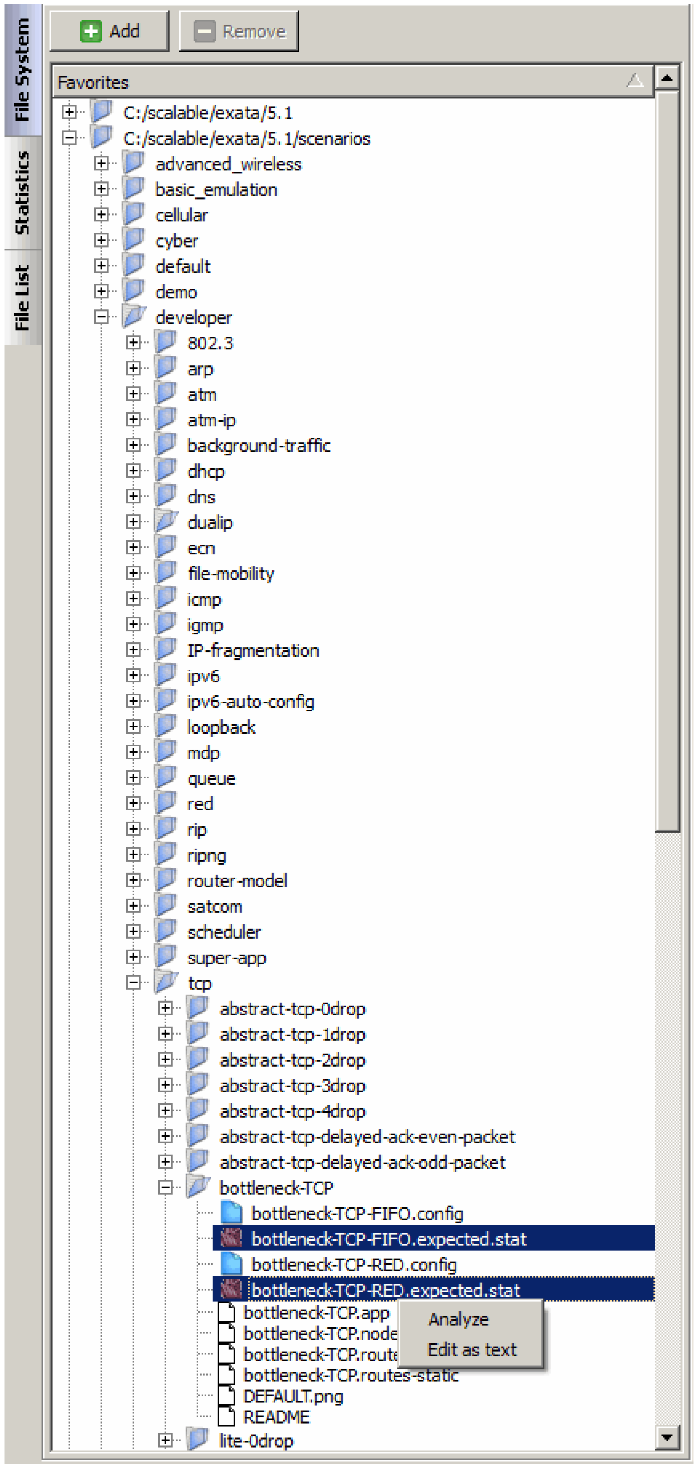


FIGURE 7-20. Opening Multiple Statistics Files

2. Statistics from the selected files can be analyzed in the same way as statistics from a single experiment (see for Section 7.2.1, Section 7.2.2, and Section 7.2.3). Statistics from all selected files are plotted in the same graph.

   For our example, in the **Statistics** panel, click the **Application** button and select Gen/FTP Server: Throughput.

   Figure 7-21 shows the graph that is displayed. (A region of the graph has been selected in the **Overview** panel to conveniently compare the statistics.)

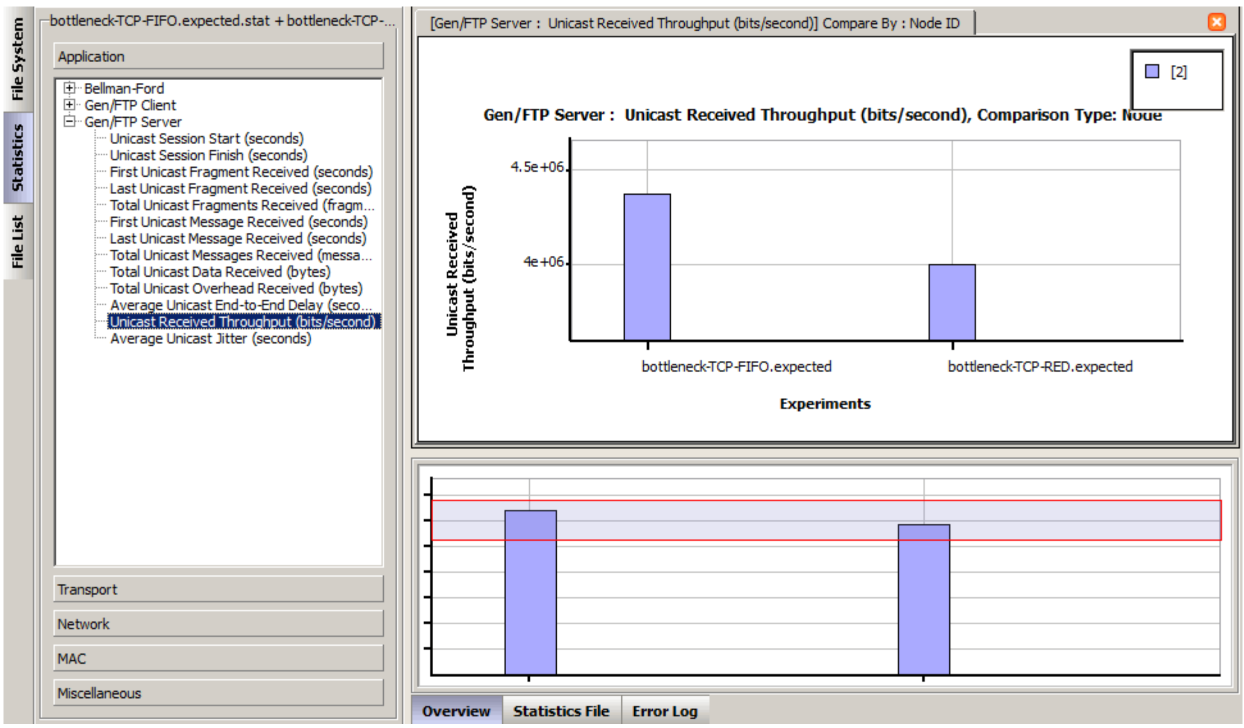


FIGURE 7-21. Statistics from Multiple Experiments

For multiple experiments, the statistics that are displayed depend on both whether comparison is done by node ID or IP address (by selecting **Options > Compare By > Node ID** or **Options > Compare By > IP Address**) and whether the option to display the sum or average is selected (by selecting **Options > Multiple Experiments > Display Metric Sum** or **Options > Multiple Experiments > Compare Metric Average**).

- If the option to display the sum is selected, then the statistic displayed is the sum of the statistics for all nodes (whether the comparison is done by node ID or IP address).
- If the option to display the average is selected and the comparison is done by node ID, the statistic displayed is the average of the statistics over all nodes.
- If the option to display the average is selected and the comparison is done by IP address, the statistic displayed is the average of the statistics over all interfaces.

7.2.5 Customizing Graphs

This section describes how to change the appearance of a graph. To set the properties for a graph, go to **View** and select **Graph Properties, or** click the **Graph Properties** button in the **View** toolbar. This opens the **Graph Property** dialog box. The configurable options are categorized under three tabs: **Graph Background**, **View Options**, and **Legend Options**.

The top panel of the dialog box shows a preview of the current graph with the modified properties.

Note: Options selected in the **Graph Property** dialog box apply only to the selected graph.

7.2.5.1 Graph Background Tab

The background color can be set in this tab.

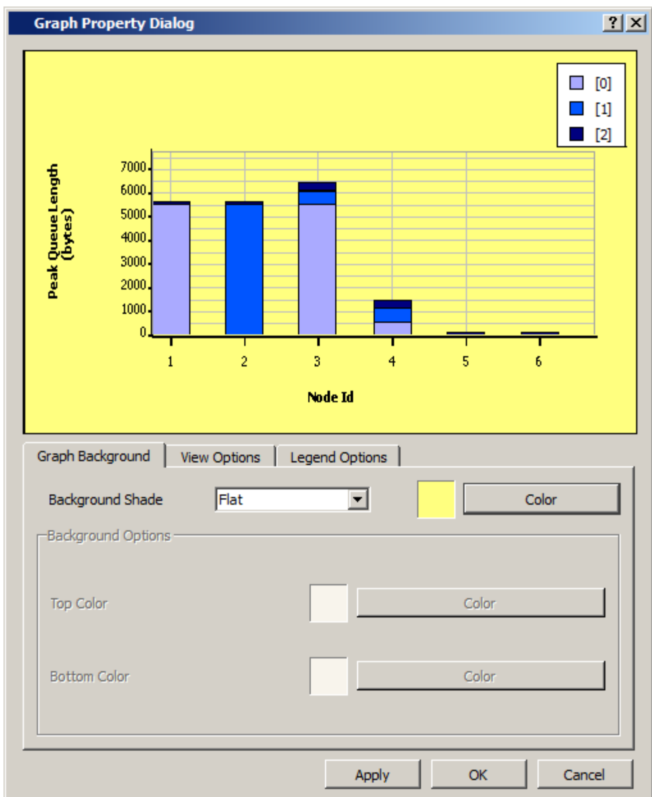


FIGURE 7-22. Graph Background Tab

- To select a flat background, set **Background Shade** to *Flat* and click the **Color** button to select a color.
- To select a gradient background, set **Background Shade** to *Linear Gradient* and select shades for **Top Color** and **Bottom Color** by clicking the two **Color** buttons. The background is displayed as a linear interpolation of the two colors.

7.2.5.2 View Options Tab

Graph options can be set in this tab.

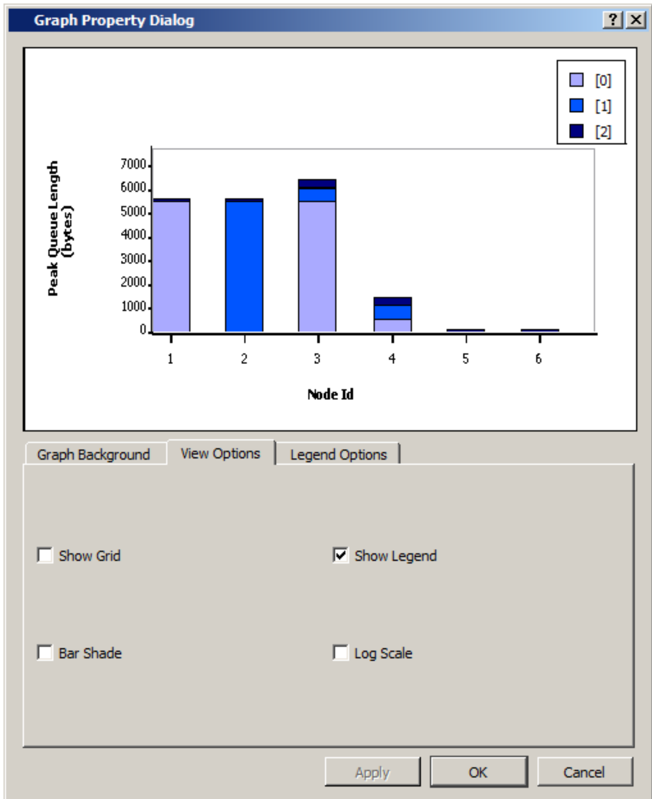


FIGURE 7-23. View Options Tab

- Check **Show Grid** to display a grid on the graph.
- Check **Show Legend** to display a legend.
- Check **Bar Shade** to use a gradient shading for bars in a graph.
- Check **Log Scale** to use a log scale for metric values.

7.2.5.3 Legend Options Tab

The color for the legend background and the colors used to plot different metrics (for example, metrics for different instances) can be selected in this tab.

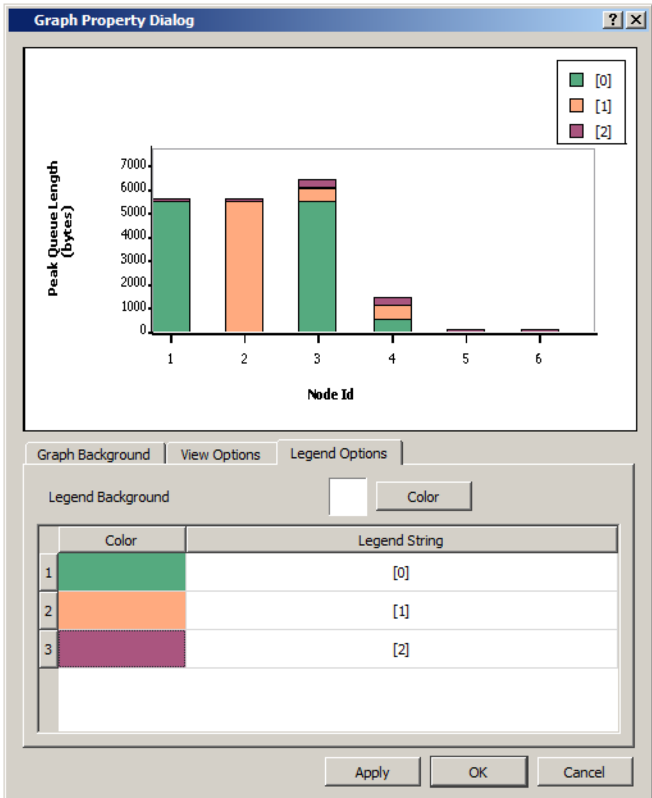


FIGURE 7-24. Legend Options Tab

- To change the legend background color, click the **Color** button to select a color.
- To change the color used to plot a metric, click on the appropriate **Color** column and select a color.

8 EXata Packet Tracer

Packet Tracer is a module of the EXata GUI that provides a visual representation allowing you to analyze information contained in trace files. Trace files are text files in standard XML format with a ".trace" extension, that contain information about the movement of packets up and down the protocol stack.

Packet Tracer presents the trace data in a structured visual format that can easily be interpreted. It provides search, sort, and filter functionality for meaningful and efficient results, and provides the following:

- Support for viewing trace data of packets through multiple protocols across layers and nodes in a tabular format.
- Support for viewing the protocols and their attributes in a hierarchical tree format.
- The ability to search the attributes of the protocols (for example, Source Port, TTL, Fragment Offset, etc.) for matches to various conditions.
- Ability to filter the entire data of a trace file based on various parameters (for example, Tracing Protocol, Originating Node, etc.). The filter should be a queue-based filter with progressive filtering, so that the output of a level serves as the input to the next level.
- Support for column sorting, string searching, and moving to a particular record of the tabular trace data.

Note: Trace file data is collected only when packet tracing is enabled, as described in Section 4.2.10.

Packet Tracer features are described in the following sections:

- Components of Packet Tracer: provides an overview of the components of Packet Tracer.
- Using Packet Tracer: describes how to analyze trace files using Packet Tracer.

8.1 Components of Packet Tracer

This section provides an overview of the different components of Packet Tracer.

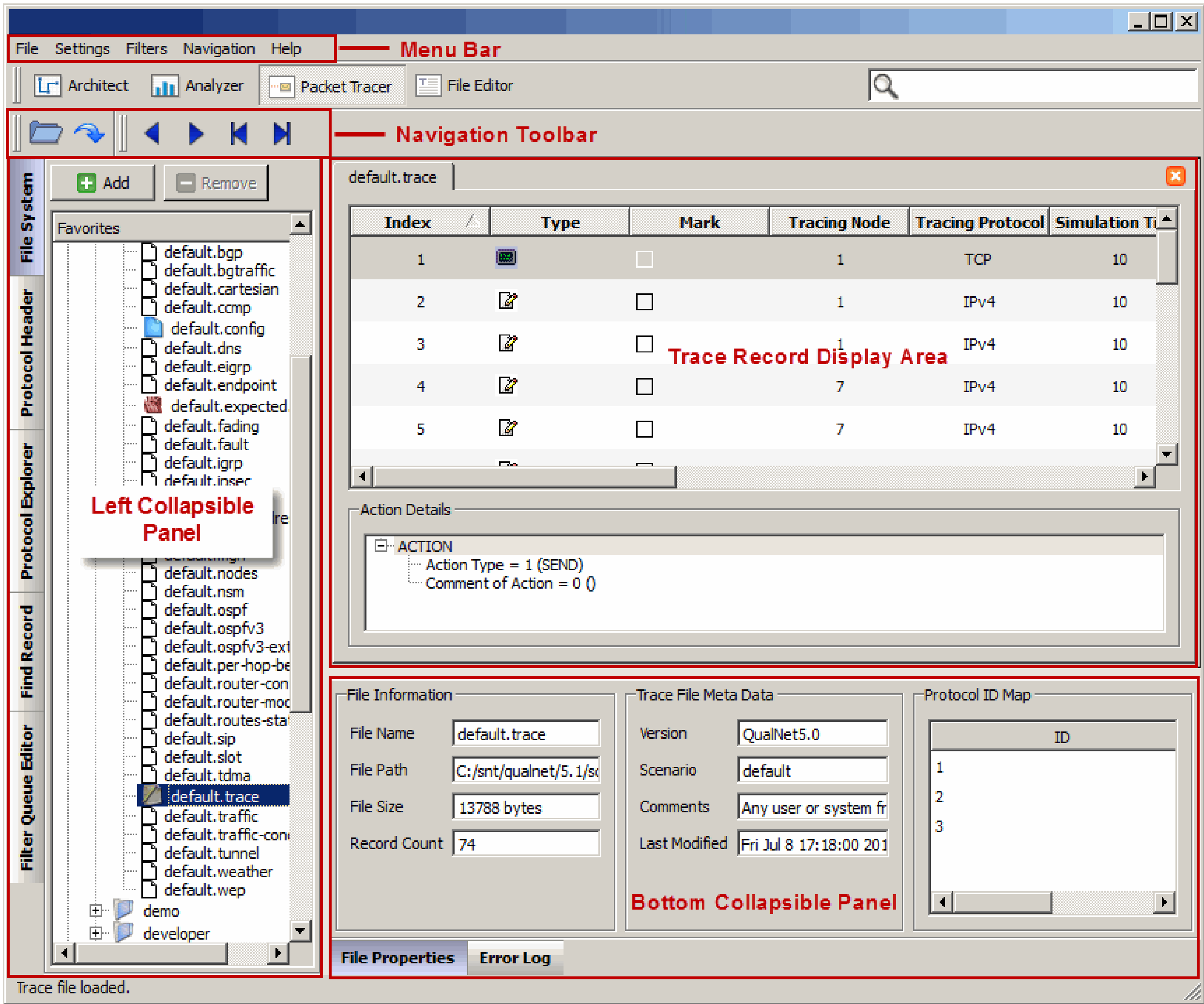


FIGURE 8-1. Packet Tracer Components

8.1.1 Menu Bar

This section explain the commands available from the **Main Menu** bar.

8.1.1.1 File Menu

The **File** menu provides the following commands for file operations.

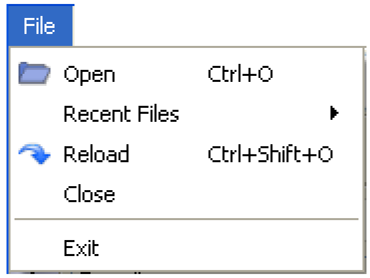


FIGURE 8-2. File Menu

Command	Description
Open	Opens a file browse dialog in the user's last opened folder. If no recent files have been opened, it opens the EXATA_HOME\scenarios\user folder.
Recent Files	Displays a list of recently opened files. Selecting a file from this list opens it in a tab on the canvas.
Reload	Closes the currently open trace file and reloads it in a new tab.
Close	Closes the currently open trace file.
Exit	Exits from EXata application. If there are any unsaved changes in any of the open scenarios, you will be prompted to save them.

8.1.1.2 Settings Menu

The **Settings** menu provides the following commands for file operations.

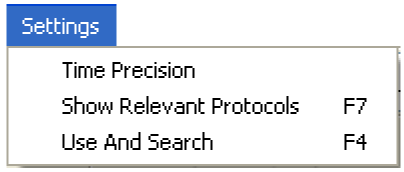


FIGURE 8-3. Settings Menu

Command	Description
Time Precision	It allows user to display the simulation time data precision up to the specified places. A pop-up window opens, when the user clicks on this menu item.
Show Relevant Protocols	It allows user to view the details of relevant protocols (the list of protocols corresponding to the traces available in .trace file) for the currently loaded trace file.
Use And Search	It will enable/disable a global flag for find record pane. If this menu item is enabled, Find Record (in left collapsible pane) will search for those records in Data Record pane that satisfy all the conditions specified by rules available in Search Queue Editor.

8.1.1.3 Filters Menu

The **Filters** menu provides the following commands for file operations.

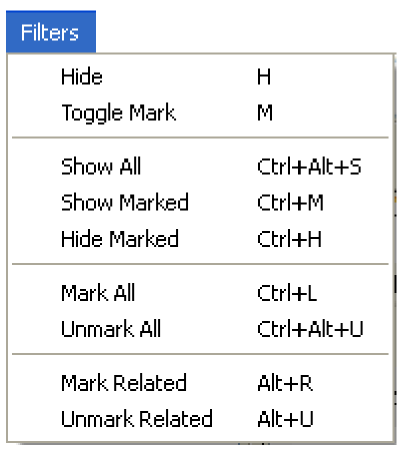


FIGURE 8-4. Filters Menu

Command	Description
Hide	Hides the currently selected data record from the Data Record pane.
Toggle Mark	Checks/unchecks the check box of currently selected record in the Data record pane.
Show All	Refreshes the table to show all the records present in the trace file.
Show Marked	Allows the user to view all the marked data records in Data Record pane and hide all the unmarked record.
Hide Marked	Hides all the marked data records in Data Record pane.
Mark All	Allows user to mark all the available data records in Data Record pane.
Unmark All	When user clicks on this menu item, it unmarks all the data records in data record pane.
Mark Related	When user clicks on this menu item, it marks all the related data records in Data Record pane with the currently marked record. Records having the same Originating Node and Message Sequence Number shall be treated as related records. If no record is marked, it displays a message “Mark at least One Record”. If there is no related record it shows a warning message “No Record Found”.
Unmark Related	Allows user to unmark all the related data records in Data Record pane with the one selected.

8.1.1.4 Navigation Menu

The **Navigation** menu provides the following commands for file operations.

FIGURE 8-5. Navigation Menu

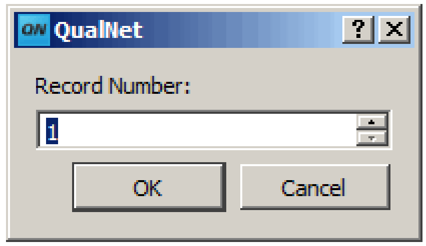


FIGURE 8-6. GoTo Dialog Box

Command	Description
GoTo	Jumps to a particular record number in the records table. A dialog box as shown in Figure 8-6 above, will accept the record number to jump to.
First Record	Selects the first record in the data record pane.
Last Record	Selects the last record in the data record pane.
Previous Record	Selects the previous record from the currently selected data record in data record pane.
Next Record	Selects the next record from the currently selected data record in the data record pane.
Page Up	Display previous page (one screen back) of the data records.
Page Down	Display next page (one screen forward), of the data records.
Previous Marked	Select the previous marked data record to the currently selected record. This option will work only when all the data records are shown.
Next Marked	Navigates to the next marked record. This option will work only when all the data records are shown.
First Marked	Navigates to the first marked record. This option will work only when all the data records are shown.
Last Marked	Navigates to the last marked record in the data record pane. This option will work only when all the data records are shown.
Previous Related	Navigates to the previous related record in Data Record pane.
Next Related	Selects to the next related data record to the currently selected data record in Data Record pane.
First Related	Selects the first related data record.
Last Related	Selects the last related data record.

8.1.1.5 Help Menu

This is the same as the **Help** menu described in the components of Architect. See Section 3.1.1.6 for a description of the **Help** menu.

8.1.2 Toolbars

This section describes the following Packet Tracer toolbars:

- Standard Toolbar
- Navigation Toolbar

8.1.2.1 Standard Toolbar

The **Standard** toolbar is used to open and close files. The following table describes the buttons of the **Standard** toolbar.

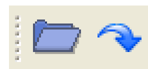
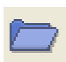


FIGURE 8-7. Standard Toolbar

Button	Command	Description
	**Open File**	Opens a file chooser in the user's last opened folder. If no recent files have been opened, it opens the EXATA_HOME/scenarios/user folder. This button performs the same function as the **File > Open** command. See Section 8.1.1.1.
	**Reload**	Closes the currently open trace file and reloads it in new tab. This button performs the same function as the **File > Reload** command. See Section 8.1.1.1.

8.1.2.2 Navigation Toolbar

The **Navigation** toolbar is used to browse through the record table. The following table describes the buttons of the **Navigation** toolbar.

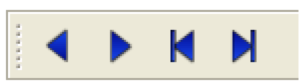


FIGURE 8-8. Navigation Toolbar

Button	Command	Description
	**Previous Record**	Selects the previous record from the currently selected data record in record table. This button performs the same function as the **Navigation > Previous Record** command. See Section 8.1.1.4.
	**Next Record**	Selects the next record from the currently selected data record in the record table. This button performs the same function as the **Navigation > Next Record** command. See Section 8.1.1.4.
	**First Record**	Selects the first record in the record table. This button performs the same function as the **Navigation > First Record** command. See Section 8.1.1.4.
	**Last Record**	Selects the last record in the record table. This button performs the same function as the **Navigation > Next Record** command. See Section 8.1.1.4.

8.1.3 Left Panels

The following panels are available to left of the data record panel:

- File System Panel
- Protocol Header Panel
- Protocol Explorer Panel
- Find Record Panel
- Filter Queue Editor Panel

Note: These five panels occupy the same space and at most one of them can be opened at any time.

8.1.3.1 File System Panel

This is same as **File System** panel of Architect. See Section 3.1.3.1 for a description of the **File System** panel.

8.1.3.2 Protocol Header Panel

The **Protocol Header** panel displays header data of a trace record as shown in Figure 8-9. Click on any row of the record table, it shows a hierarchical tree structure of protocols for that packet.

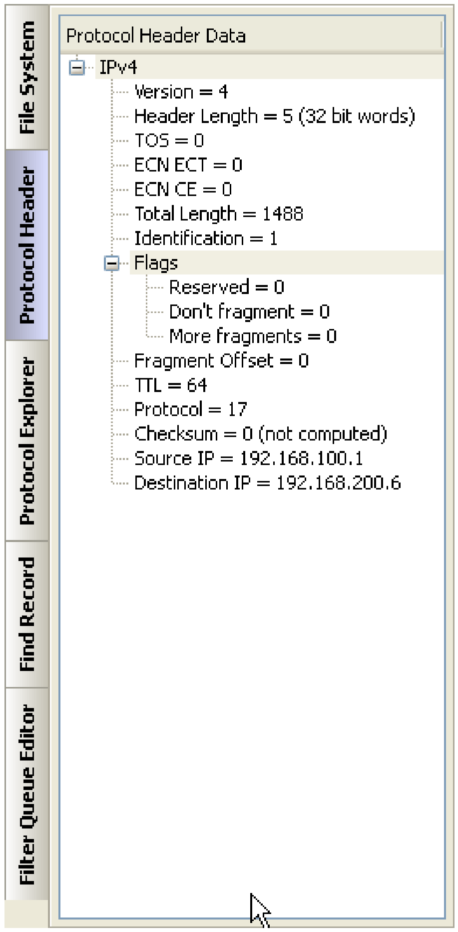


FIGURE 8-9. Protocol Header Panel

Note: This panel is blank when there is no trace file loaded and no table record is selected.

8.1.3.3 Protocol Explorer Panel

The **Protocol Explorer** displays the list of protocols supported by Packet Tracer tool (see Figure 8-10). For each protocol, it also displays attribute fields and their types.

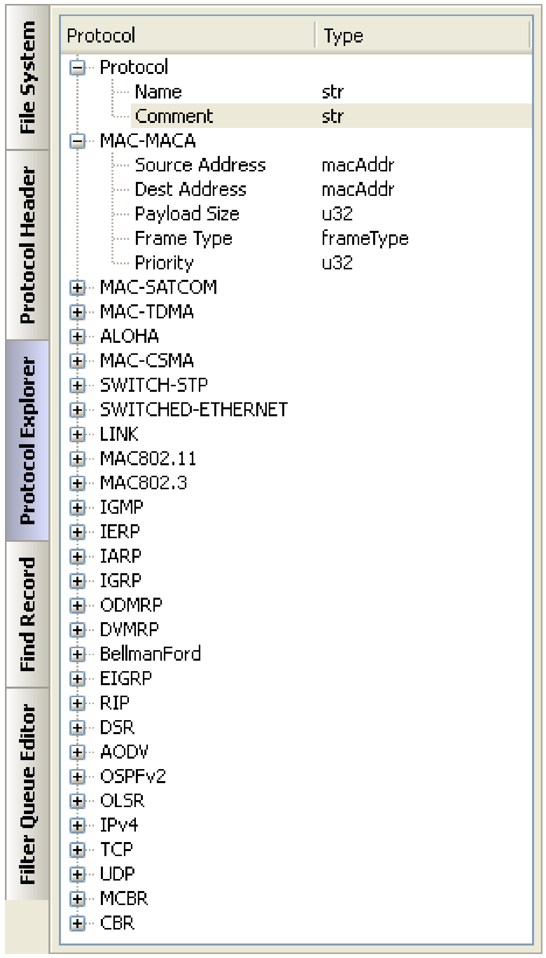


FIGURE 8-10. Protocol Explorer Panel

8.1.3.4 Find Record Panel

The **Find Record** panel is used to create and display search rules which can be applied to find records in the trace file.

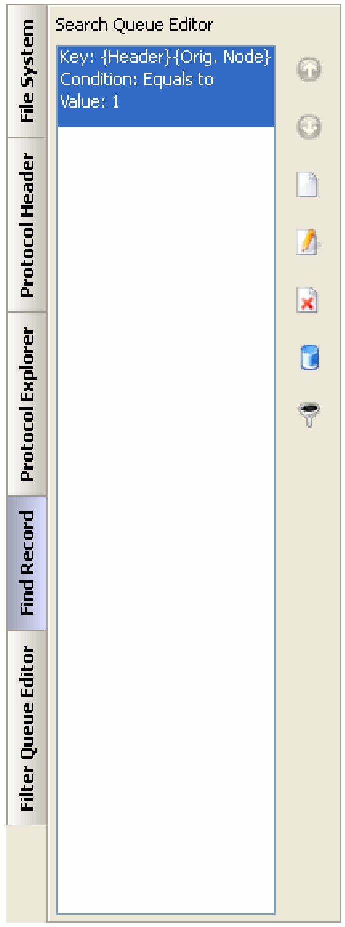


FIGURE 8-11. Find Record Panel

Note: This panel is blank if no trace file is loaded or no rule is added.

See Section 8.2.2 for details of using search rules.

8.1.3.5 Filter Queue Editor Panel

The **Filter Queue Editor** provides advanced filtering and searching functions on various parameters of the trace file data.

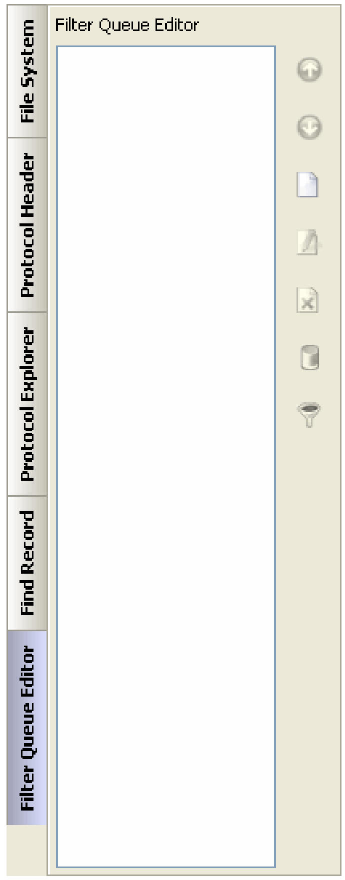


FIGURE 8-12. Filter Queue Editor Panel

See Section 8.2.2 for details of the filtering and searching functions that can be performed from the Filter Queue Editor panel.

8.1.4 Bottom Panels

The following panels are available to bottom of the data record pane:

- File Properties
- Error Log

Note: These two panels occupy the same space and at most one of them can be opened at any time.

8.1.4.1 File Properties Panel

This File Properties displays the properties of trace file, as shown in Figure 8-13. It has three groups:

- **File Information**: This group box shows the file's name, absolute path, size, and record count.
- **Trace File Meta Data**: This group shows the EXata kernel version, name of scenario, comments, and last modification date.
- **Protocol ID Map**: This group shows the list of protocols present in the loaded trace file. This list has two columns displaying Protocol ID and Protocol Name, but will remain blank when no trace file is loaded.

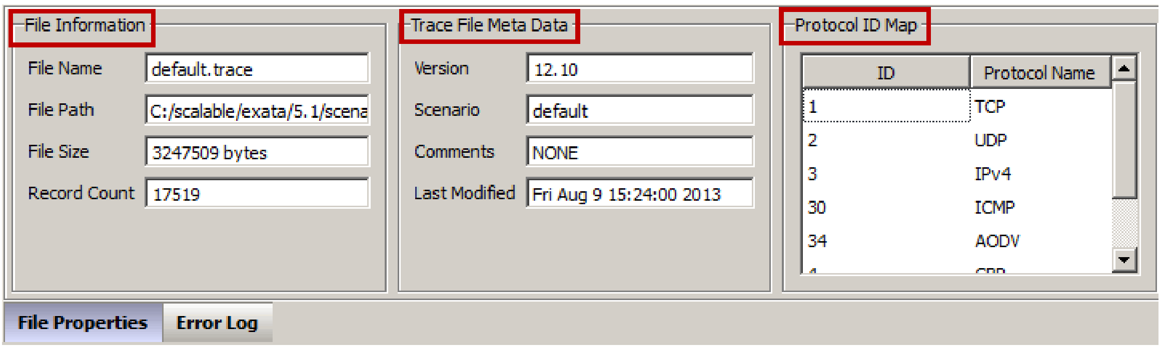


FIGURE 8-13. File Properties Panel

8.1.4.2 Error Log Panel

The **Errors Log** panel displays the parsing information corresponding to the currently selected data record from the Record Table (see Figure 8-14). It allows users to isolate errors, during the parsing of the data records.

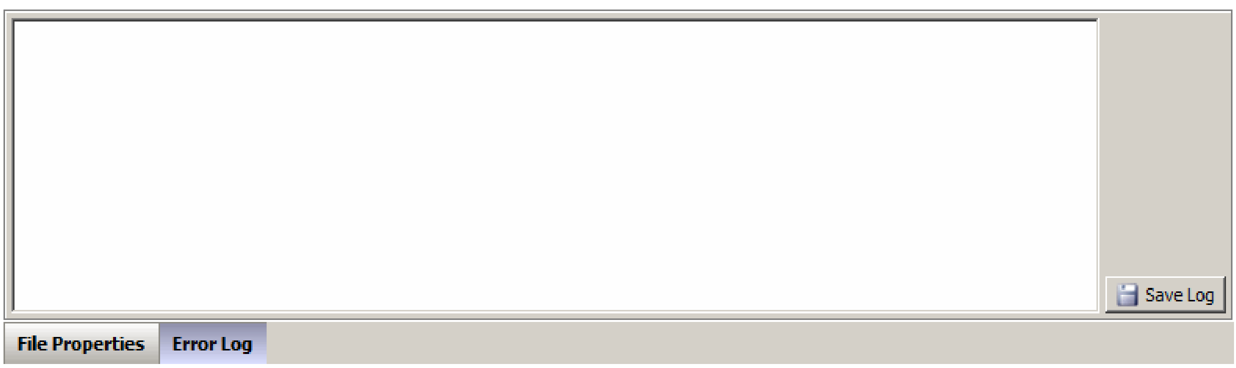


FIGURE 8-14. Error Log Panel

8.2 Using Packet Tracer

This section describes how to use the Packet Tracer features for analyzing trace files.

8.2.1 Opening Trace Files

Trace files can be opened using the following methods:

1. In the **File System** panel, either double-click on the trace file, or right-click and select **Open**.

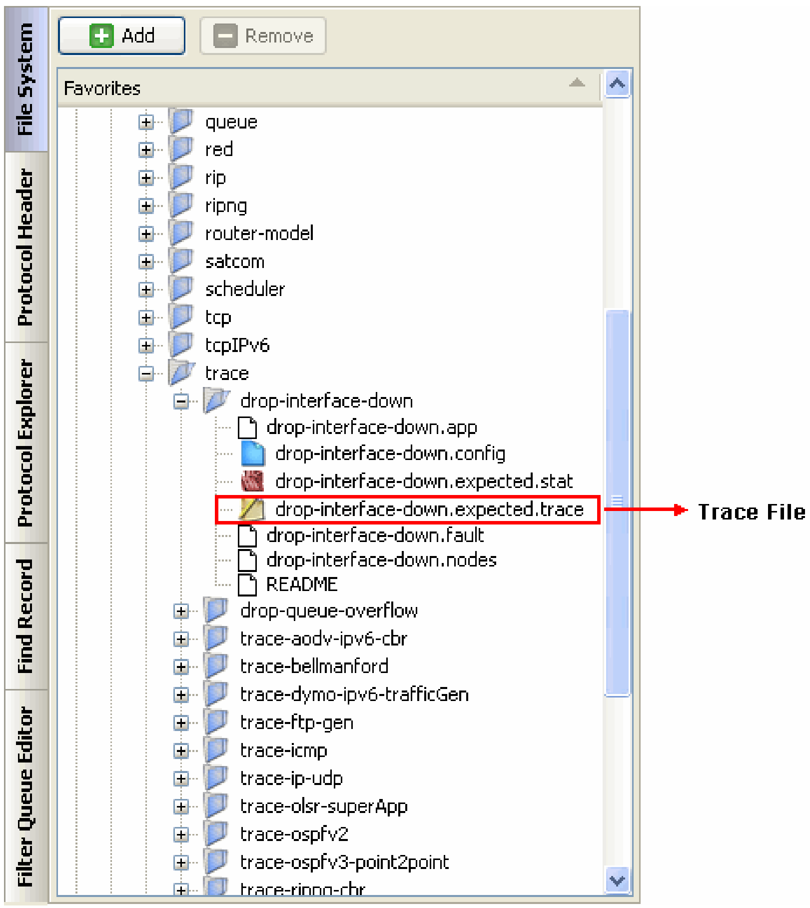


FIGURE 8-15. File System Showing a Trace File

2. Select the **File > Open...** menu option and select a trace file to open.
3. Click the **Open File** button in the **Standard** toolbar and select a trace file to open.

The records in the trace file are displayed in a tabular format, as shown in Figure 8-16.

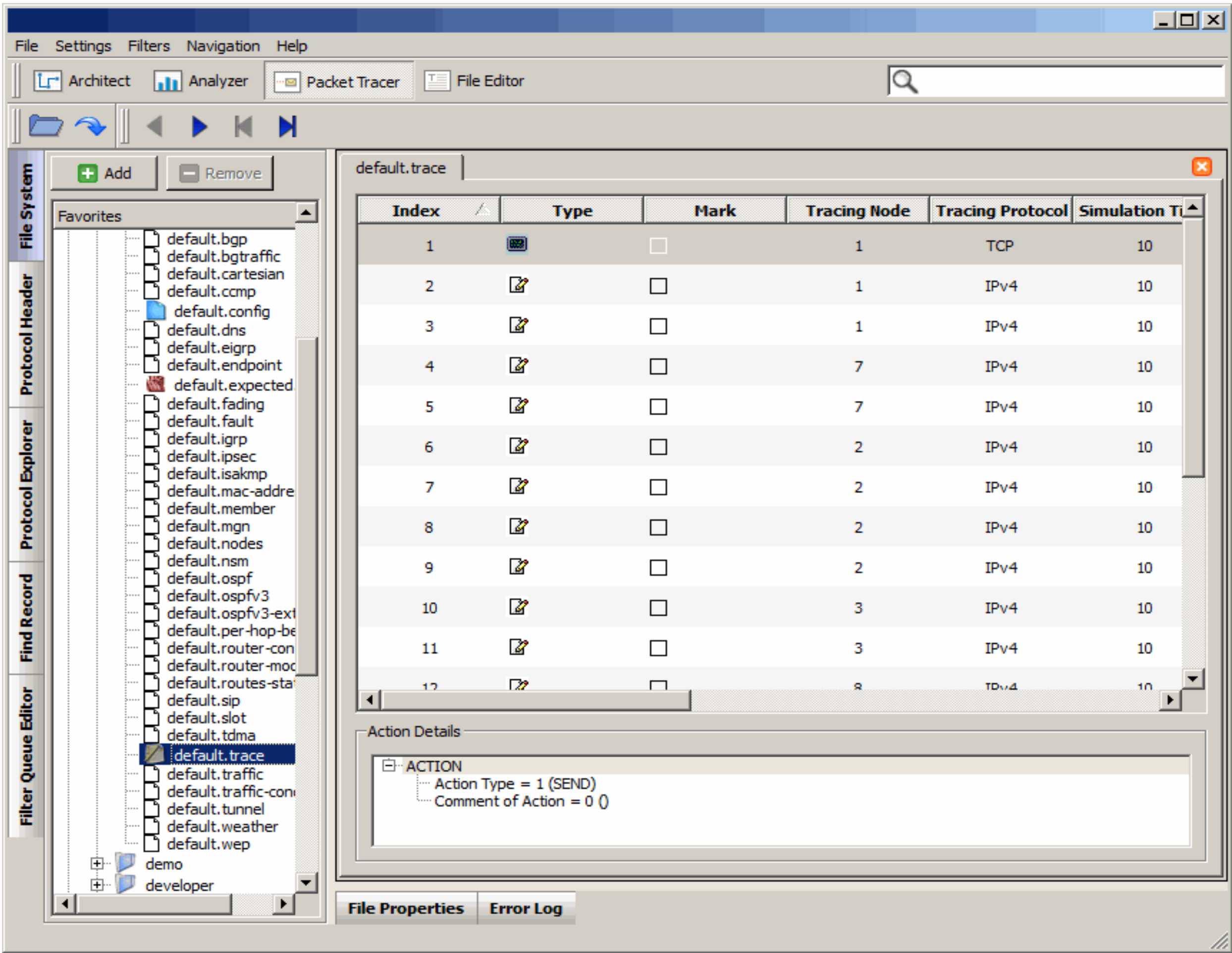


FIGURE 8-16. Trace File Displayed in Packet Tracer

The columns of the record table are described in Table 8-1.

TABLE 8-1. Tracer Table Columns

Column	Description
Index	The index number of the record.
Type	The packet type represented by the record (either Application, Control, or Fragment). You can move your mouse over an icon and its description appears over it
Mark	The selection status of a record (for example, whether you have selected a record).
Tracing Node	The node at which the packet is traced.
Tracing Protocol	The tracing protocol for the packet.
Simulation Time	The simulation time for the packet.

TABLE 8-1. Tracer Table Columns (Continued)

Column	Description
Originating Node	The node from which the packet originated.
Message Sequence Number	The message sequence number for the packet.
Originating Protocol	The originating protocol for the packet.
Action Type	The action taken for the packet (one of Send, Receive, Enqueue, Dequeue, and Drop).

Sorting the Records by Columns

You can click on any column header to sort the values in that column in ascending or descending order. The records in the table are resorted according to the sort order in the selected column. For a file with a large number of records this process may take some time.

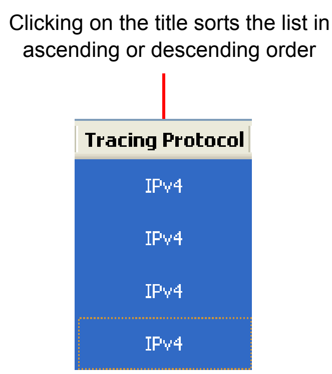


FIGURE 8-17. Sort the Trace Record

Action Details Window

A window appears at the bottom of the table, known as the Action window (see Figure 8-18). It contains information about the action taken for packets that are dropped, enqueued, or dequeued. For dropped packets, it displays the reason. For enqueued and dequeued packets, it provides Queue ID and Queue Priority.

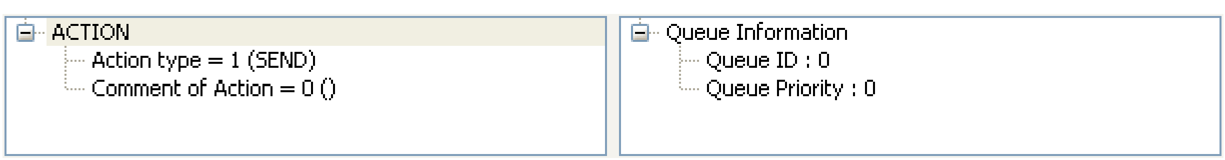


FIGURE 8-18. Action Window

8.2.2 Finding Records Using the Find Record Panel

The **Find Record** panel displays a list of added rules corresponding to the currently opened trace file. This allows user to view data records satisfying the selected search rule. This **Find Record** panel contains Move Up, Move Down, Add New Rule, Edit Rule, Delete Rule, Remove All, and Find Next Record buttons as shown in Figure 8-11.

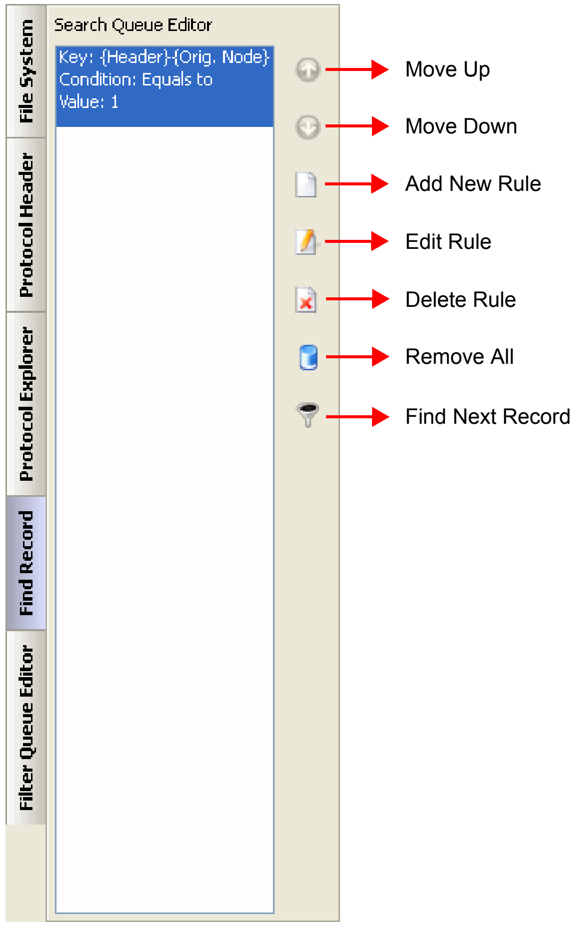


FIGURE 8-19. Find Record Panel

To search for records using search rules, do the following:

1. Click on **Add New Rule**  button. A Rule Editor dialog appears as shown in Figure 8-20.

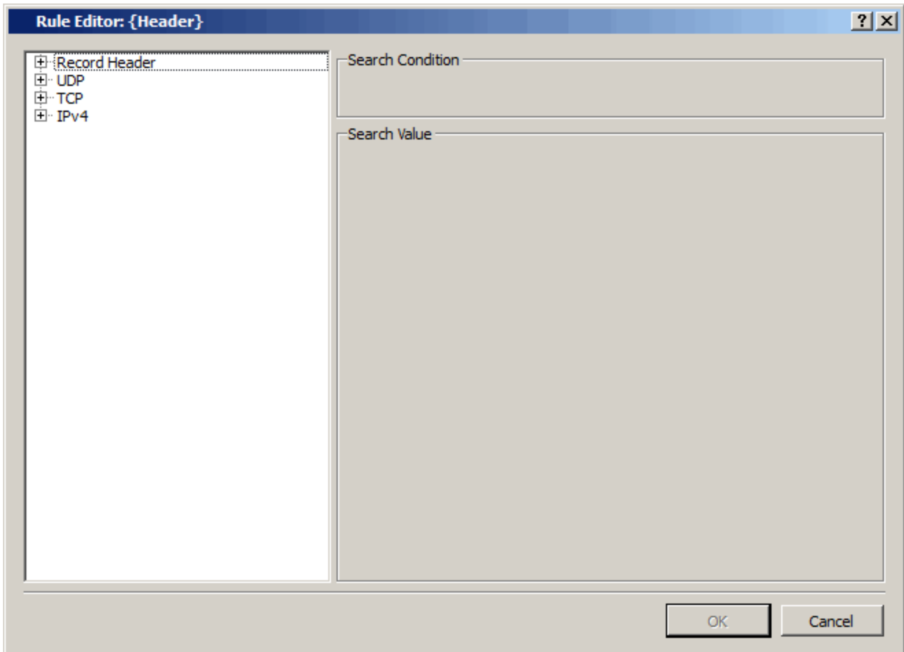


FIGURE 8-20. Rule Editor

2. Select the trace elements you want by expanding the items in the left column.

3. Choose the condition from the available search list as shown in Figure 8-21 (one of Equal to, Greater than, Less than, Greater than or equal to, less than or equal to, Not equal to, Range, and List).

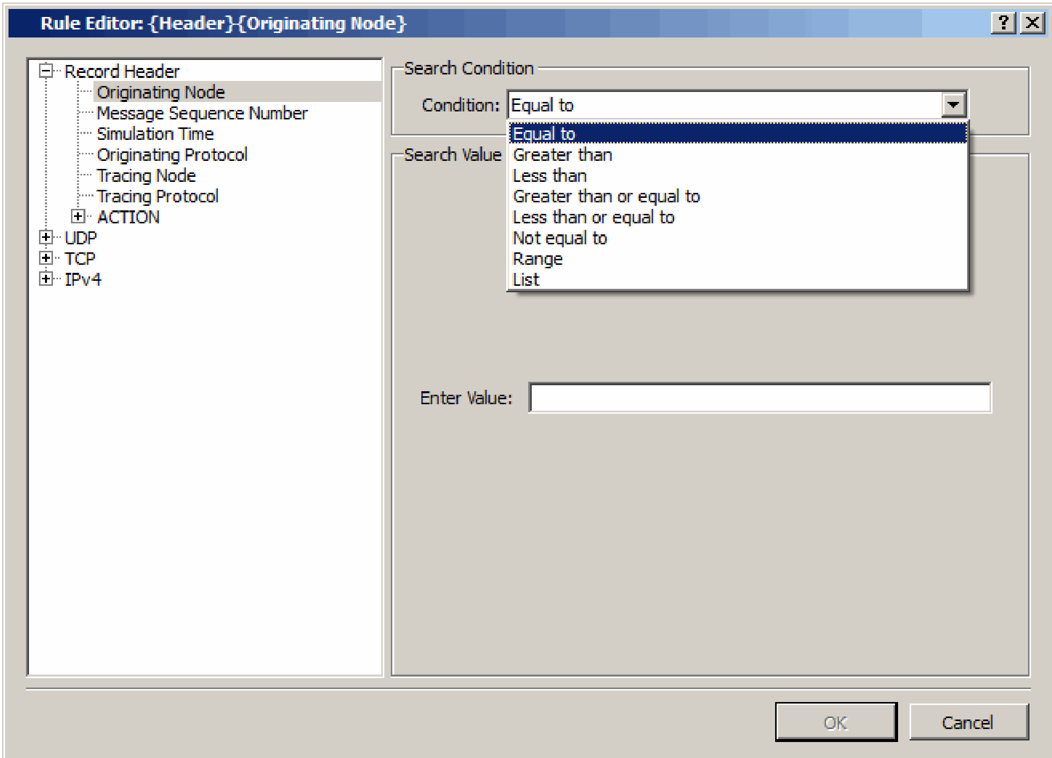


FIGURE 8-21. Setting Search Conditions

4. Specify the desired value in text box as shown in Figure 8-22 (in this case “List” was selected). Click **OK** to add the new rule in the **Find Record** panel.

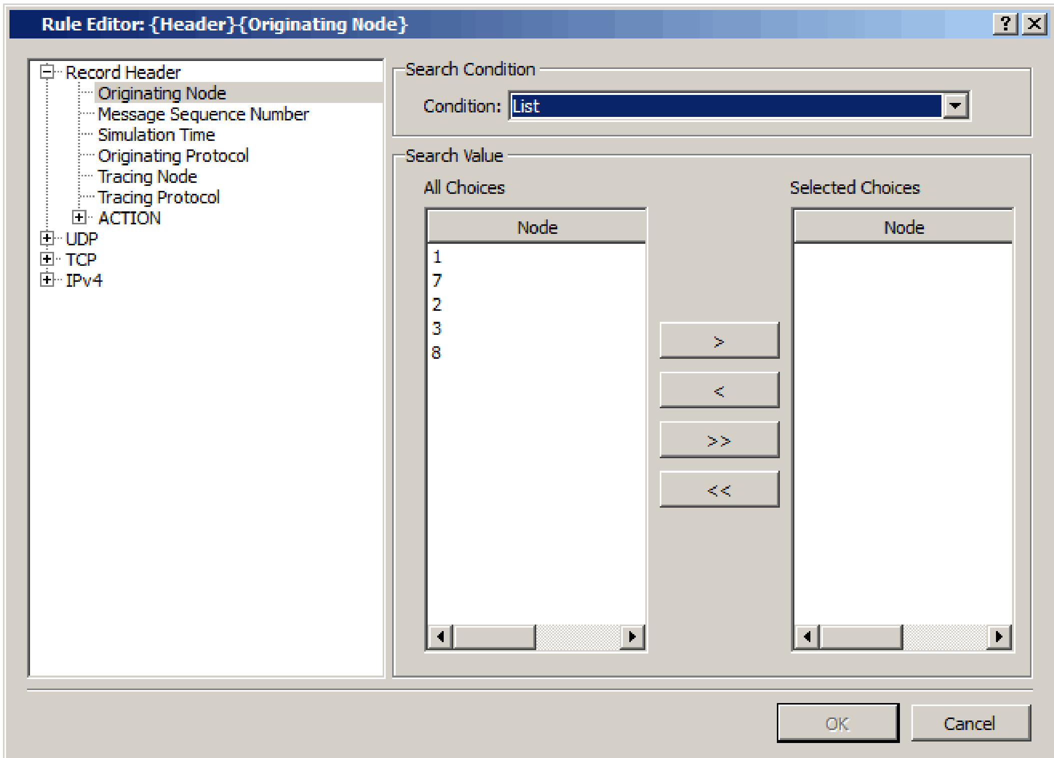


FIGURE 8-22. Setting Search Values

5. Click on **Find Next**  button and select the record satisfying the selected rule. If no data record is found, a "No Records Found" message will be displayed.

Note: You can use the **Delete Rule**  button to remove the highlighted rules from the **Find Record** panel and **Remove All**  button to remove all rules. The **Edit Rule**  button can be used to edit the highlighted/selected rules.

8.2.3 Using Filter Queue Editor

The **Filter Queue Editor** provides advanced filtering/searching functions on various parameters of the trace file data.

There are two kinds of filtering operations that you can perform using the **Filter Queue Editor**. They are:

- **Record Header Filter** - a search performed on the data visible in the Tracer Table based on various parameters explained below.
- **Record Body Filter** - a search performed on data displayed in the Protocol Header Data window.

8.2.3.1 Record Header Filter

This search can be performed on the following parameters of the trace file data:

- Tracing Node
- Tracing Protocol
- Simulation Time
- Originating Node
- Message Sequence Number
- Originating Protocol
- Action

8.2.3.1.1 Filtering on Originating Node and Tracing Node

To display only those records in the trace file that have their Originating Node as 1 or 3, perform the following:

1. Click on the **Filter Queue Editor** tab in the Left Panel. This **Tracer Table** button opens a pane called Filter Queue Editor, shown in Figure 8-23. From here, you can perform several advanced search functions, explained later in this chapter.

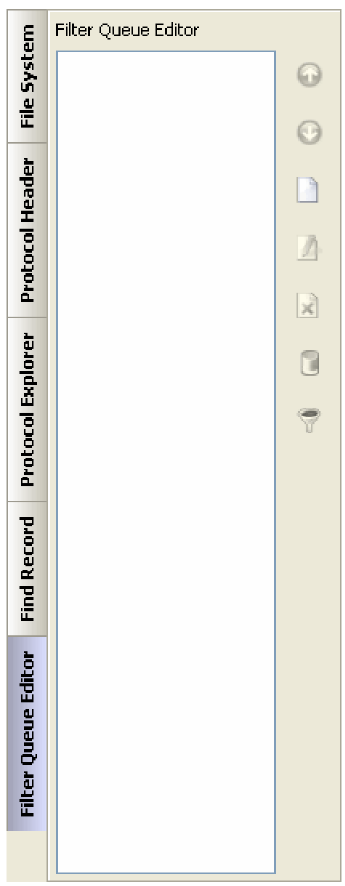


FIGURE 8-23. Filter Queue Editor Window

2. Use the **Add New Rule**  button to define a new filter rule. This launches a Rule Editor dialog as shown in Figure 7-23.
3. **In** t**he Rule Editor** dialog window, choose the **Originating Node** option in the left tree.

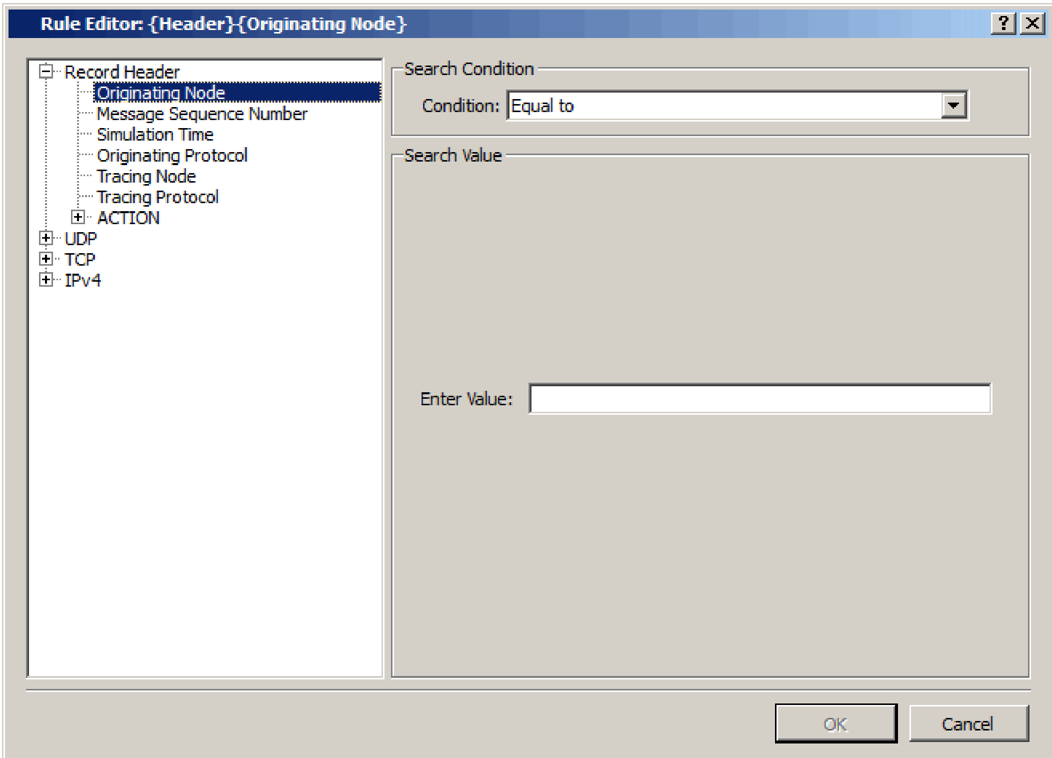


FIGURE 8-24. Rule Editor Window

4. Select **List** from the Condition drop-down list box.

   The blank area on the right of the Filter Queue Editor is now replaced by two list boxes as shown in Figure 8-25. The left list box contains all the available node IDs from the data in the trace file, whereas the right list box is empty.

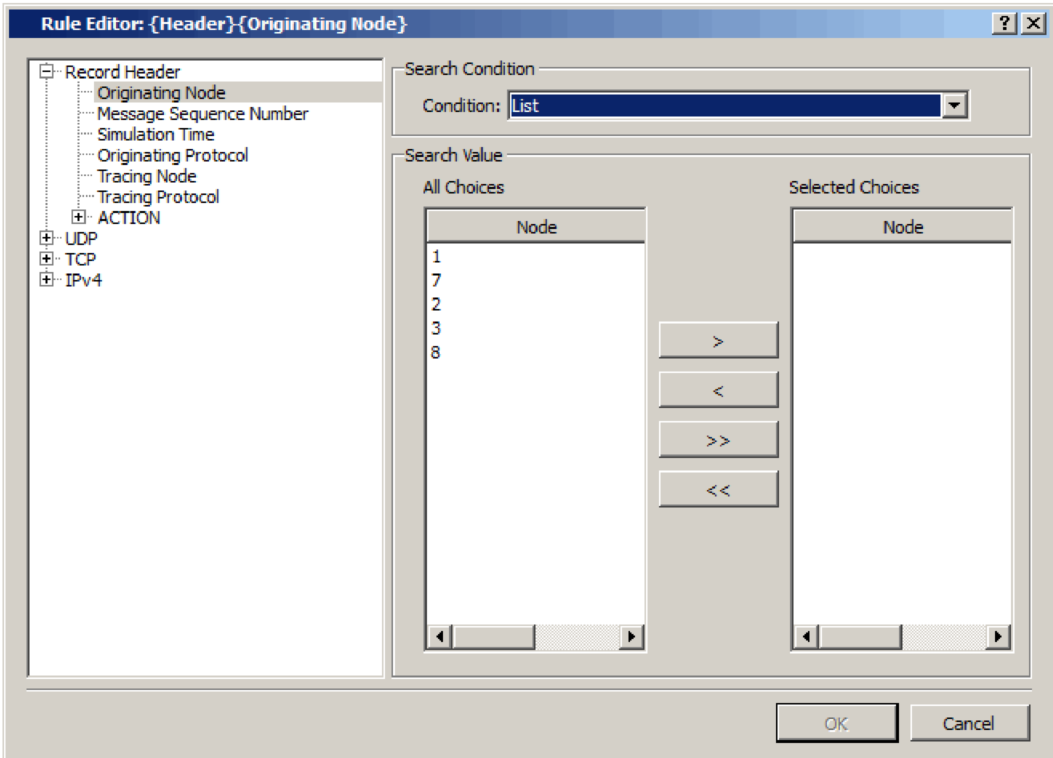


FIGURE 8-25. Filter Queue Editor Window

5. Choose the nodes that you want to filter data on as follows:

a. Select them in the Available Node IDs (**All Choices**) list and click the **>** button each time to add them to the **Selected Choices** list on the right.

b. Use the **<** button to move a selection back to the left box, or deselect it. Use the **>>** and **<<** buttons to move all the list items to either the left or right box.

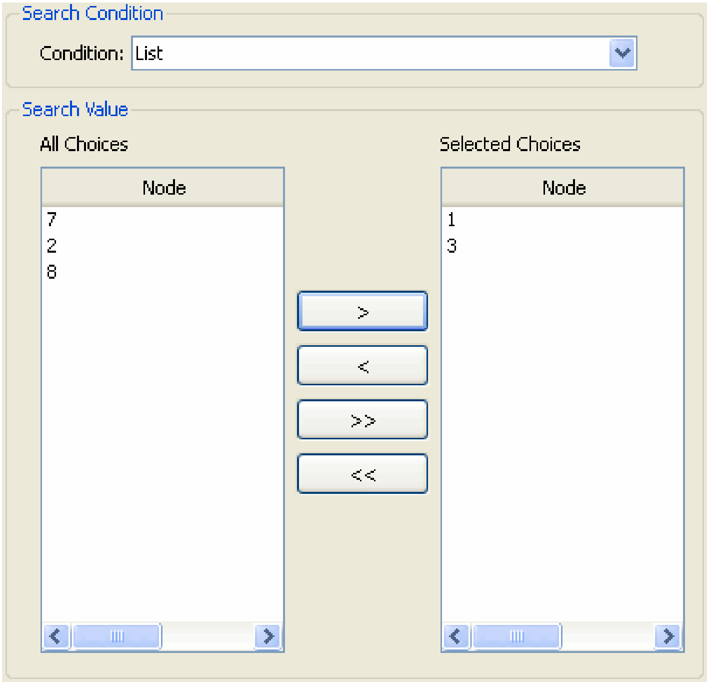


FIGURE 8-26. Making a Selection from Available Options

6. Click **OK** to accept the rule and add it to the filter queue list box.

7. A check box allows you to set the strategy of this rule to show/hide. Uncheck the **Filter Strategy** check box to hide the records with the Originating Node ID in their headers. Check the check box to only show these records.

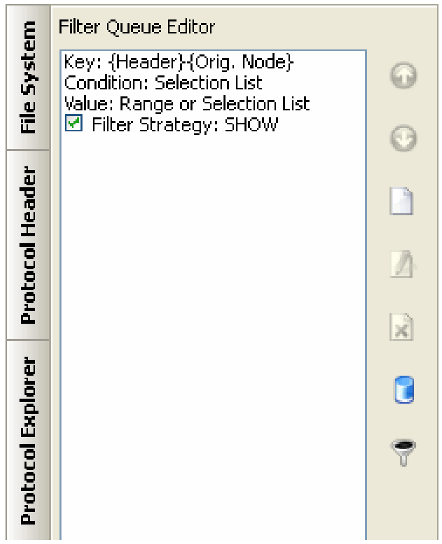


FIGURE 8-27. Rules in the Filter Queue Editor

To apply the rule to the records table, press the **Filter Records Now**  button, which gets activated when the filter queue list is non empty. This will start applying the filter rule to the visible records in the records table. The filtered data containing only the records (i.e. packets) with your chosen originating nodes appears in the records table.

Note: When multiple rules are in the filter queue, the resultant rule of all these rules will be applied to the record in the table when **Filter Records Now** button is pressed.

8.2.3.1.1.1 Operations on Rules in Filter Queue

The following operations are available: Move up, Move down, Add Rule, Edit Rule, Delete Rule, Remove All Rules, and Filter Records Now as shown in Figure 8-28.

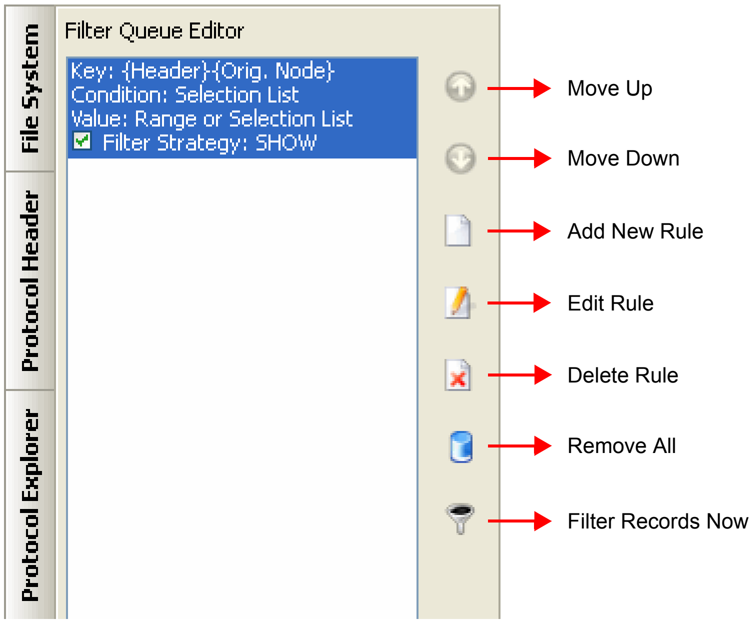


FIGURE 8-28. Operations on Rules in the Filter Queue

8.2.3.1.2 Filtering on Message Sequence Number and Simulation Time

Filtering on **Message Sequence Number** and **Simulation Time** involves specifying a range for the respective parameter that is used for the search and filtering. The following steps describe the filtering configuration for Simulation Time:

1. In the Rule Editor dialog, select the **Add New Rule**  button, and then expand the Record Header and select **Simulation Time**.
2. From the Search Condition pull-down menu, select **Range**. A Minimum and Maximum search value window appears (see Figure 8-29).
3. Enter the range for which you want to filter the trace file data in these and click on **OK**.

   The Tracer table now displays only those records (packets) that have their Simulation Time within the specified range.

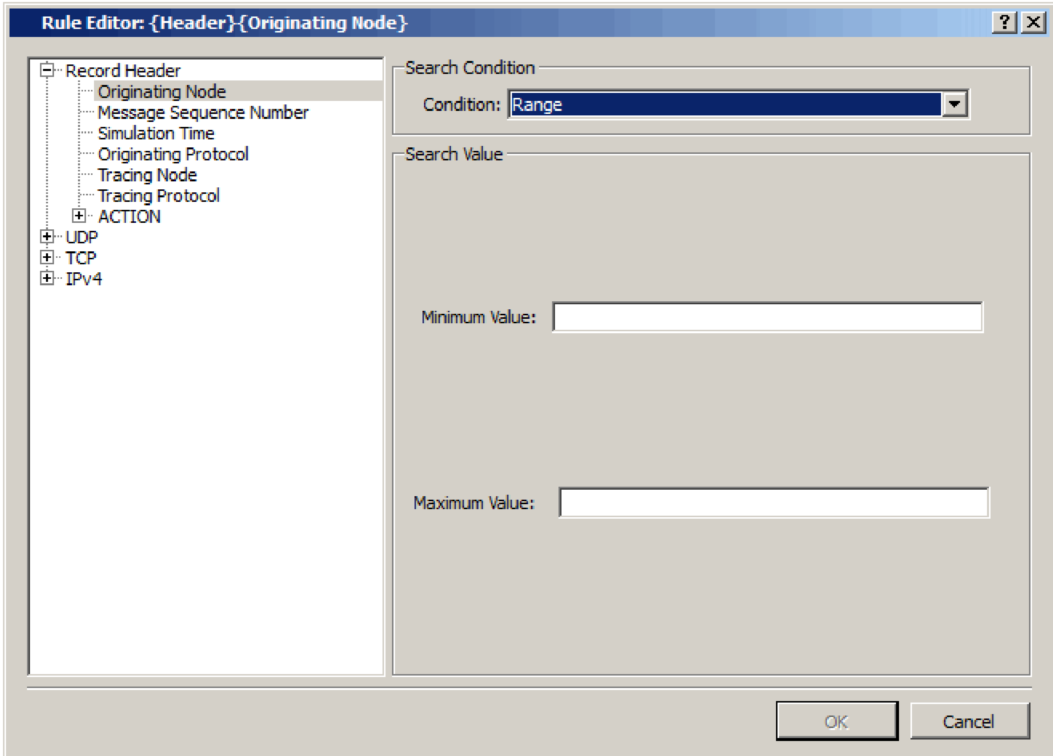


FIGURE 8-29. New Panel Containing Range for Filtering Trace Data

4. Apply the rule by clicking on the **Apply Filter Now**  button.

Note: The same procedure can be used for filtering configuration on Message Sequence Number, as well.

8.2.3.1.3 Filtering On Action

An **Action** can be any one of: Send, Receive (RECV), Enqueue, Dequeue, and Drop. The Filter Queue Editor can be configured for filtering on Action as follows:

1. In the New Filter dialog, select and expand the **Action** node under the Record Header tree as shown in Figure 8-30. Select the **Action Type** sub node. Two list boxes appear on the right-hand side of the Filter Queue Editor dialog inside Search Value area.

   The **All Choices** list containing all the Action types and the **Selected Choices** list on the right containing the ones selected by the user.

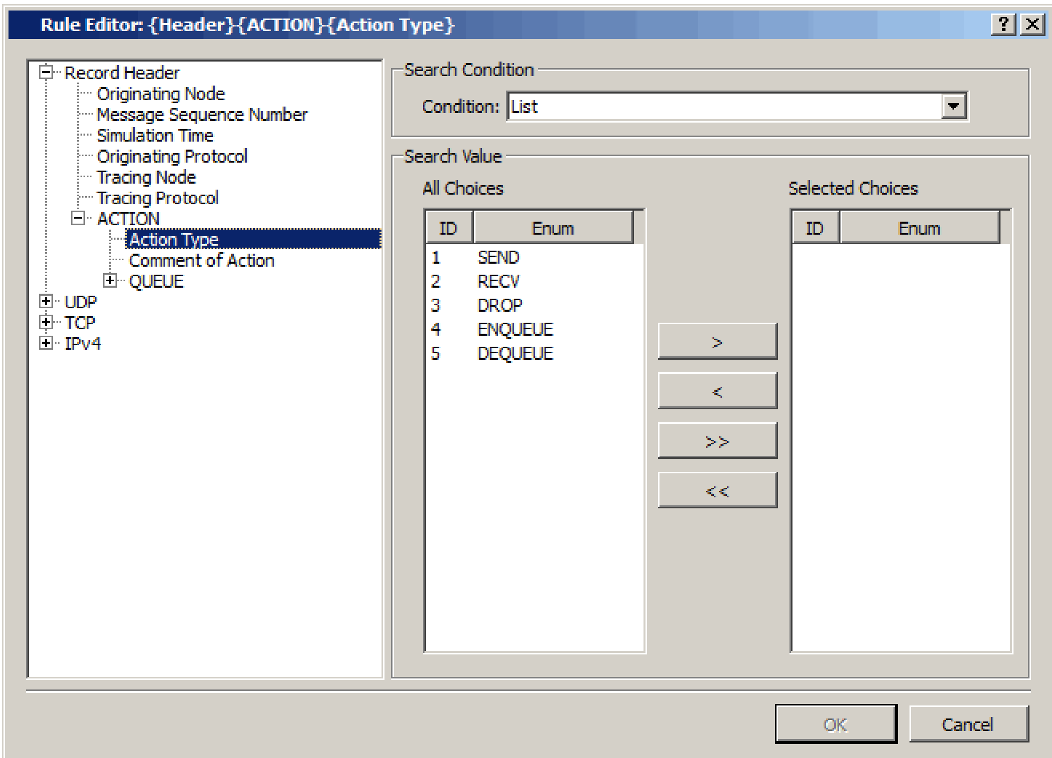


FIGURE 8-30. Selecting Action Radio Button

2. From the **All Choices** list box on the left, select the search values (one at a time) and click on the **>** button to add them to the **Selected Choices** list on the right as shown in Figure 8-31. Repeat for as many actions as you want to select. You can use the **<** button to remove an action from the selected list and the **>>** and **<<** buttons to select or deselect all items.

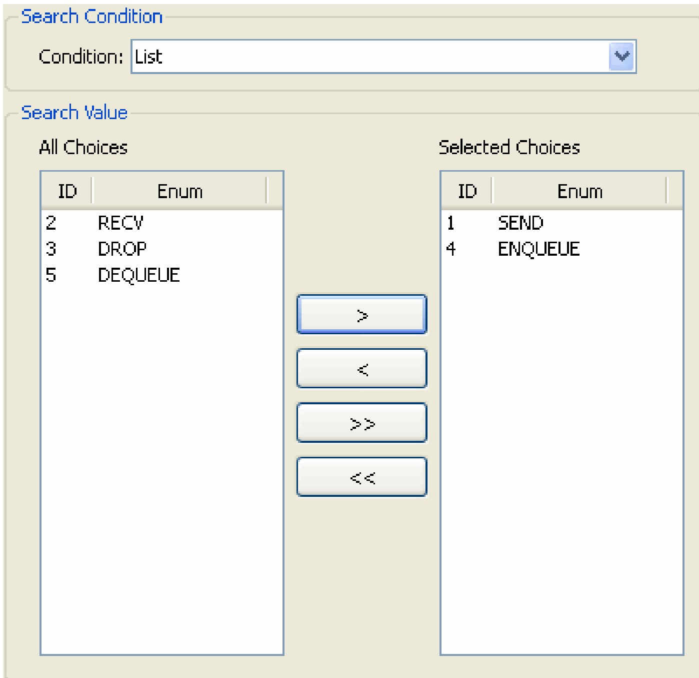


FIGURE 8-31. Selecting Actions

3. Once you have selected all search values, click the **OK** button. The Tracer table is refreshed showing only the selected records (packets) as shown in Figure 8-32. As described earlier in step 7 on page 439, you can choose to Hide, rather than Show your selection.

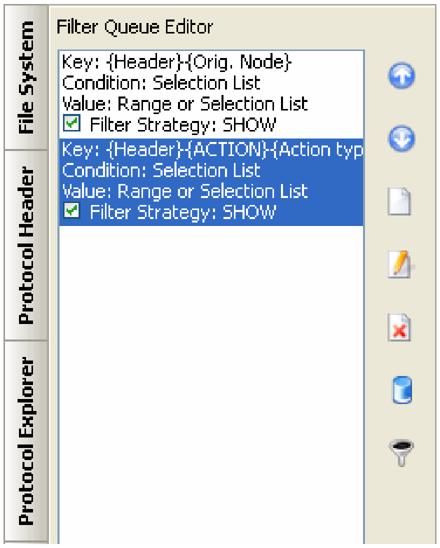


FIGURE 8-32. Shows Only Actions that can be Verified

4. Apply the rule by clicking on the **Apply Filter Now**  button.

8.2.3.1.4 Compound Progressive Filtering

The **Filter Queue Editor** performs compound progressive filtering where multiple parameters can be filtered in a queue-based manner, i.e., the output of one stage of filtering serves as the input for the next.

This progressive filtering lets you to perform complex filtering operations, for example: Show me all the records which have their Tracing Protocol as IPv4 or UDP, and their Action as either Send or Enqueue, and their Originating Node as either 1, 3, or 5, and from these records hide the ones that have their Message Sequence Number between 4 and 6. This filtering is done on a sample trace file.

How it works:

1. Open the sample trace file called default.trace. The Tracer table displays the entire trace file data as shown in Figure 8-33.

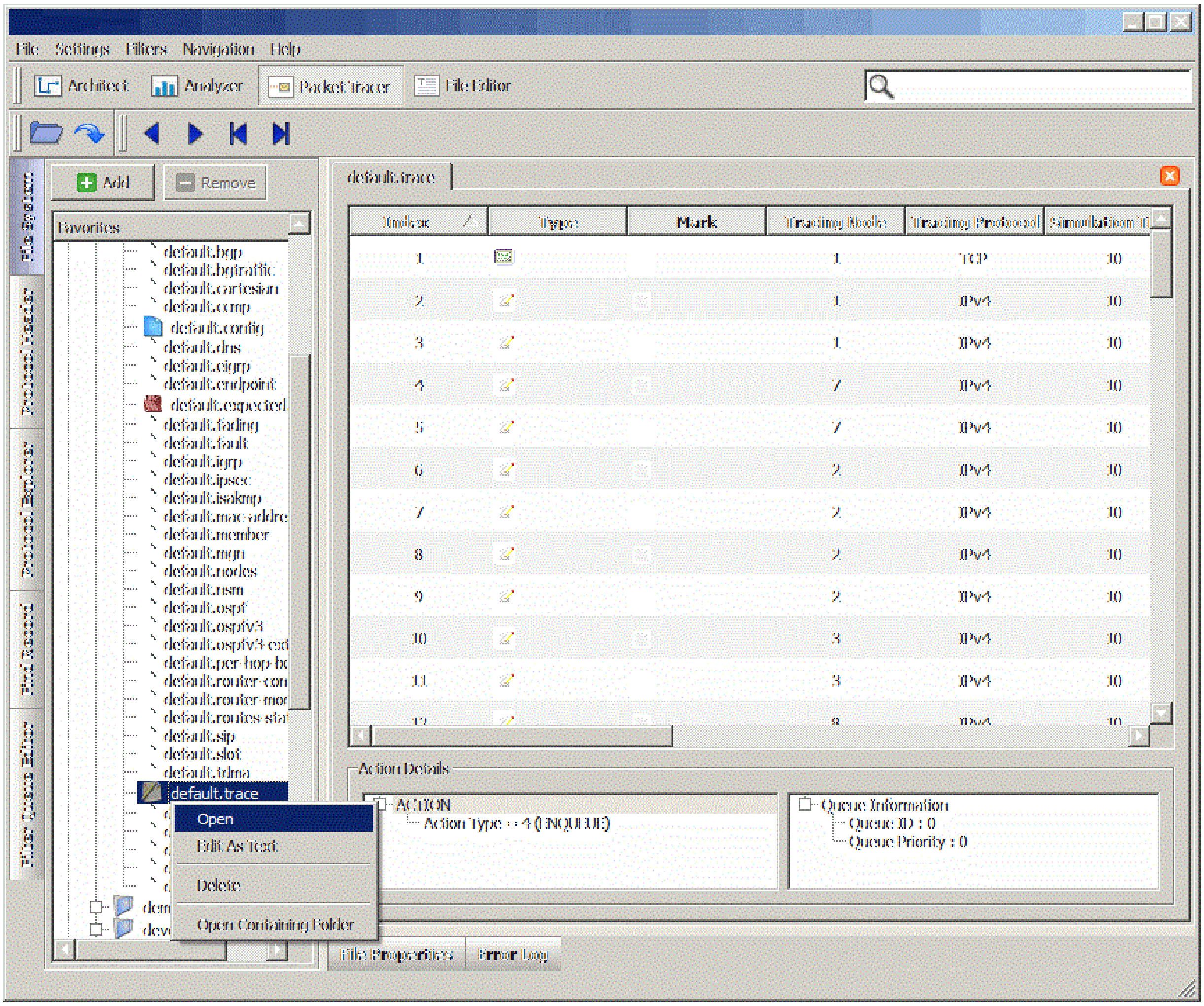


FIGURE 8-33. Sample Trace File

2. Open the **Filter Queue Editor** and in the New Filter dialog, choose **Tracing Protocol** as shown in Figure 8-34, then select **List** from the Search Condition drop-down box.

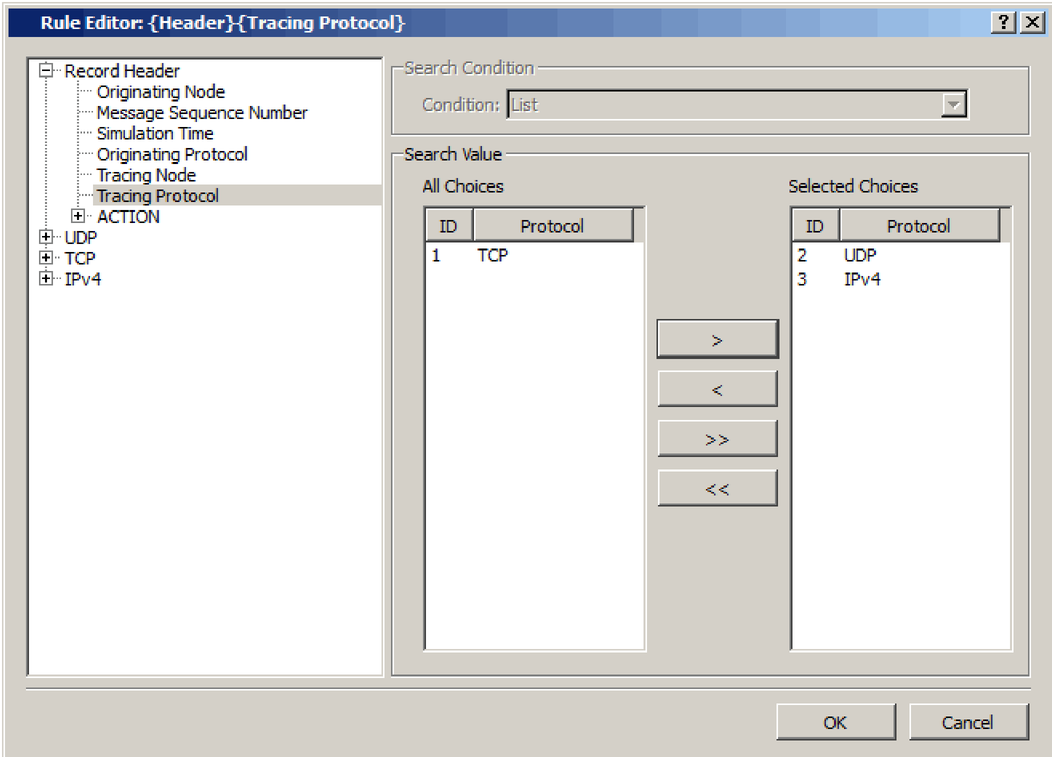


FIGURE 8-34. Selecting Tracing Protocols for Defining Rule

3. From the **All Choices** list inside Search Value box, choose the protocols that you want for Tracing Protocol (in this case UDP and IPv4) as shown in Figure 8-34 and click **OK**.

4. Click on the **Add New Rule** button in the Filter Queue Editor and from the New Rule dialog that appears choose **Action type**. Select the desired Action types (Send and Enqueue in this case) as explained earlier.

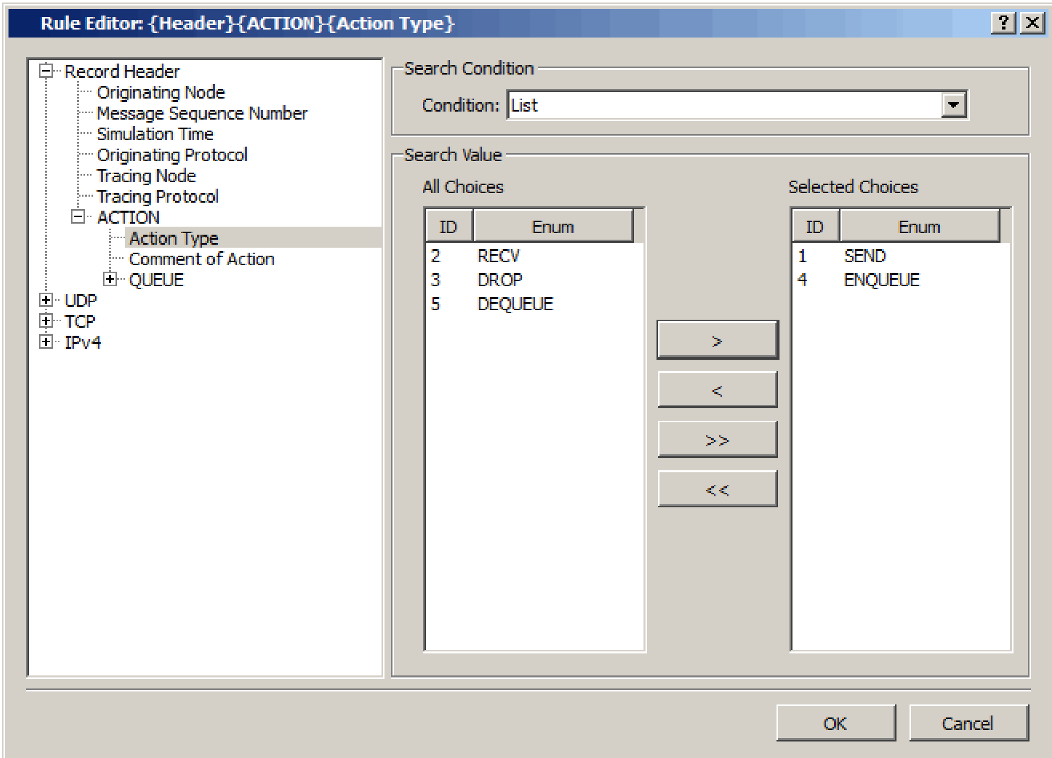


FIGURE 8-35. Choosing Your Actions

5. Define a new rule by clicking on the **Add New Rule** button, and then from the New Filter dialog that appears, choose **Originating Node** and select the **List** condition.
6. Make your selections for Originating Node (1, 3, and 8, in this case) as shown in Figure 8-36 and click **OK**.

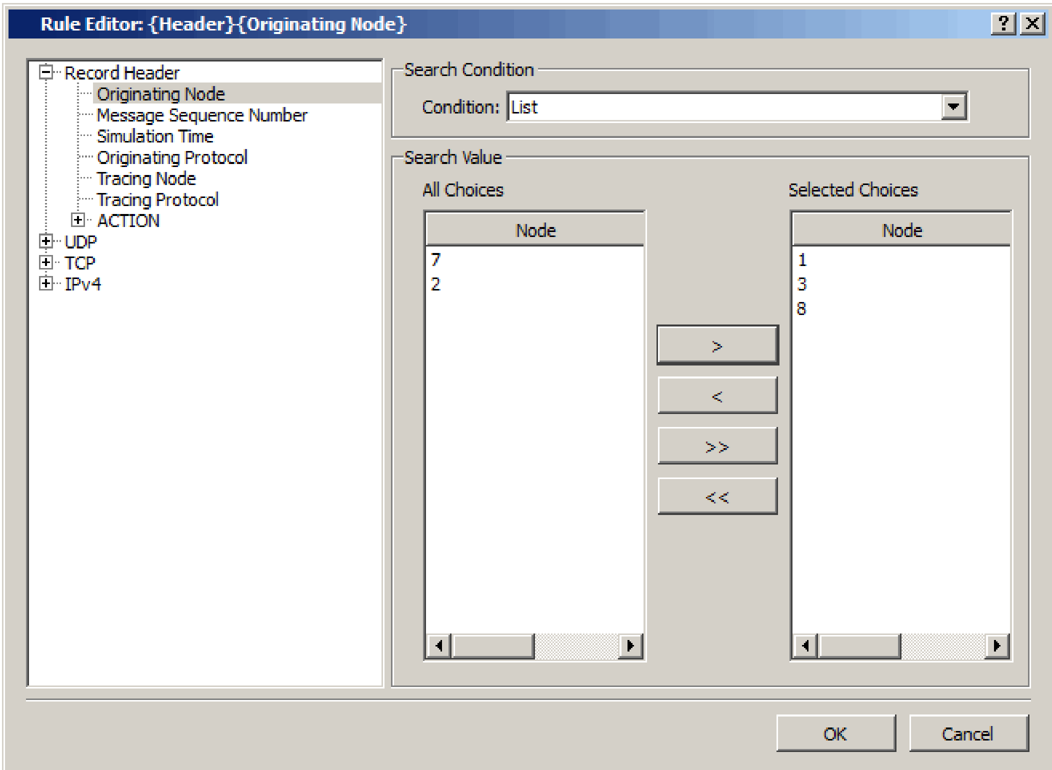


FIGURE 8-36. Choose Originating Node

7. Select **Message Sequence Number** from the Rule Editor dialog and select **Range** in the Condition drop-down list as shown in Figure 8-37.

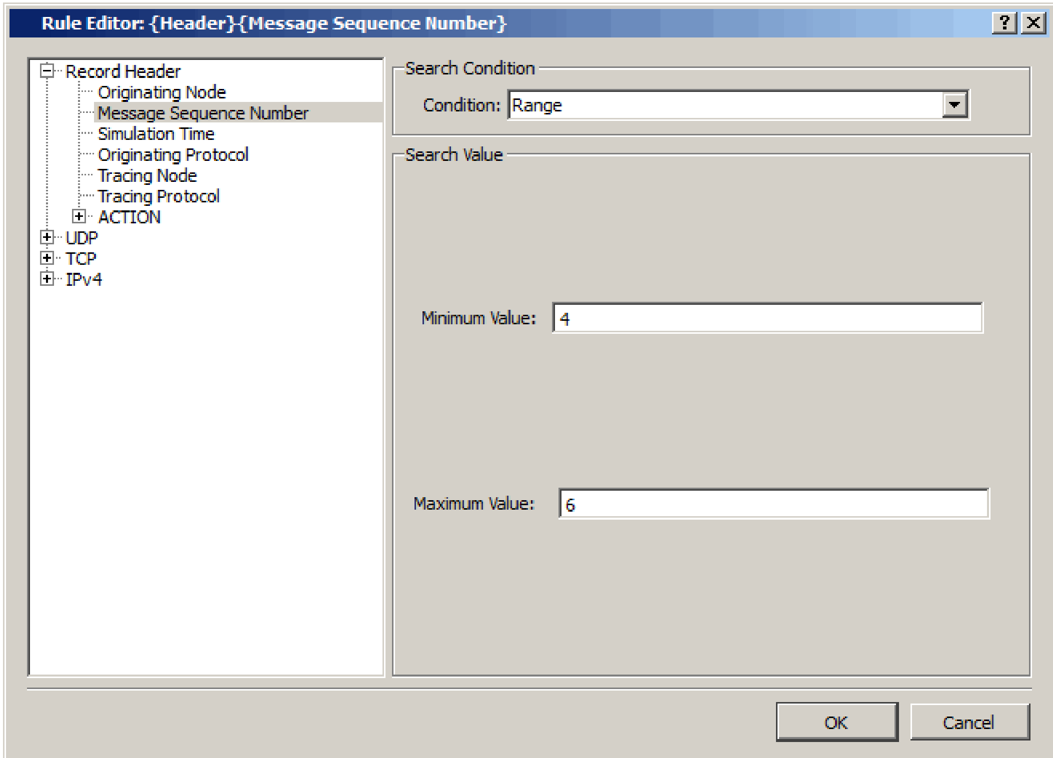


FIGURE 8-37. Updated Tracer Table

8. Enter the range for the Message Sequence Number: Minimum Value = 4 and Maximum Value = 6 (as shown in Figure 8-37) then click **OK**.

Note: After the parameters have been entered for the tracer data, the records can be filtered in the desired sequence by applying the formed Filter Queue.

9. When clicking the **Filter Records Now** button, the record table view updates to display the data from only those records (packets) that satisfy all the conditions that were entered in the Filter Queue as shown in Figure 8-38.

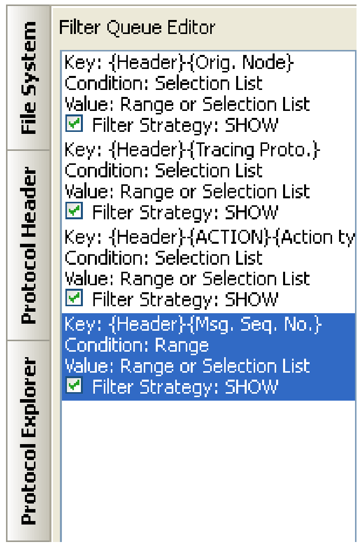


FIGURE 8-38. Filter Queue with all Rules Defined

8.2.3.2 Record Body Filter

In the previous section, we configured the Filter Queue Editor with search parameters like Originating Node, Tracing Protocol, etc. (these are displayed as columns in the Tracer table), and constitute the Record Header and so this type of filter is known as Record Header Filter.

In this section we will configure the Filter Queue Editor with Protocol Header Data (Attributes and Values) as shown in Figure 8-39, rather than the data in the Tracer table (record table).

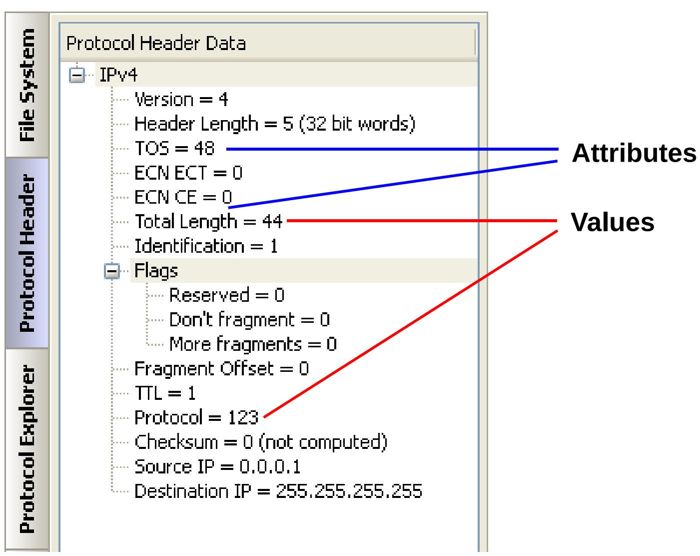


FIGURE 8-39. Protocol Header Tree

A sample of the data in the Protocol Header Data window is displayed when a row of the Tracer table is clicked as shown in Figure 8-39. This figure shows that each Protocol has Attributes (or properties) and values associated with each of these attributes. They can be identified by noting that the Attributes appear to the left of the '= ' sign and their Values to the right of it.

Using the Filter Queue Editor you can search or filter the trace file data to display only those records that contain a certain value of an Attribute by configuring it for Record Body Filtration. In the following section, we will filter records containing a specific Attribute Value from among the trace file data.

8.2.3.2.1 Protocol Header Field Search

This filtration is performed on the attribute values in the Protocol Header Data. The attribute Identification can be seen under protocol IPv4. In the figure below, this Value is 4 (but for other records it can vary). In the following scenario we will search only those records in the trace file that have the Value of the Attribute Identification under IPv4 as 4 (four).

To accomplish this search task, perform a Header Field Search as follows:

1. Open the **FilterQueueEditor**, click on the **New Rule** button and from the Rule Editor dialog that appears, choose *IPv4 tree node*.
2. Click on the **+** beside IPv4 to expand that node. The expanded IPv4 node is shown.

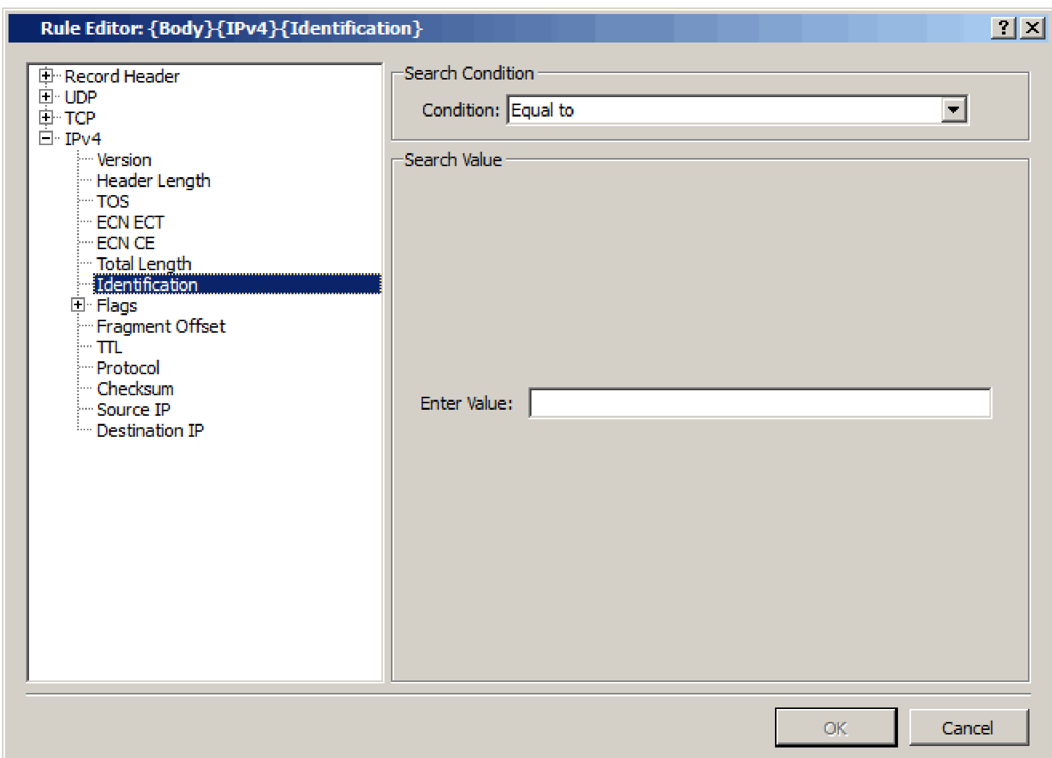


FIGURE 8-40. Expanded IPv4 Node Shown

3. From the list of nodes under IPv4, select **Identification**.
4. From the Search Condition drop-down list on the right pane, select a comparison condition (Equal to, Greater than, Less than, Greater than or equal to, less than or equal to, Not equal to, and Range).
5. Enter the value(s) for the filter configuration and click **OK** (in this case we have entered 4).

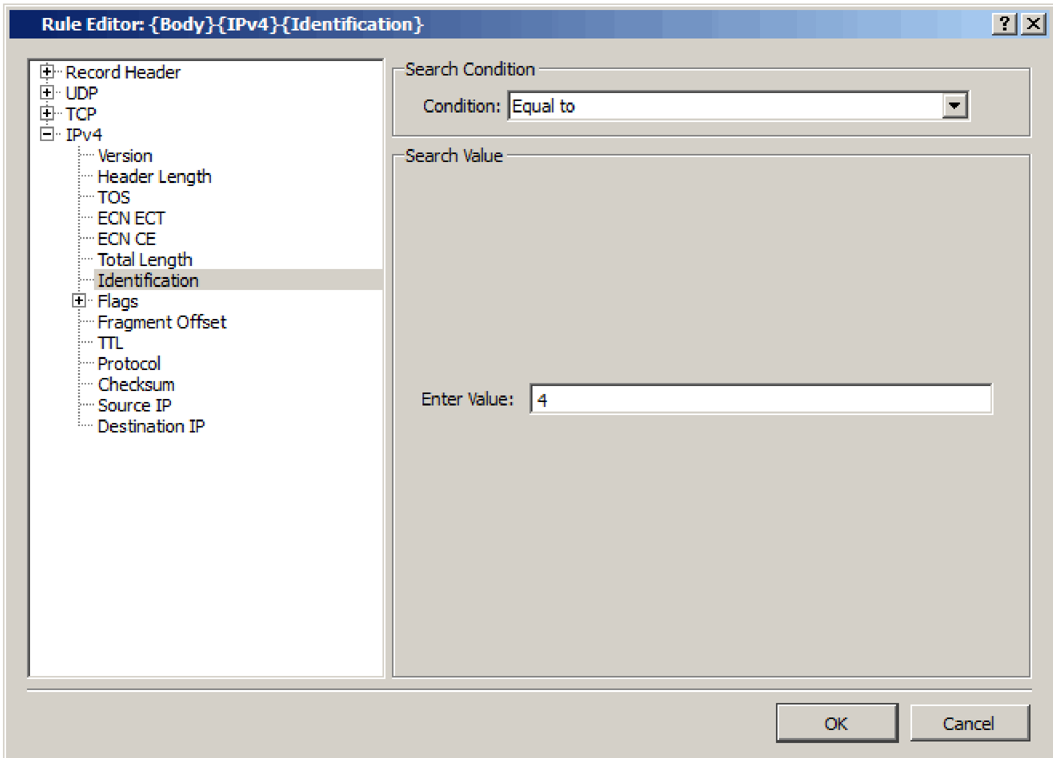


FIGURE 8-41. Providing the Attribute’s Value for the Rule

6. The new rule appears in the Filter Queue Editor. To view the results of your filtering operation on the Protocol header tree press **Filter Records Now**  button. Only those records in the trace file will be displayed in the Tracer table that have the Value of their Identification attribute of IPv4 protocol as 4.

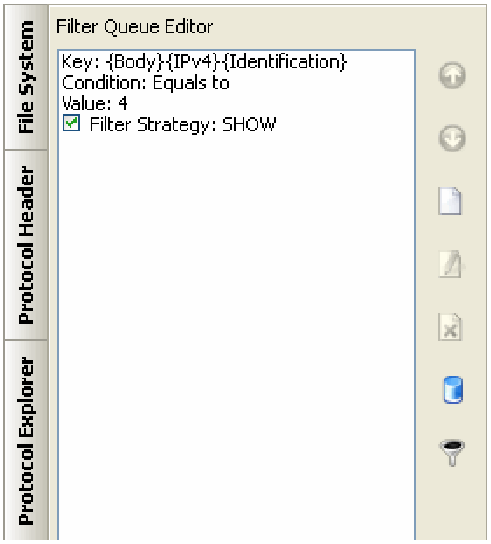


FIGURE 8-42. New Rule Added to Filter Queue

In this example of a Header Field Search, the search item was a simple number. Similarly, we may filter records on any other protocol header-field-value like Source IP and also specifying multiple rules to make any combination of values, including progressive attribute filtering on the records for these rules.

Figure 8-43 shows records filtered by an IP address and Figure 8-44 shows multiple rules for filtering header values.

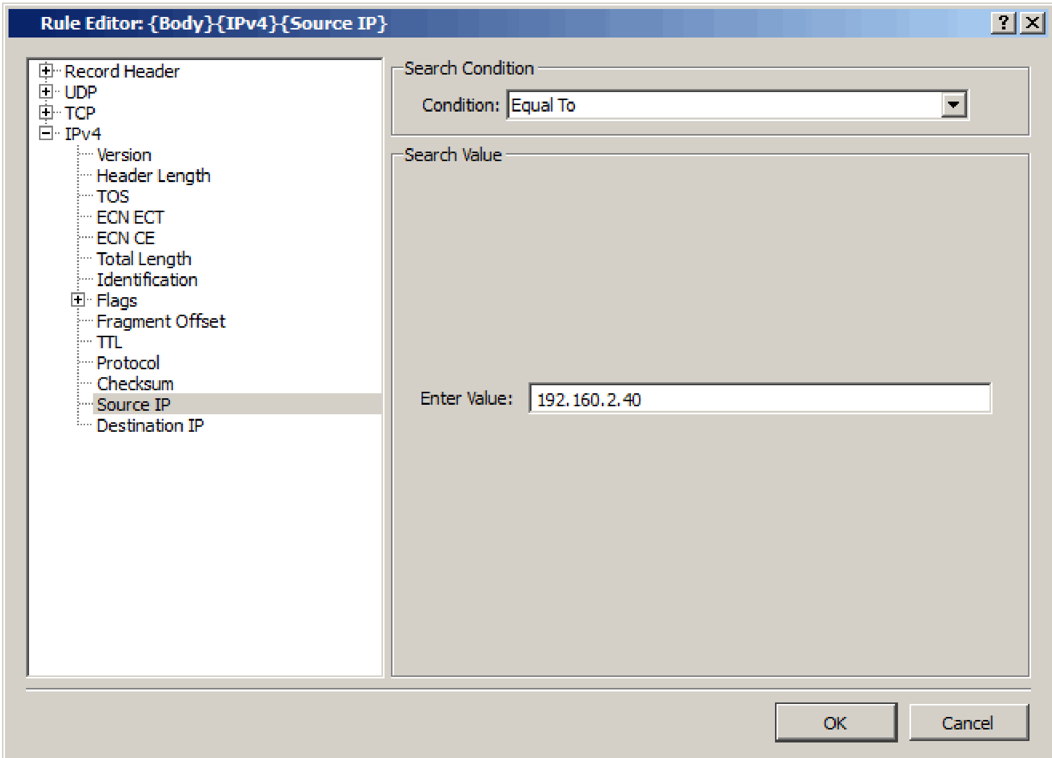


FIGURE 8-43. Selecting a Node that Represents the IP Address

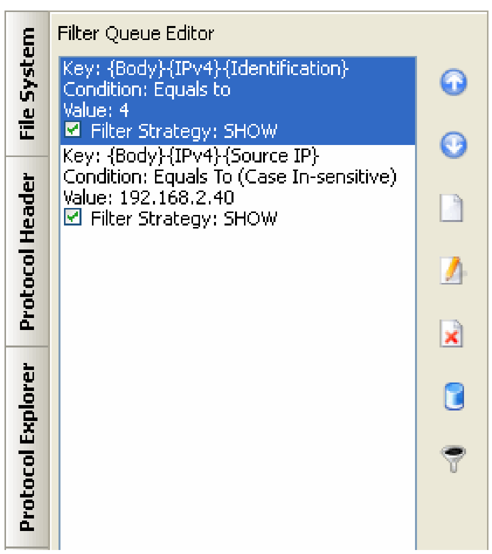


FIGURE 8-44. Multiple Rules Defined for Protocol Header Values

9 EXata File Editor

This chapter describes EXata File Editor, which is a text editing tool that can be used for viewing and editing files associated with a scenario File Editor provides helpful utilities, like auto-completion of parameters and values and syntax highlighting, which makes it particularly useful for viewing and editing scenario configuration (.config) files.

This application can be launched from the Architect view either by clicking the **Open Scenario in File Editor** icon on the Run Toolbar or by clicking the **File Editor** button on the Components Toolbar.

The major sections of this chapter are:

- Components of File Editor
- Using File Editor

How to Get to File Editor

To switch to the File Editor component of EXata GUI, click the 

button in the **Components** toolbar. You can open and edit any text file associated with any scenario.

If a scenario is open in Architect, you can switch directly to File Editor and view the scenario configuration (.config) file for that scenario by clicking the **View Scenario in File Editor**  button on the **Run** toolbar or by selecting the **View Scenario in File Editor command from the Tools menu.**

FIGURE 9-1. Tools Menu - View Scenario in File Editor

Note: Clicking the **View Scenario in File Editor**  button or selecting the **View Scenario in File Editor command** opens the scenario configuration (.config) file in the text editing tool selected in the **Preferences** dialog (see Section 3.1.2.3). By default, File Editor is used as the text editing tool.

9.1 Components of File Editor

This section provides an overview of the different components of File Editor (see Figure 9-2).

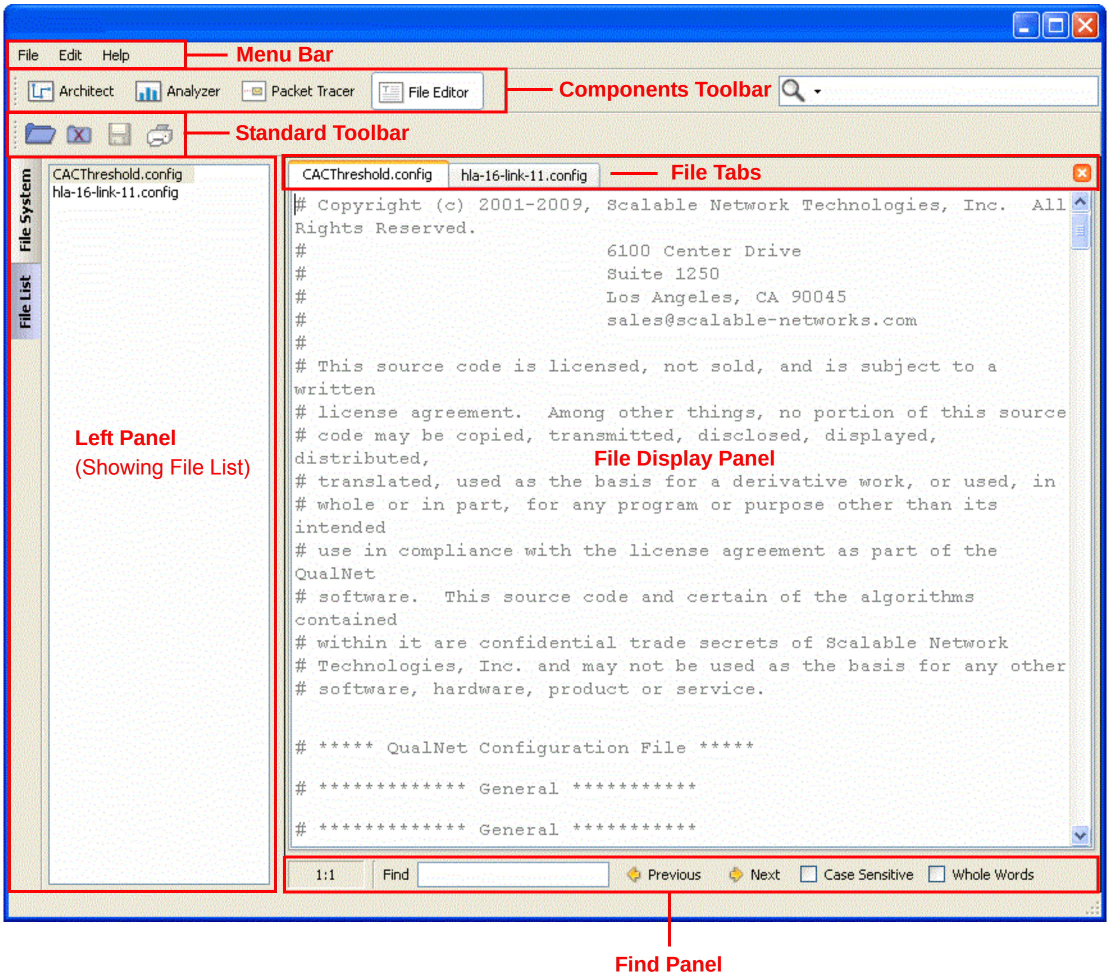


FIGURE 9-2. Components of File Editor

9.1.1 Menu Bar

This section describes the menus available from the **Menu** bar.

9.1.1.1 File Menu

The **File** menu provides the following commands for file operations:

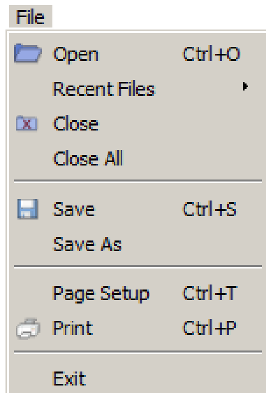


FIGURE 9-3. File Menu

See Section 3.1.1.1 for a description of the **File** menu commands.

9.1.1.2 Edit Menu

The **Edit** menu provides the following commands for editing file editing:

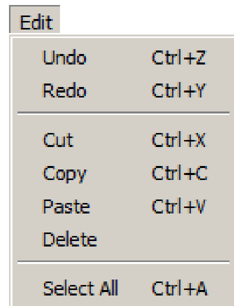


FIGURE 9-4. Edit Menu

Command	Description
Undo	Undoes the most recent action.
Redo	Redoes the most recent undone action.
Cut	Cuts the selected text.
Copy	Copies the selected text.
Paste	Pastes the previously cut or copied contents.
Delete	Deletes the selected text.
Select All	Selects the entire file.
Insert Unicode control character	Displays a list of unicode control characters. If a character from this list is selected, it will be inserted in the file at the cursor's position.

9.1.1.3 Help Menu

See Section 3.1.1.6 for a description of the **Help** menu.

9.1.2 Standard Toolbar

The **Standard** toolbar contains buttons (from left to right) to open, delete, save and print files.

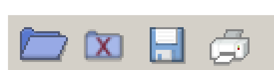


FIGURE 9-5. Standard Toolbar

9.1.3 Left Panels

The following panels are available to the left of the File Editor window:

- File System Panel
- File List Panel

Note: These two panels occupy the same space and only one of them can be opened at any time.

9.1.3.1 File System Panel

See Section 3.1.3.1 for a description of the **File System** panel.

9.1.3.2 File List Panel

This panel lists the files currently open in File Editor. Clicking on a filename displays it in the **File Display Panel**.

You can close a file (or all open files) by right-clicking on the file name in the **File List Panel** and selecting **Close Selected** (or **Close All**).

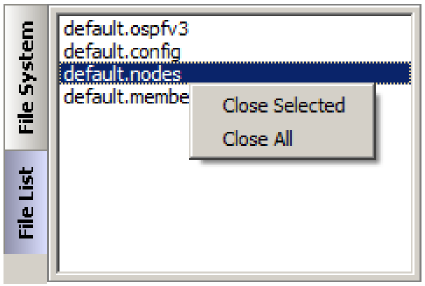


FIGURE 9-6. File List Panel

9.1.4 Find Panel

The **Find Panel** provides search functionality, which is used to find text strings in the active file. The functions are described in the table below.

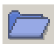
FIGURE 9-7. Find Panel

9.2 Using File Editor

This section describes how to open and edit files in File Editor.

9.2.1 Opening Files

Files can be opened in File Editor in the following ways:

- From **File** Menu or **Standard** Toolbar of File Editor: Select the **File > Open** command or click the **Open**  button on the **Standard** Toolbar. Select a file from the file browser that is opened.
- From **File System** Panel of Architect, Analyzer, Packet Tracer, or File Editor: Right-click on a file name and select **Edit as Text**. In addition, any file which does not have the extension .config, .anim, or ,stat can be opened in File Editor by double-clicking on the file name.
- From Design Mode of Architect: If a scenario is open in Design mode of Architect, then to open the scenario configuration (.config) file for that scenario in File Editor, click the **View Scenario in File Editor**  button on the **Run** toolbar or select the **Tools > View Scenario in File Editor command.**

  Note that for this option to open files in File Editor to work, File Editor must be selected as the default text editing tool (see Section 3.1.1.2**).**

File Editor uses syntax highlighting when displaying scenario configuration (.config) files: parameters, qualifiers, comments, values, and value options (i.e., values that are selected from a list) are displayed in different colors (see Figure 9-8).

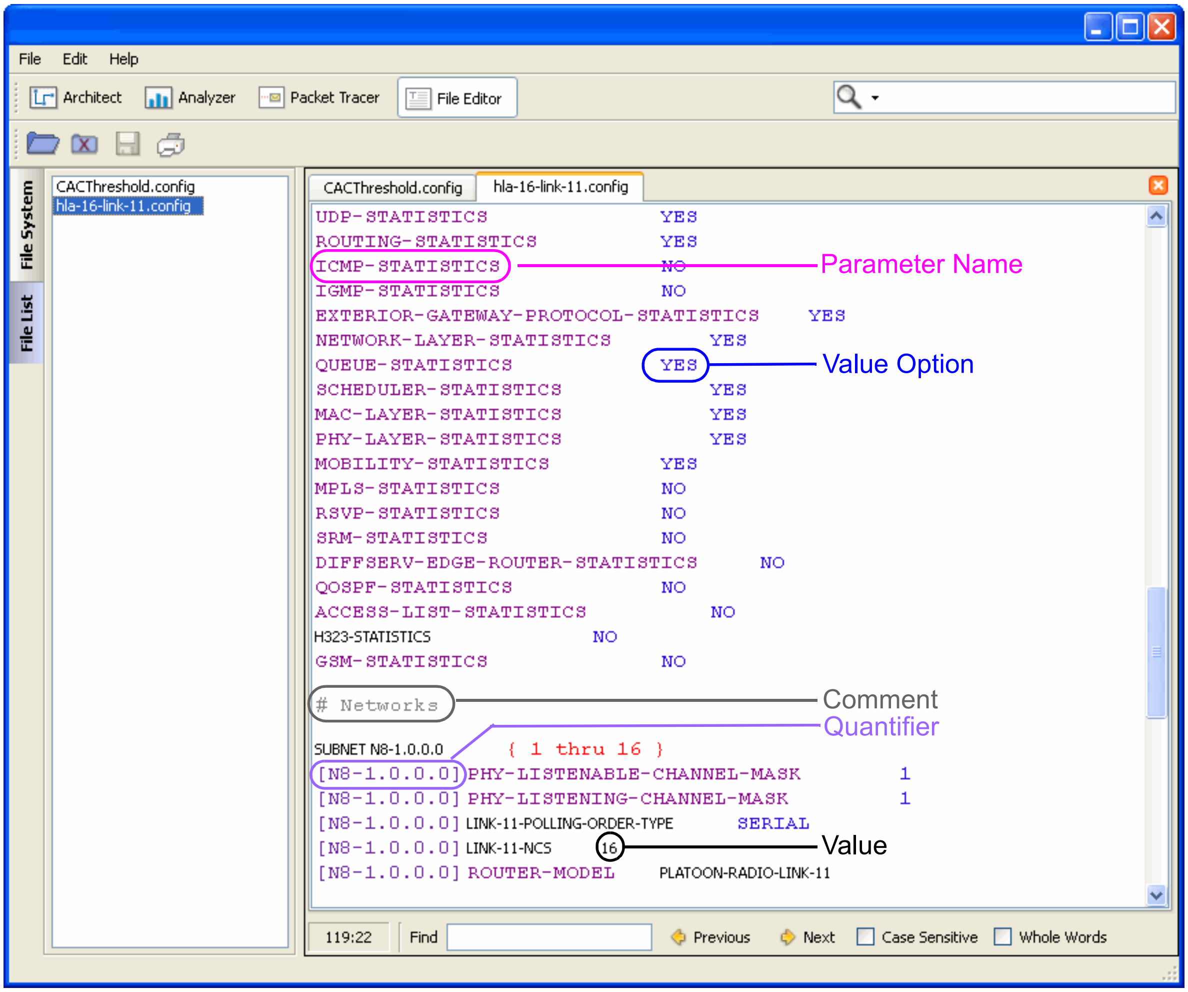


FIGURE 9-8. Syntax Highlighting in File Editor

9.2.2 Editing Files

Files can be edited in File Editor in the same way as any standard text editing tool.

In addition, File Editor provides a convenient auto-completion feature for entering parameter names and value options in scenario configuration (.config) files, which works as follows:

After you have typed three or more characters of a parameter name, a drop-down list is displayed listing all parameters starting with the character string entered until that point. You can select a parameter from the list or continue to type the next characters of the parameter name. As you continue to type, the list gets truncated to match the part of parameter name entered.

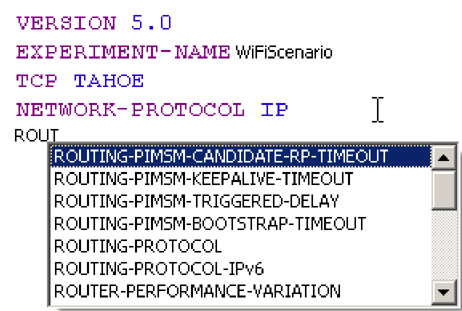


FIGURE 9-9. Auto-completion of Parameter Names

After a parameter name has been entered (by typing or by selection), enter a space. If the parameter value is of type List, i.e., the parameter value can be one of several enumerated items (see Section 4.1.1), then the list of options for that parameter is displayed. You can select a value from the list or start typing the option. As you continue to type, the list of options get truncated to match the part of option entered.

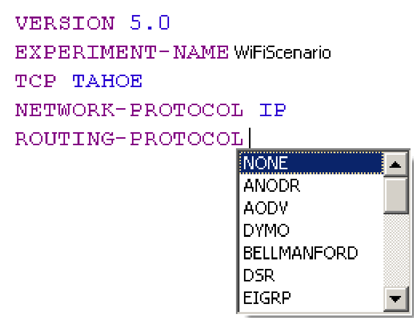


FIGURE 9-10. Auto-completion of Parameter Values

A License and Libraries Information

A valid license is required to run EXata. The supported license schemes are described in Section A.1.

To use a library, it must be activated (i.e., the EXata executable file must be compiled with it) and you must have a license that enables it.

Libraries Enabled by an Evaluation License: A typical evaluation license enables all model libraries for the evaluation period (which is typically 30 days). The TIREM Advanced Propagation library cannot be enabled by an evaluation license.

Libraries Enabled by a Commercial or University License: A commercial or university license enables the base version of EXata and all additional libraries purchased by the user. The base version of EXata includes the following model libraries:

- Developer Model Library
- Multimedia and Enterprise Model Library
- Network Management Model Library
- Wireless Model Library

To view the list of libraries enabled by your license, see Section A.3 and Section A.4.

Model Libraries Precompiled with EXata

The source code, scenarios, and documentation for the following model libraries are included in the EXata distribution. In addition, the precompiled executable files included in the distribution (see Section 1.4.3) have been compiled with these libraries:

- Advanced Wireless Model Library
- Cellular Model Library
- Cyber Model Library
- Developer Model Library (including STK interface)
- Federation Interfaces Library
- LTE Model Library
- Multimedia and Enterprise Model Library
- Network Management Model Library
- Sensor Networks Model Library
- UMTS Model Library

- Urban Propagation Model Library
- Wireless Model Library

A.1 Types of Licenses

This section describes the types of license schemes supported by EXata.

EXata supports three types of licenses: node-locked, floating (also known as client-server), and dongle.

- **Node-locked License**: This type of license is tied to a specific machine (which is identified by its MAC address) and cannot be used on a different machine. Use this license scheme if you plan on using EXata always on the same machine. This type of license does not require a license server and is very easy to install.
- **Floating License**: This type of license requires a license server. A license server provides a floating license that can be served to a number of client machines that are within the licensed IP range. Any machine on the sanctioned network can use a floating license, if it can connect to the license server.

  Note: When using a license server, each client machine must have EXata installed and must have a client license file. The client license file directs EXata to checkout the actual license from the server. The license is returned to the server upon completion of use. Only the purchased number of licenses can be in use at any given time, but they are not tied to a particular machine.

- **Dongle License**: This is a special type of license that is associated with an external USB device. In order to use this license, the associated USB device must be plugged into your computer. This is restricted scheme which is primarily used for classified environments and requires approval by the Scalable Network Technologies' sales department

A.2 Installing Licenses

This section describes how to install EXata licenses.

Installing Node-locked Licenses

Copy the license file into EXATA_HOME\license_dir, where EXATA_HOME is the directory where EXata is installed. The name for a node-locked license file is of the form EXata-<type>-<date>.lic, where <type> is the type of license and <date> is the date when support for the product expires. <type> is one of the following: Commercial-node-locked, University-node-locked, Evaluation, or Temporary-node-locked. <date> is in the format YYYY-MM-DD-HH-MM-SS. For example, the license file may be named EXata-Commercial-node-locked-2013-05-19-01-19-52.lic.

Be sure to check the file extension on your license file. It should end in ".lic". Depending on your system and method of copying, you may end up with an extra extension, such as EXata-Commercial-node-locked-2013-05-19-01-19-52.lic.txt, which will not work.

Installing Floating Licenses

If you have a floating license, then you will need to install a license file on each machine on which you wish to run EXata and on the license server. The same license file is used on the clients and the server. The name for a floating license file is of the form EXata-<type>-<date>.lic, where <type> is the type of license and <date> is the date when support for the product expires. <type> is one of the following: Commercial-floating or University-floating. <date> is in the format YYYY-MM-DD-HH-MM-SS. For example, the license file may be named EXata-Commercial-floating-2013-05-19-01-19-52.lic.

Be sure to check the file extension on your license file. It should end in ".lic". Depending on your system and method of copying, you may end up with an extra extension, such as EXata-Commercial-floating-2013-05-19-01-19-52.lic.txt, which will not work.

- **Client License File**: Copy the license file into EXATA_HOME\license_dir, where EXATA_HOME is the directory where EXata is installed.
- **Server License File**: Activate the server license on the qlm license server with the license file. The instructions for starting the license server with the license file are located on your EXata license download page. Use your EXata account username and password to access this page.

Installing Dongle Licenses

To use a dongle license, do the following:

- Copy the license file into EXATA_HOME\license_dir, where EXATA_HOME is the directory where EXata is installed. (This step is similar to installing a node-locked license).
- Install the dongle device driver for your system. The instructions for installing the driver will be provided to you by email.
- Plug the dongle device into your system.

A.3 License and Libraries Information from Command Line

This section describes how to obtain information on your license and libraries installed on your system from the command line.

To obtain license and libraries information, open a command window and type the following command:

```
exata -print_libraries
```

This command prints license and libraries information about your system in four columns. A typical output of this command as shown in Figure A-1.

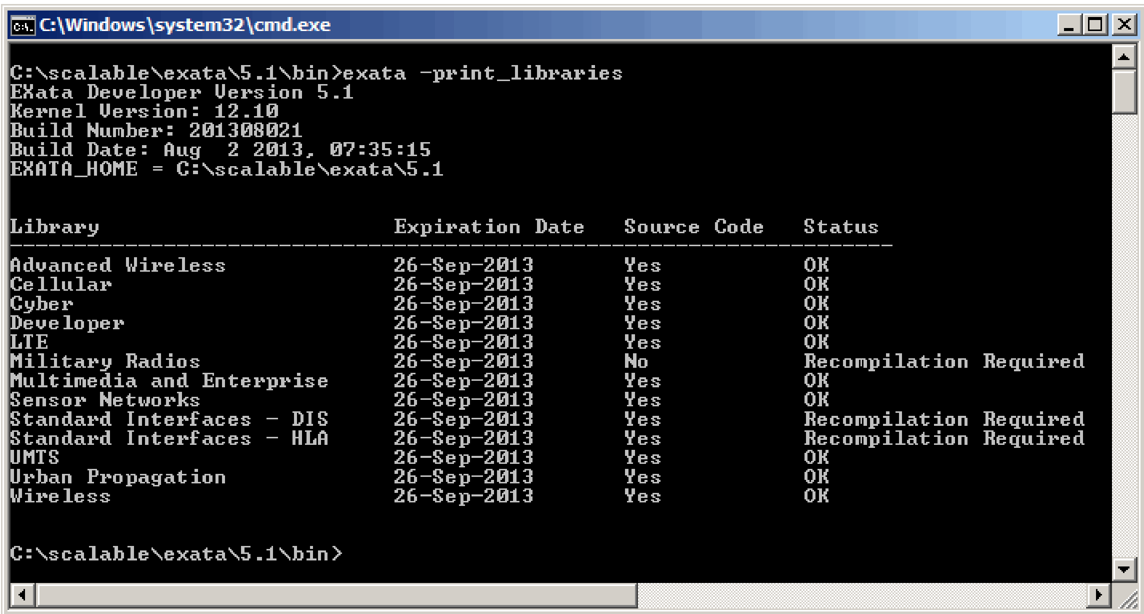


FIGURE A-1. License and Libraries Information from the Command Line

The columns of the output contain the following information:

- **Library**: This column displays the library name. If the library has multiple modules which can be enabled individually, then each module will be listed separately with the library name in the following format: <library-name> - <module name>.
- **Expiration Date**: This column displays the expiration date of the license for the library.
- **Source Code**: This column indicates whether or not the source code of the library is present.
- **Status**: This column indicates the status of the library. The status can be: OK, License Expired, or Recompilation Required.
  - **OK**: You have a valid license for the library and your EXata executable has been compiled with it. Thus, you can use the library without having to recompile EXata.
  - **License Expired**: You had a license for the library, but it has since expired.
  - **Recompilation Required**: You have a valid license for the library but your EXata executable has not been compiled with it. You will need to recompile EXata in order to use models in the library. If the source code for the library is present (which is indicated in the third column), you can enable the library and recompile EXata.

A.4 License and Libraries Information from the GUI

The **License and Libraries** dialog of the EXata GUI enables users to view information about the license, replace the license file, view the status of libraries, and get help should license problems be encountered.

You can launch the **License and Libraries** dialog from Architect, Analyzer, Packet Tracer, or File Editor by selecting **License and Libraries** from the **Help** menu.

The **License and Libraries** dialog also open automatically when you start the EXata GUI if your license file is missing, expired, or invalid.

The **License and Libraries** dialog has two tabs:

- Status
- Troubleshooting

A.4.1 Status Tab

The **Status** tab (see Figure A-2) displays information about the license and libraries installed on your system.

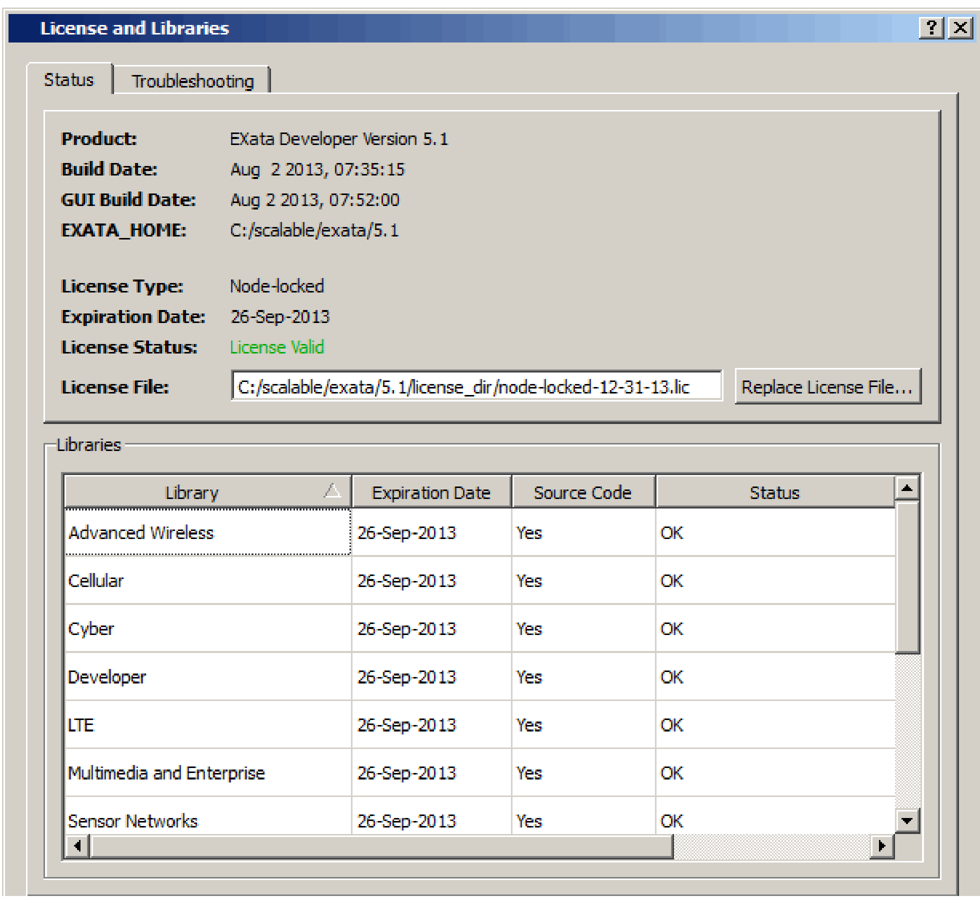


FIGURE A-2. License and Libraries Dialog - Status Tab

The top part of the dialog displays the following information:

- Build date (date the EXata distribution was created)
- EXATA_HOME (value of the EXATA_HOME environment variable)
- License type and expiration date (date of expiration of the base license)
- License status

The **License File** field shows the location of the current license file. You can select a new license file by clicking the **Replace License File…** button, browsing to the location of the new license, and clicking the **Open** button. This will copy the license file into the appropriate directory.

Note: If you replace the license file, you must restart the GUI for the new license to take effect.

The bottom part of the dialog displays information about the libraries installed on your system in four columns: **Library**, **Expiration Date**, **Source Code**, and **Status**. See Section A.3 for a description of these columns.

A.4.2 Troubleshooting Tab

The **Troubleshooting** tab indicates the license status, provides troubleshooting details, and provides the ability to generate a detailed report about your system.

A.4.2.1 Troubleshooting Details

If there is a problem with your license (as indicated by the **License Status** field), the **Troubleshooting Details** window displays detailed error messages and, possibly, steps to resolve the problem.

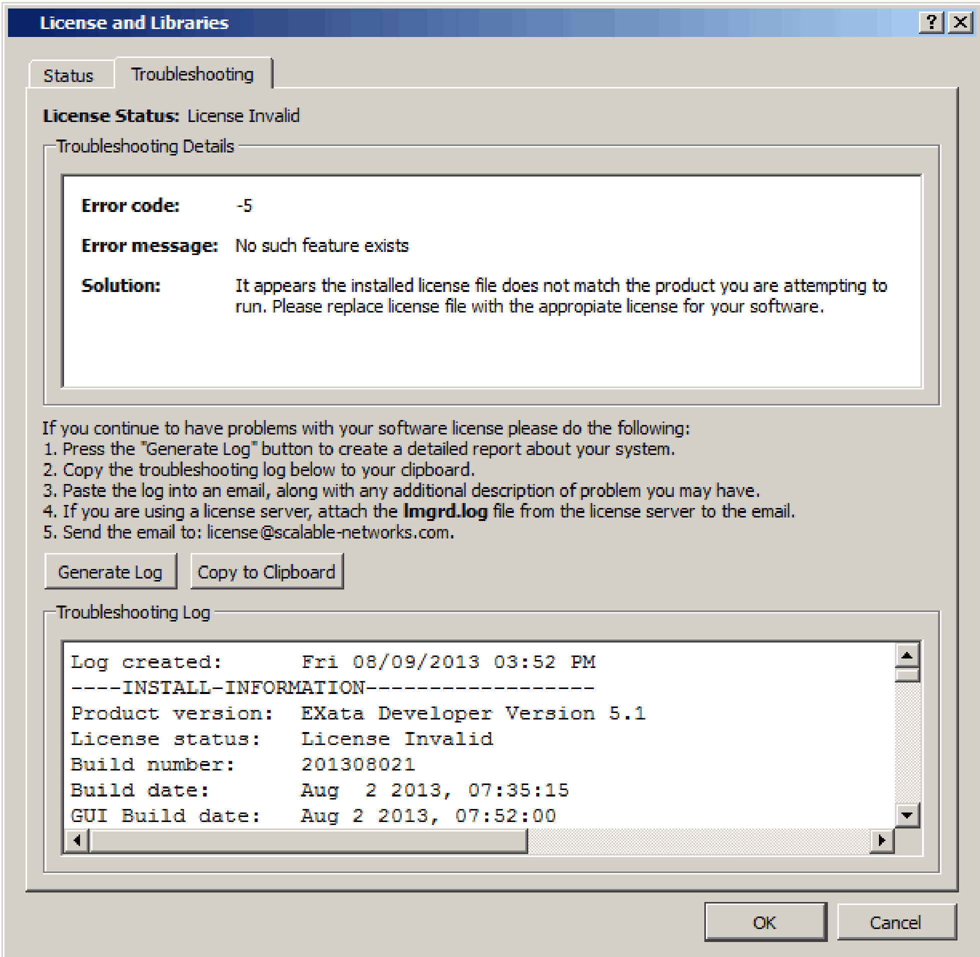


FIGURE A-3. License and Libraries Dialog - Troubleshooting Tab

A.4.2.2 Troubleshooting Log

The **Troubleshooting Log** is a detailed log report, which can help the support team at Scalable Network Technologies to diagnose your specific license error.

To get help on your license error, do the following:

1. Click the **Generate Log** button; a detailed log report will be displayed in the window.
2. Click the **Copy to Clipboard** button to copy the contents of the log to the clipboard.
3. Paste the contents of the clipboard into an email, and send it to license@scalable-networks.com where the support team can diagnose the problem and suggest solutions.

A.5 Helpful Links

For help with license issues, visit the FAQ page:

http://support.scalable-networks.com/categories/20014366-license-support

B Online Help

The Online Help interface can be used to search for specific topics that are documented in the online manuals. Online help is accessed by using the **Keyword Search** text box, from the toolbars (only in Architect), or from any of the properties editors.

B.1 Using Keyword Search Help

The Search Text Box is located in the top right corner of all GUI components.

As you start typing keywords or search topics in the search text box, a list of keywords or help topics starting with the text entered is displayed (see the example in Figure B-1).

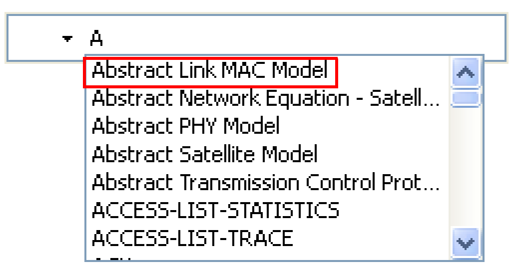


FIGURE B-1. Search Text Box

Example:

In the following example, we selected **Abstract Link MAC Model** from the pull-down list and then pressed **Enter**. Figure B-2 displays the search results for the Abstract Link MAC Model.

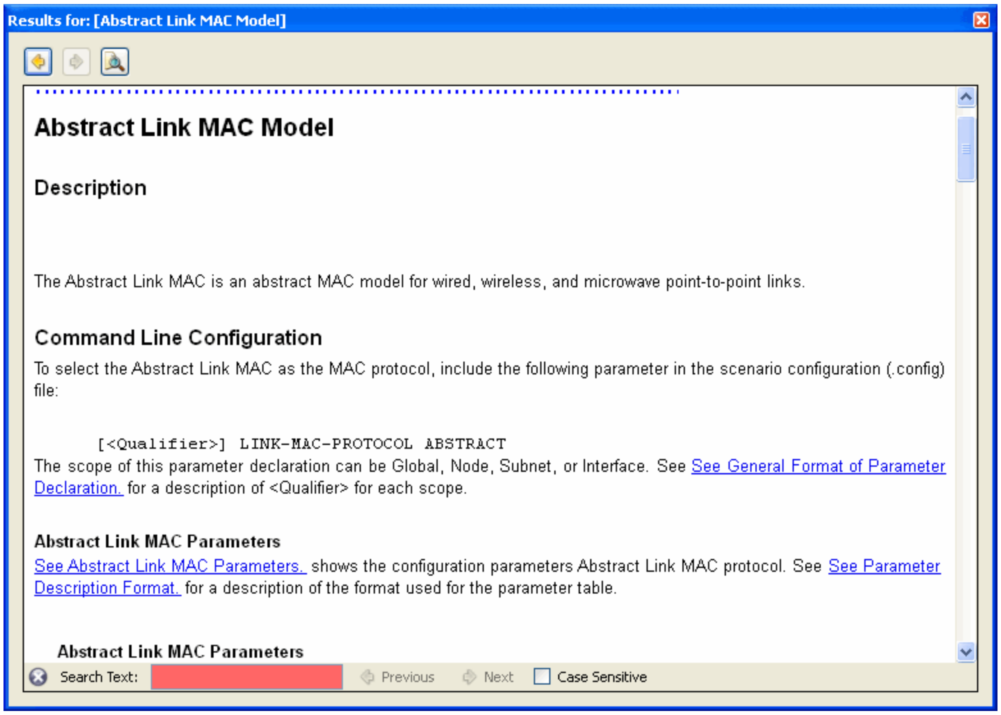


FIGURE B-2. Search Results for Abstract Link MAC Model

Note: In the results window above, using the search text feature, you can search for key words within the selected help topic using the Previous, Next, and Case Sensitive buttons.

B.2 Using Toolbar Help

The Online Help Interface can be accessed from any of the toolbars in Architect by right-clicking on a specific tool button.

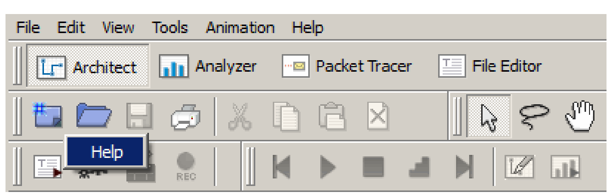
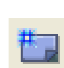
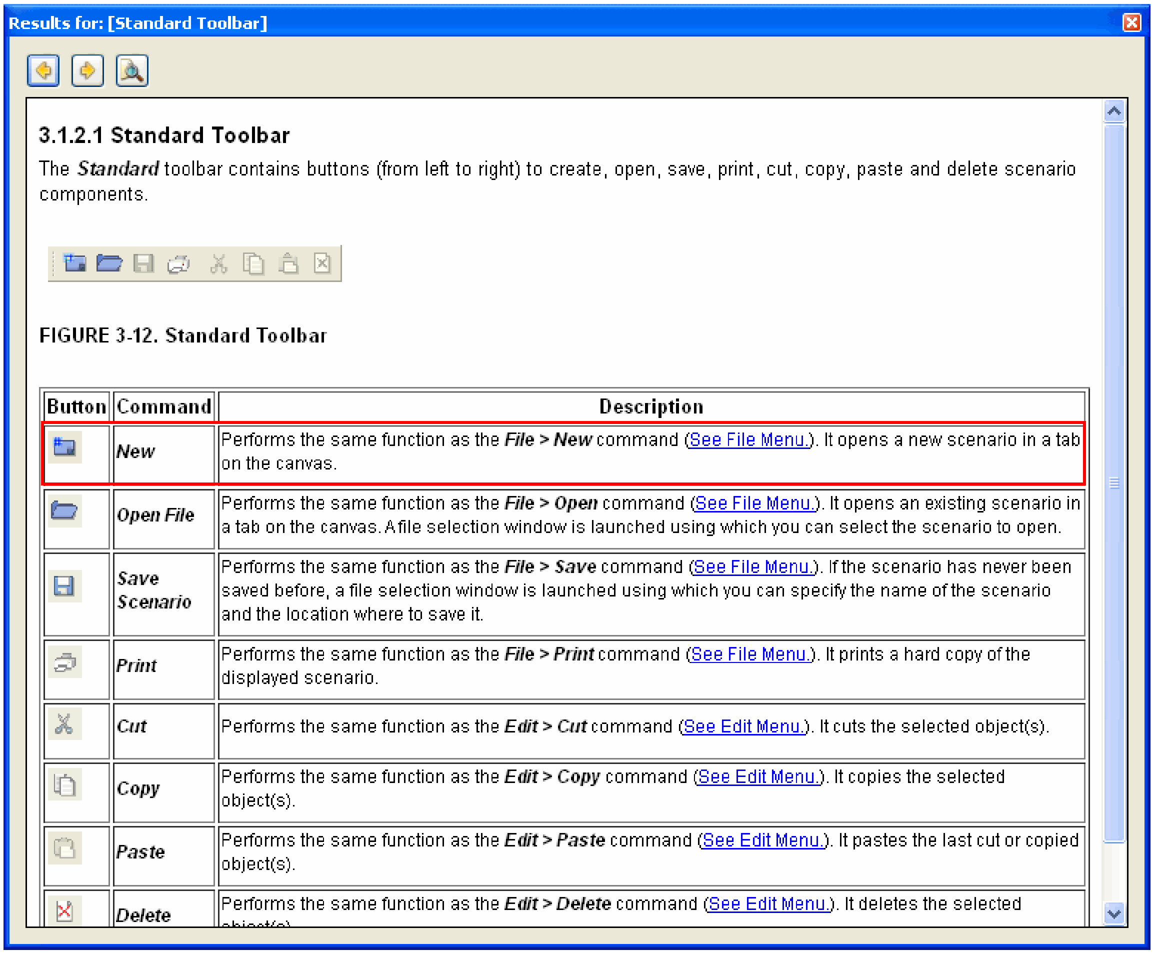


FIGURE B-3. Toolbar Help Buttons

Example:

In the following example, we right-clicked on the **New**  button from the Standard Toolbar. Figure B-4 displays the search results for the **New** button.



Results for: [Standard Toolbar]

3.1.2.1 Standard Toolbar

The *Standard* toolbar contains buttons (from left to right) to create, open, save, print, cut, copy, paste and delete scenario components.

FIGURE 3-12. Standard Toolbar

Button	Command	Description
	New	Performs the same function as the ***File > New*** command (See File Menu.). It opens a new scenario in a tab on the canvas.
	Open File	Performs the same function as the ***File > Open*** command (See File Menu.). It opens an existing scenario in a tab on the canvas. A file selection window is launched using which you can select the scenario to open.
	Save Scenario	Performs the same function as the ***File > Save*** command (See File Menu.). If the scenario has never been saved before, a file selection window is launched using which you can specify the name of the scenario and the location where to save it.
	Print	Performs the same function as the ***File > Print*** command (See File Menu.). It prints a hard copy of the displayed scenario.
	Cut	Performs the same function as the ***Edit > Cut*** command (See Edit Menu.). It cuts the selected object(s).
	Copy	Performs the same function as the ***Edit > Copy*** command (See Edit Menu.). It copies the selected object(s).
	Paste	Performs the same function as the ***Edit > Paste*** command (See Edit Menu.). It pastes the last cut or copied object(s).
	Delete	Performs the same function as the ***Edit > Delete*** command (See Edit Menu.). It deletes the selected

FIGURE B-4. Help for Create a New Scenario Button

B.3 Using Properties Editors Help

The Online Help interface can be accessed from all the properties editors by either clicking the **Help** 

icon in the top-right corner or by right-clicking on a specific property in the properties editor (see Figure B-5).

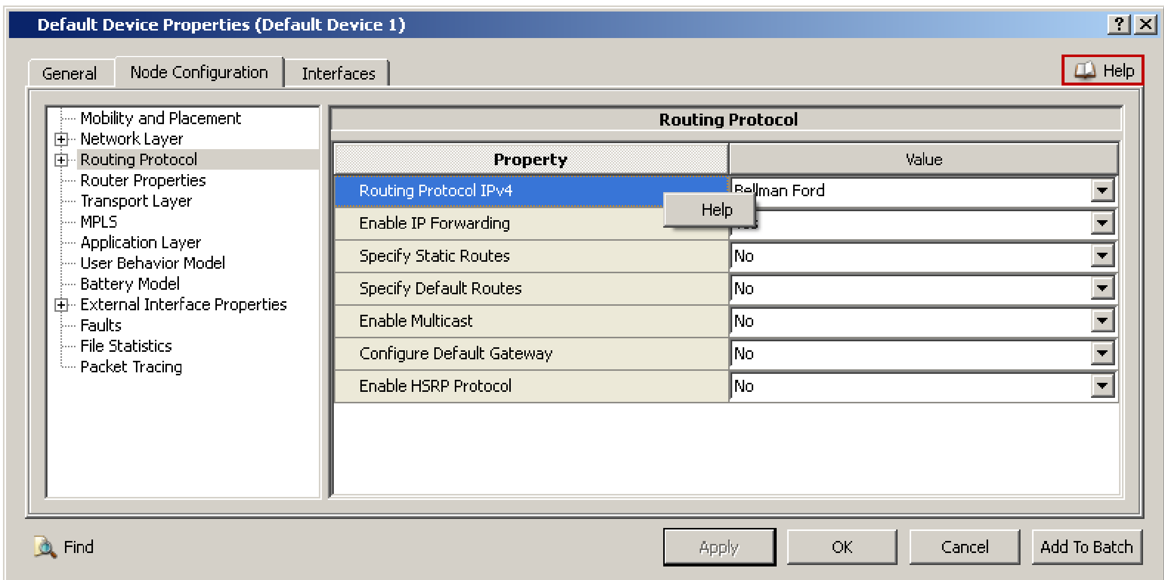


FIGURE B-5. Help from Properties Editor Help Icons

Example:

In the following example, we clicked on the Help icon, which displays information about the active properties editor (see Figure B-6). Right-clicking on an individual property displays information about that property (see Figure B-7).

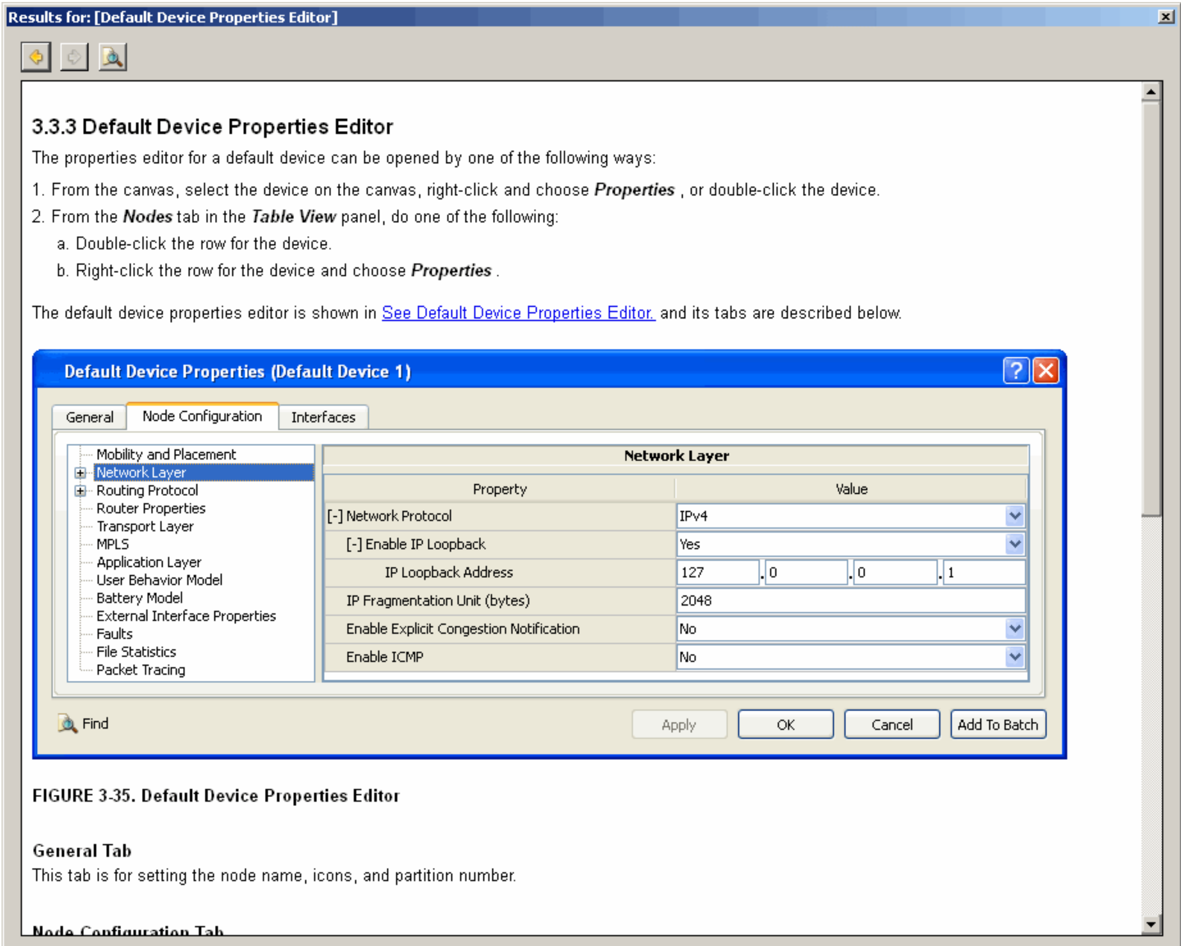
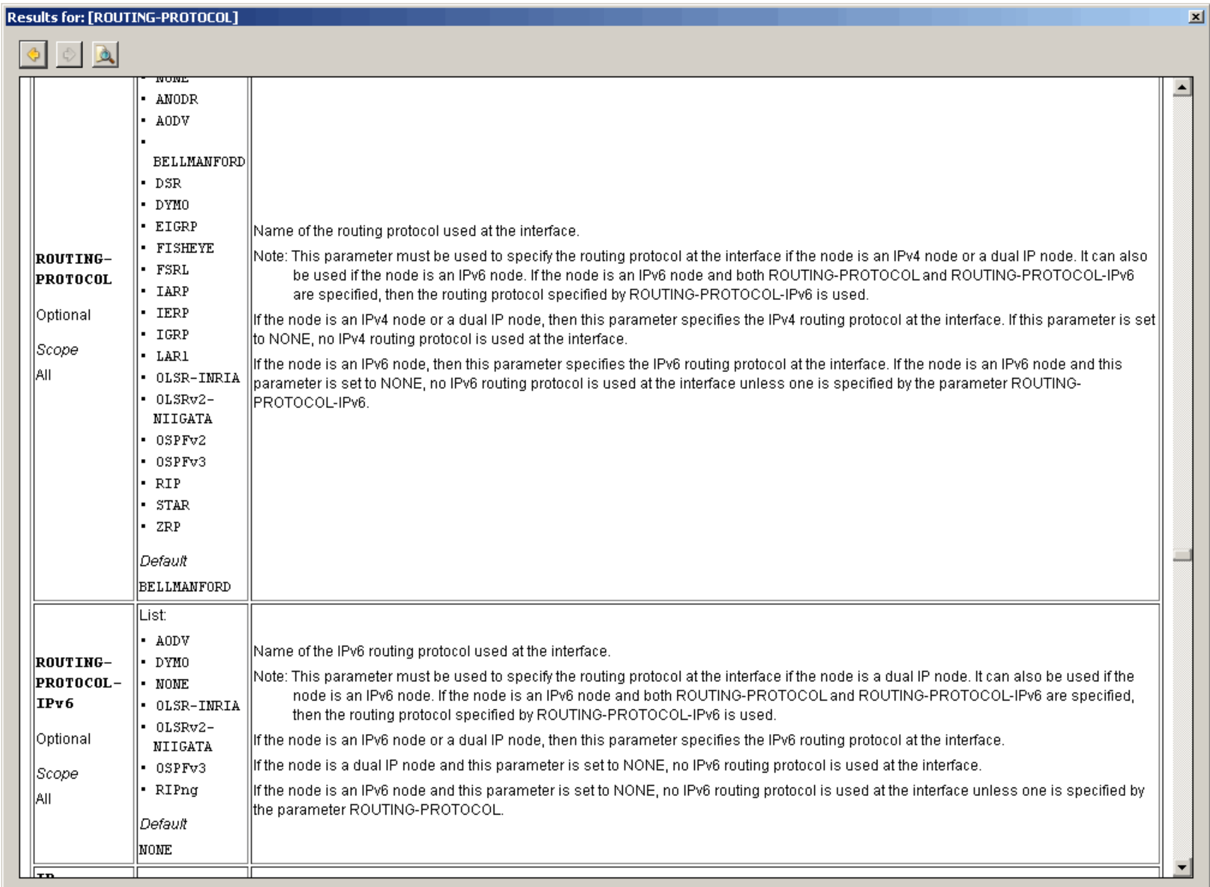


FIGURE B-6. Help on Default Device Properties Editor



Results for: [ROUTING-PROTOCOL]

ROUTING-PROTOCOL Optional *Scope* All	• NONE • ANODR • AODV • BELLMANFORD • DSR • DYMO • EIGRP • FISHEYE • FSRL • IARP • IERP • IGRP • LAR1 • OLSR-INRIA • OLSRv2-NIIGATA • OSPFv2 • OSPFv3 • RIP • STAR • ZRP *Default* BELLMANFORD	Name of the routing protocol used at the interface. Note: This parameter must be used to specify the routing protocol at the interface if the node is an IPv4 node or a dual IP node. It can also be used if the node is an IPv6 node. If the node is an IPv6 node and both ROUTING-PROTOCOL and ROUTING-PROTOCOL-IPv6 are specified, then the routing protocol specified by ROUTING-PROTOCOL-IPv6 is used. If the node is an IPv4 node or a dual IP node, then this parameter specifies the IPv4 routing protocol at the interface. If this parameter is set to NONE, no IPv4 routing protocol is used at the interface. If the node is an IPv6 node, then this parameter specifies the IPv6 routing protocol at the interface. If the node is an IPv6 node and this parameter is set to NONE, no IPv6 routing protocol is used at the interface unless one is specified by the parameter ROUTING-PROTOCOL-IPv6.
ROUTING-PROTOCOL-IPv6 Optional *Scope* All	List: • AODV • DYMO • NONE • OLSR-INRIA • OLSRv2-NIIGATA • OSPFv3 • RIPng *Default* NONE	Name of the IPv6 routing protocol used at the interface. Note: This parameter must be used to specify the routing protocol at the interface if the node is a dual IP node. It can also be used if the node is an IPv6 node. If the node is an IPv6 node and both ROUTING-PROTOCOL and ROUTING-PROTOCOL-IPv6 are specified, then the routing protocol specified by ROUTING-PROTOCOL-IPv6 is used. If the node is an IPv6 node or a dual IP node, then this parameter specifies the IPv6 routing protocol at the interface. If the node is a dual IP node and this parameter is set to NONE, no IPv6 routing protocol is used at the interface. If the node is an IPv6 node and this parameter is set to NONE, no IPv6 routing protocol is used at the interface unless one is specified by the parameter ROUTING-PROTOCOL.

FIGURE B-7. Help on a Specific Property

C Advanced Run Settings

Using the **Run Settings** dialog, the EXata GUI can be configured to run a simulation in the following modes:

- locally, using a single processor
- locally, in parallel mode on a shared-memory multi-processor system
- locally, in parallel mode on a distributed platform (refer to *EXata Distributed Reference Guide* for the requirements for running EXata on a distributed platform)
- remotely, with the simulator running on a server in any of the above modes

C.1 Local Execution

To set the run mode for local execution in EXata GUI, do the following:

1. Open the Run Settings window (see Figure C-4) by pressing the **Run Settings** button (which is to the left of the **Run Simulation** button).

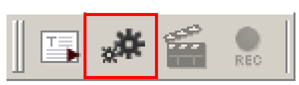


FIGURE C-1. Run Settings Button

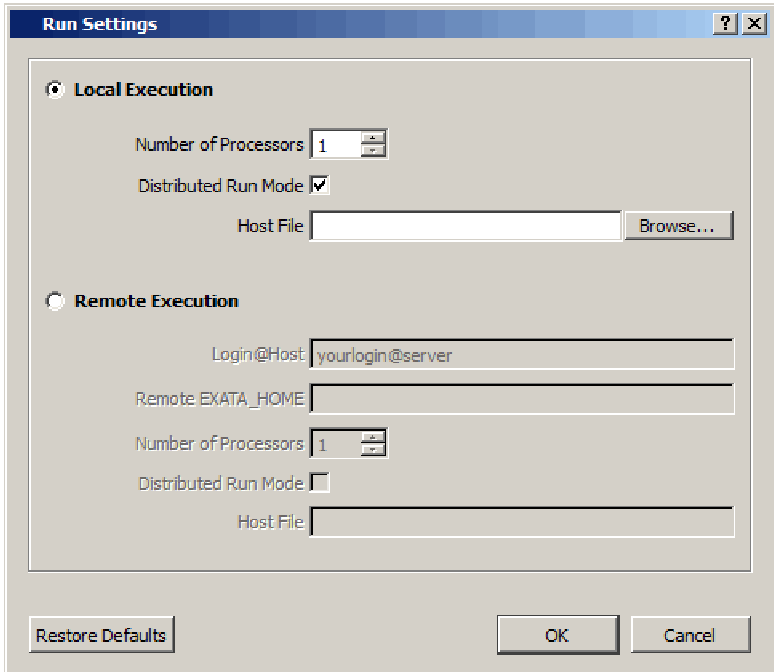


FIGURE C-2. Run Settings for Local Execution

2. Select the **Local Execution** radio button.
3. Set **Number of Processors** to the number of processors on which to run EXata.
4. Check the **Distributed Run Mode** option to run EXata on a distributed platform, and in the **Host File** field that is displayed, enter the name of the host file, which lists the computers to be used. Refer to *EXata Distributed Reference Guide* for details of the host file.

> **Note:** The **Distributed Run Mode** option is available only if EXATA_HOME/bin contains a distributed-mode executable (exata.mpi). Refer to *EXata Distributed Reference Guide* for details of compiling EXata on a distributed platform.

5. Press **OK** to save the changes.

These run settings will be used for all subsequent simulations.

C.2 Remote Execution

This section describes the system requirements and setup for running EXata remotely and how to set the run mode for remote execution in EXata GUI.

C.2.1 System Requirements

To run EXata remotely, the following are required:

- An EXata installation on a server computer capable of accepting OpenSSH connections.
- An EXata installation on a client computer capable of starting OpenSSH sessions.

  Most Linux configurations are capable of doing this out of the box.

  On Windows, the following are required:

  - Cygwin (installed in the default location, C:/cygwin). For information on installing Cygwin packages, refer to http://www.cygwin.com/cygwin-ug-net/setup-net.html.
  - OpenSSH package (typically found within the **Net** package category)
- Bash shell must be associated with the login account on the server computer.

C.2.2 Setup for Remote Execution

EXata uses SSH copying, port forwarding, and remote shell commands to allow for remote execution to be established between the client and the server. In order for the client and the server to be able to talk to other, they must be configured with key based access so that the client can connect to the server without having to prompt for the server password on each command. This can be manually configured, but for easy setup, you may execute script `$EXATA_HOME/bin/setupRemoteExec.sh`.

The setupRemoteExec.sh script configures a predefined authorization for a client to access a server by utilizing a secured authorized key. A session between client and server remains secure using this mechanism, but authorization is granted without a password prompt. This is necessary so that the GUI is able to launch repeated calls to the simulator and transfer files, without having to prompt the user or save a user's password in the system settings. OpenSSH is also utilized as a transport mechanism during animation of a remote simulation, to relay inter-process communications between the GUI and the simulator over the "interactive" port (port 4000). Normally the interactive port is only accessed locally, but OpenSSH enables remote access using the "GatewayPorts" feature of OpenSSH. It is thus essential that GatewayPorts is not disabled on the server system.

To run this script, type the following commands at the command prompt (or in Cygwin on Windows) on the local client computer:

```
cd $EXATA_HOME/bin
chmod +x exata.remote.sh
chmod +x setupRemoteExec.sh
./setupRemoteExec.sh  yourlogin@hostname
```

where

yourlogin	Your login name on the server.
hostname	Name or IP address of the server.

You will be prompted for your password multiple times and will be asked permission to allow access to the server. Follow all the prompts and agree to grant permission. After completing the steps, you should be able to run any OpenSSH command to the server without being prompted for as password. For example, the following command executed on the client computer should allow immediate terminal access to the server without a password prompt:

```
ssh  yourlogin@hostname
```

Notes: 1. If you receive the following error:

```
ssh-keygen: command not found
```

It is likely that ssh is not properly installed on the client. Reinstalling may correct the issue.

2. If you receive the following error:

```
Connection refused
```

It is likely that sshd has not been activated on the server. Consult your system administrator to have the server's SSH daemon process accept connections from your client.

C.2.3 Run Mode Settings for Remote Execution in GUI

To set the run mode for remote execution in EXata GUI, do the following:

1. Open the Run Settings window (see Figure C-4) by pressing the **Run Settings** button (which is to the left of the **Run Simulation** button).

FIGURE C-3. Run Settings Button

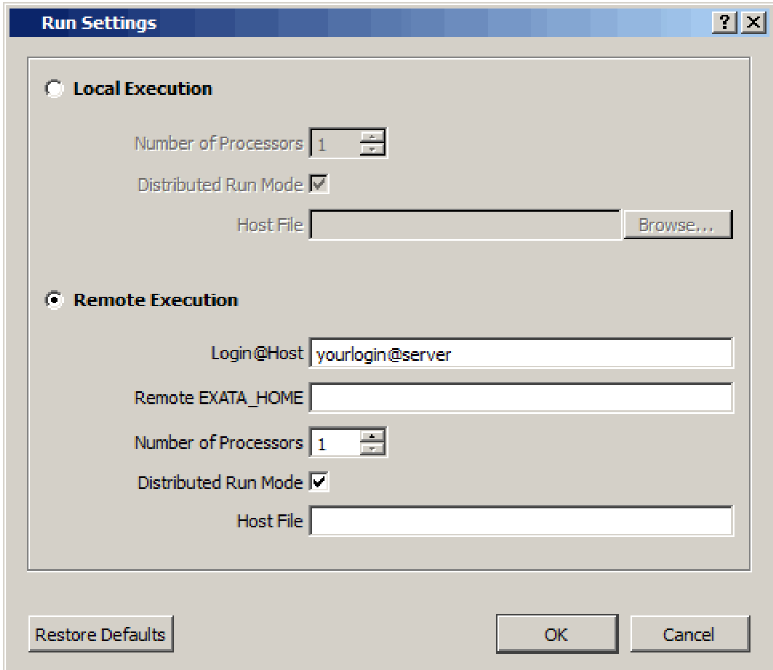


FIGURE C-4. Run Settings for Remote Execution

2. Select the **Remote Execution** radio button.
3. Enter the login and hostname in the **Login@Host** field.
4. Set **Remote EXATA_HOME** to the directory on the remote machine where EXata is installed.
5. Set **Number of Processors** to the number of processors on which to run EXata.
6. Check the **Distributed Run Mode** option to run EXata on a distributed platform, and in the **Host File** field that is displayed, enter the name of the host file, which lists the computers to be used. Refer to *EXata Distributed Reference Guide* for details of the host file.

Note: The **Distributed Run Mode** option is available only if EXATA_HOME/bin contains a distributed-mode executable (exata.mpi). Refer to *EXata Distributed Reference Guide* for details of compiling EXata on a distributed platform.

7. Press **OK** to save the changes.

These run settings will be used for all subsequent simulations.

Note: Before running EXata on a remote machine for the first time, you must run the setupRemoteExec.sh script, as described in Section C.2.2.

If EXata is run in remote mode, information about the connection is printed in the output window prior to running the simulation. Figure C-5 shows an example of EXata simulation run remotely on a machine with two processors.

```
Executing EXata via remote ssh on host:myaccount@snt-dual-xeon3

If this is the first time you are connecting to this server, then you must
execute the following command on the client before running the simulation:
  c:/scalable/exata/5.1/bin/setupRemoteExec.sh  myaccount@snt-dual-xeon3

Configuration File: default.config
Number of Processors: 2
Remote EXATA_HOME: /home/myaccount/scalable/exata/5.1
Local Working Directory: /cygdrive/C/scalable/exata/5.1/scenarios/default
Remote Working Directory: /home/myaccount/.remoteEXataDir
Interactive Port: 4000
Remote Run Mode: Normal
Remote Simulator Command: /home/myaccount/scalable/exata/5.1/bin/exata default.config -np 2 -interactive
localhost 4000

Sending files to host

Starting EXata simulation
EXata Version 5.1
EXATA_HOME = /home/myaccount/scalable/exata/5.1
Attempting license checkout (should take less than 2 seconds) ... success.

Partition 0, Node 1 (126.02, 88.53, 0.00).
Partition 0, Node 2 (485.56, 214.68, 0.00).
```

FIGURE C-5. Running EXata in Remote Mode

D Utility Programs

EXata provides the following utility programs:

- Radio Range Utility Program
- Run Test Case Script
- Urban Grid Script
- Shapefile-to-XML Utility Program

D.1 Radio Range Utility Program

The radio range program in the bin subdirectory is a command-line utility that returns the distance that the radio transmission can reach using the set of parameters in the specified configuration file. It assumes that there is no interference from other transmitters. The values given by the radio range utility are approximate, since the random components such as fading and shadowing are not take into consideration when estimating the range.

The radio range program is located in EXATA_HOME/bin. In Windows, the executable is called radio_range.exe and in UNIX, the executable is called radio_range.

To use the radio range program, type the following command:

`radio_range <input-file>`

or

`radio_range <input-file> <tx-node> <rx-Node>`

where

`<input-file>`	Name of the scenario configuration file
`<tx-node>`	Node ID of the transmitting node
`<rx-node>`	Node ID of the receiving node

If the transmitting and receiving nodes are not specified, then the transmitting node is assumed to be node 2 and the receiving node is assumed to be node 1.

The radio range utility considers all combinations of wireless interfaces at the transmitting and receiving nodes and all available wireless channels. However, the range is estimated only if the transmitter and receiver operate on the same wireless channel and use the same PHY model on their interfaces to the common channel.

The radio range utility provides an estimate of the range for the down link (from the transmitter to the receiver). To get an estimate in the reverse direction, run the utility again with the transmitting and receiving node IDs swapped.

Using radio range utility for a specified transmitter node will give the coverage plan capability for several simple propagation models (free-space, two-ray and COST 231-Hata pathloss models).

Example Usage and Output

Input:

```
..\bin\radio_range default.config
```

Output Generated:

```
Partition 0, Node 1 (224.48, 1004.00, 0.00).
Partition 0, Node 2 (271.17, 1109.12, 0.00).
Partition 0, Node 3 (354.24, 967.43, 0.00).
Partition 0, Node 7 (1097.31, 1013.80, 0.00).
Partition 0, Node 8 (943.00, 1111.56, 0.00).
Partition 0, Node 9 (972.31, 989.40, 0.00).
Partition 0, Node 4 (500.69, 1011.40, 0.00).
Partition 0, Node 5 (647.39, 1006.51, 0.00).
Partition 0, Node 6 (796.66, 1006.51, 0.00).

 Channel Index = 0, Frequency = 2400.000000 MHz,
 Tx Node = 2, Interface Index = 0,
 Rx Node = 1, Interface Index = 0,

 Radio range:  421.404m, for 802.16 data rate 14.28 Mbps
 Radio range:  298.331m, for 802.16 data rate 21.43 Mbps
 Radio range:  223.717m, for 802.16 data rate 28.57 Mbps
 Radio range:  149.519m, for 802.16 data rate 42.85 Mbps
 Radio range:  118.768m, for 802.16 data rate 42.85 Mbps
 Radio range:   94.340m, for 802.16 data rate 57.14 Mbps
 Radio range:   74.937m, for 802.16 data rate 64.28 Mbps

 Channel Index = 1, Frequency = 3400.000000 MHz,
 Tx Node = 2, Interface Index = 0,
 Rx Node = 1, Interface Index = 0,

 Radio range:  297.462m, for 802.16 data rate 14.28 Mbps
 Radio range:  210.586m, for 802.16 data rate 21.43 Mbps
 Radio range:  157.917m, for 802.16 data rate 28.57 Mbps
 Radio range:  105.543m, for 802.16 data rate 42.85 Mbps
 Radio range:   83.836m, for 802.16 data rate 42.85 Mbps
 Radio range:   66.593m, for 802.16 data rate 57.14 Mbps
 Radio range:   52.896m, for 802.16 data rate 64.28 Mbps
```

D.2 Run Test Case Script

The Run Test Case (runtestcase) script in the EXATA_HOME/bin directory is used to run any of the non-interactive scenarios provided with EXata. It runs each scenario and compares the resulting output statistics file with the expected.stat file provided with the installation. The script and the collection of scenarios are used to check both for proper installation of the software and for unintended side effects of code changes. When you change the code of one protocol, or introduce a new protocol, it should not affect the expected outputs of other existing models. Unexpected changes are often indicative of an incorrect implementation.

The script is a text file and includes, as comments, detailed instructions for its use. To use the Run Test Case script on Linuxsystems, type the following command:

```
$EXATA_HOME/bin/runtestcase -r [-np <N>]
```

where

<N>	Number of processors to run EXata on.

The Run Test Case script can be also be run on Windows by using the same command if Cygwin is installed.

D.3 Urban Grid Script

The Urban Grid script (urban_grid.pl) is a Perl script in the EXATA_HOME/bin directory that can be used to generate sample terrain features data (buildings, streets, parks, metro stations) in Qualnet Terrain Format. The data is generated in cartesian coordinates.

The streets are arranged in a grid. A metro station is placed at each of the four corners of the grid. One grid block must be selected as a park. All other blocks have a randomly placed building. Building heights are randomly selected within the specified range. Rows are numbered starting at zero, increasing in the Y direction. Columns are numbered starting at zero, increasing in the X direction. No traffic lights are generated in this sample.

The output is printed on the screen unless it is directed to a file.

To use the Urban Grid script, type the following command (all parameters must be typed on a single line):

```
urban_grid.pl <num_rows> <num_cols> <grid_width> <street_width>
<park_row> <park_col> <min_height> <max_height>
```

where

<grid_width>	Distance between blocks in meters
<street_width>	Width of streets in meters
<park_row>	Row number for the grid where the park is located
<park_col>	Column number for the grid where the park is located
<min_height>	Minimum building height
<max_height>	Maximum building height

Example Usage:

```
perl urban_grid.pl 5 5 200 12 2 2 50 100 > mapdata001.xml
```

Figure D-1 shows the Architect canvas when the terrain file generated by the above command is loaded.

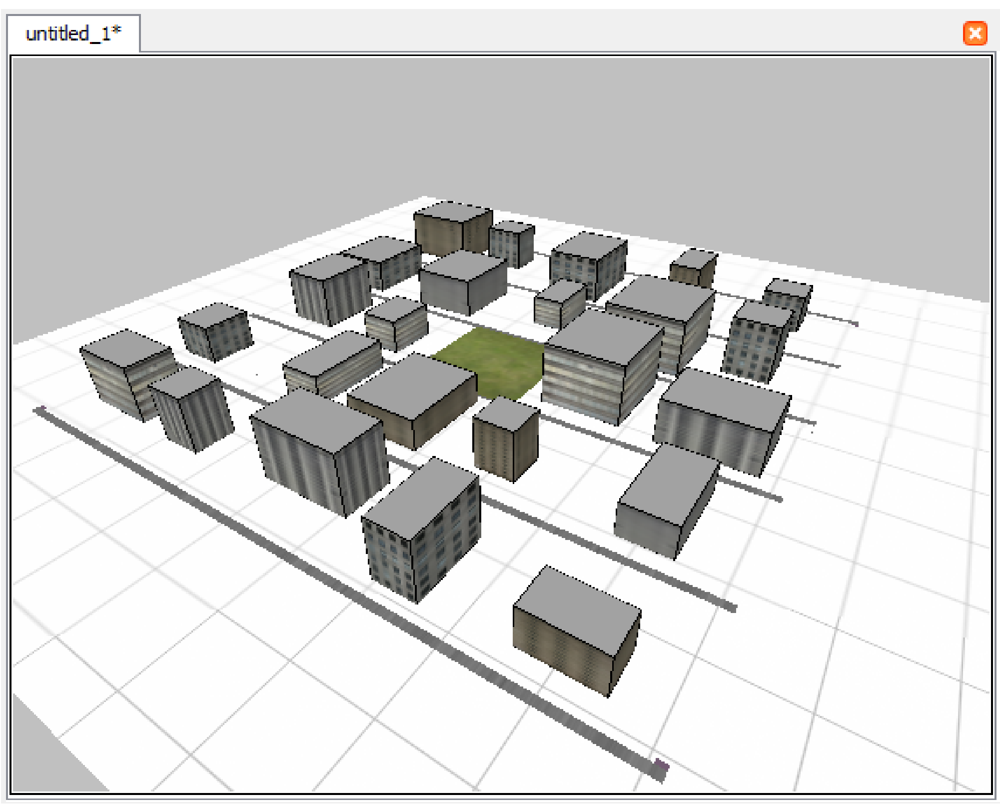


FIGURE D-1. Visualization of Terrain Data Generated by Urban Grid Script

D.4 Shapefile-to-XML Utility Program

The Shapefile-to-XML utility program is a command line program that generates terrain features files in QualNet Terrain Format from ESRI shapefiles.

The Shapefile-to-XML program is located in EXATA_HOME/bin. In Windows, the executable is called shptoxml.exe and in Linux, the executable is called shptoxml.

To use the Shapefile-to-XML program, type the following command (all parameters must be typed on a single line):

```
shptoxml [<options>] <filename>
```

where

<options> Optional parameters.

The optional parameters of the Shapefile-to-XML program are described in Table D-1.

<filename> Name of the input shapefile.

If the input file has an extension, it should be `.shp`.

TABLE D-1. Shapefile-to-XML Program Options

Option	Description
`-h`	Displays the help menu.
`-b <height> <units>`	Specifies the default height of features and the units. `<height>` is a numerical values specifying the height. `<units>` can be `ft` (for feet) or `m` (for meters). The default height of features is `35 ft`.
`-B <field-name>`	Specifies the field name associated with he feature height attribute in the database (.bdf) file. `<field-name>` can be any string. The default field name is `LV`.
`-c <id>`	Specifies the ID of the file to be generated. `<id>` is an integer (≥ `0`). This option is used in conjunction with the `-n` option to create a specific XML file instead of all XML files. The default file ID is `0`.
`-D <foliage-density>`	Specifies the foliage density. `<foliage-density>` is a real number in the range `[0.0, 1.0)`. The default foliage density is `0.15`.

TABLE D-1. Shapefile-to-XML Program Options (Continued)

Option	Description
`-F`	Indicates that the shapefile contains only foliage data. By default, shapefiles contain only building data.
`-m <min-lat> <max-lat> <min-lon> <max-lon>`	Specifies the region filter. XML files are generated only for features within the region specified by this option. `<min-lat>` is the minimum latitude. The default minimum latitude is `-90.0`. `<max-lat>` is the maximum latitude. The default maximum latitude is `90.0`. `<min-lon>` is the minimum longitude. The default minimum longitude is `-180.0`. `<max-lon>` is the maximum longitude. The default maximum latitude is `180.0`.
`-n <count>`	Specifies the maximum number of buildings to output to a file. `<count>` is an integer ($\geq$ `0`). By default, the maximum number of buildings to output to a file is 2^{32} - 1.
`-U <units>`	Specifies the units for dimensions of terrain features. `<units>` can be `ft` (for feet) or `m` (for meters). By default, `<units>` is `ft`.