precache

PreCache NetInjector Administration

Contents

PreCache NetInjector Administration

Welcome to PreCache NetInjector

NetInjector Administration

Welcome to PreCache NetInjector

PreCache NetInjector is the industry's first Internet-scale publish/subscribe communication system. Reversing the traditional 'pull' model of content distribution in favor of a subscription-based system, *NetInjector* reduces network load and bandwidth requirements and facilitates meaningful client interactions.

About this guide

— the goals, objectives, and layout of this guide

Goals and objectives

PreCache NetInjector Administration provides basic information about installing your NetInjector package and about configuring and managing your *NetInjector* network.

This guide focuses on the process of establishing and managing your network topology, including the routers and servers, the channels and their respective subjects and attributes, and users.

While including step-by-step instructions, the guide presents an overview of the interface from the perspective of NetInjector's conceptual model.

Conventions

The following typeface conventions have been observed in these pages:

- `code`
- *fields*
- links
- *methods*
- **system items**
- `values`
- *variables*

For better readability in both PDF and HTML, these conventions may sometimes print as either boldface or italic.

Getting started

— installing the package

You package includes two administrative installations: NetInjector software and the *NetInjector Management Console* software.

In this section, will describe the steps to install these applications. We also document a procedure for creating public keys for OpenSSH.

System requirements

RedHat Linux v8.0 or above

Java Runtime Engine 1.4 or above

OpenSSH v3.4 or above

NetInjector Installation

1. As root user, verify IP configuration.
2. Create group and user entries for **netinj**, e.g.:

   ```
   groupadd -g 500 netinj

   useradd -u 500 -g netinj -d /home/netinj
   netinj
   ```

3. Mount the CD-ROM.

   ```
   mount /mnt/cdrom
   ```

4. Untar the tar file from the CD into the **netinj** home directory:

   ```
   cd ~netinj

   tar xzvf /mnt/cdrom/router_*.tar.gz
   ```

5. If auto start from boot is required, add at the end of `/etc/rc.local`:

   ```
   sudo su - netinj /home/netinj/bin/
   start_router
   ```

   Note: sudo must be installed to auto start from boot.

Management Console installation

The *NetInjector Management Console* may be installed and executed remotely from any Linux workstation with proper JRE installed.

1. Untar the file admin_*.tar.gz from CD.
2. Store jnmsgui.jar and netinj_gui.ini locally.
3. Run jnmsgui.jar to start the *Management Console*.

OpenSSH

The *Management Console* protects access to remote routers by using SSH. Install an **ssh client** on the *Management Console's* host and **ssh server** on all remote routers.

To avoid entering a password or passphrase when accessing a remote router, you may want to select the public-key scheme for SSH access.

To create a public-key under OpenSSH:

1. On all remote routers, create **.ssh** directory with attribute `700`:

   ```
   mkdir /home/netinj/.ssh; chmod 700 /home/
   netinj/.ssh
   ```

2. For the *Management Console's* host, generate the two id files `id_rsa` and `id_rsa.pub` under `/home/netinj/.ssh`:

   ```
   ssh-keygen -t rsa
   ```

   Leave the passphrase empty to avoid the need to enter a passphrase.

3. For all remote routers, copy the content of `id_rsa.pub` into the file:

   ```
   /home/netinj/.ssh/authorized_keys
   ```

NetInjector Overview

— a brief consideration of the basic concepts

Push and pull

The traditional publisher posts content to a server and waits for a member of the target audience to find the content and request it.

This 'pull' model is resource intensive and depends on a wide range of complex transactions for success. Not least of these is the requirement that the interested parties find one another in the first place.

Publishing content in a distributed 'push' environment reverses this model: the publisher posts content to a pool of receptive members. There is no need to create redundant content, to manage server farms holding enormous volumes of content, or to facilitate the hapless surfer's search efforts: members of the group receive data automatically, according to the terms of their relationship with the publisher.

> *In creating a stock quote service, the traditional internet news site must store all data related to every stock and post it to everyone who requests the information.*
>
> *The* NetInjector *news provider can post stock quotes targeted to the specific interests of existing subscribers, rapidly responding to their known preferences.*

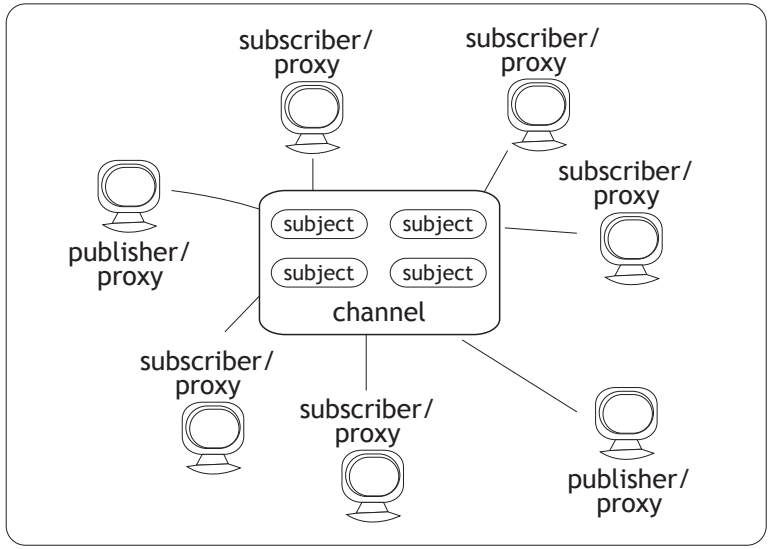
Channels

A key construct of the *NetInjector* solution is the channel. Each channel is a namespace of subjects and attributes. The channel exists as a communications port, facilitating the transfer of data from publisher to subscriber. The channel also assures security and provides stable quality of service.

Each area of interest or content type is a *subject*. Subjects provide a category for content within a channel. Publishers create content for a subject; subscribers receive the subject content.

Channels are further defined by a set of *attributes*, which enhance content distribution. These attributes provide additional flexibility for sorting and routing information and, therefore, packets.

> *The stock service for a news service would be organized as a subject within a NetInjector news channel. The channel's subjects could include, in addition to stocks, other areas of interest such as sports and politics.*
>
> *Within the channel, predefined attributes would allow the publisher and subscriber to filter information by ticker symbol, price, dates, or other data contained within the media provider's information feed.*



Agents

Each client accesses the channel via an agent or *proxy*; the agent is implemented via a proxy. The proxy connection to the router provides an interface for the client to publish and receive notifications for the channel and its subjects.

Proxies receive notifications for the duration of each session. However, the event agent supports methods to pause, suspend, and restart open proxies, avoiding the need to close connections.

Notifications

When a publisher posts new content for a subject within a channel, a notification is prepared for the subscriber base. This notification conforms to the parameters established by the initial configuration of the channel and subject, as well as to the programming logic of the event agent or application.

As the notification passes through the system, the NetInjector router determines the ultimate destination of the notification. At each network node, the notification advances toward the greatest number of subscribers.

Thus, content which targets several subscribers only passes through the network once for a given network node or router. By such *content-based routing*, *NetInjector* routes packets based on content rather than by explicit request, greatly reducing loads.

> *Each notification to a subscriber would contain information about the stock which met the subscriber's particular criteria, consistent with the application logic.*
>
> *Thus, if the application supports filtering by share volume, then a subscriber could ask to be notified when a stock's daily volume reached a certain level of trading activity. So, too, the agent could be programmed to deliver notifications based on price or any other criteria.*

Filtering

Subscribers request content based on their interests. However, the entire data stream for a subject could be vast, exceeding both the requirements of the publisher to serve a target audience and the subscriber's need for content.

By writing filters to better tailor content to a subscriber's interests, the application developer can take best advantage of *NetInjector*'s ability to reduce server load and accelerate the distribution of meaningful content.

These filters may sort data by the predefined criteria established when the channel was created, or they may be created as a concatenation of filters. The application developer may also create filters for specific instances.

> *In delivering notifications, the stock service could allow a subscriber to receive notifications for a given stock once its price exceeded a certain level and trading activity crossed a predetermined barrier.*
>
> *Furthermore, the service could allow the subscriber to create unique filters based on any data within the stream, facilitating notifications by such constructs as moving averages or price/earnings ratios.*

Infrastructure

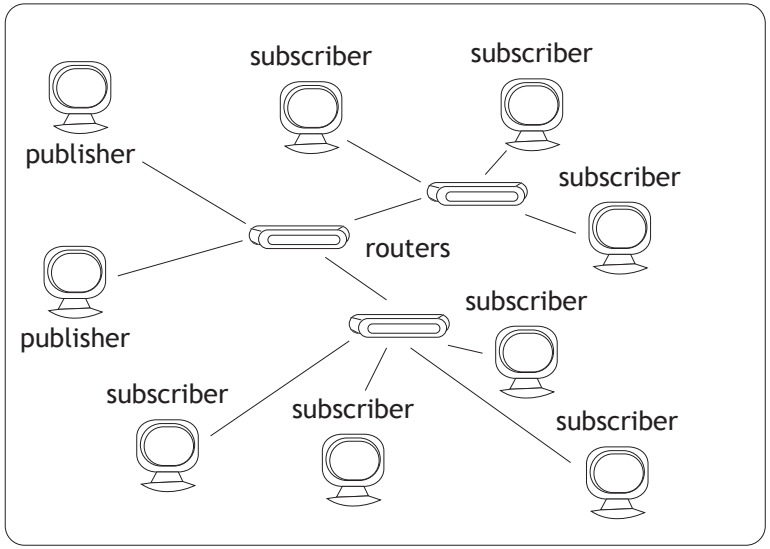
NetInjector supports a two-stage router environment. One stage routes publisher content; the second delivers subscriber content. Together, this simple topology facilitates efficient data flow between the publisher and the subscriber. Because *NetInjector* maintains a logical distinction between stages, no data flows within a stage.

Data flow

In a traditional network, each request from a subscriber results in a unique transaction with the server.

In a *NetInjector* network, the publisher's notification only propagates to those subscribers interested in the content. If all of the subscribers in a neighborhood request the same data, *NetInjector* generates only one notification.

Each data flow is finely tuned to the subjects and attributes of the channel, ensuring that content is distributed to interested subscribers.



The publisher creates content for the channel, creating a data stream for all subscribers. The routers distribute the content based on subscription information. Each notification propagates towards the neighborhood of interested subscribers.

NetInjector Management Overview

— a high-level view of NetInjector management

The principal task of the *PreCache NetInjector Management Console* is to provide you with the means of creating and maintaining your Network *topology*

A *NetInjector* network topology is a working roadmap of channels, subjects, and attributes, user groups, and the hardware serving the network. By carefully mapping your software initiatives to the physical hardware, you can establish a topology that will maximize your *NetInjector* platform.

Your network topology's physical requirements consist of routers and servers.

router: distributes content within and across the network; routers serve either publishers or subscribers

server: Aquila Information Server (AIS), which manages network communications, mapping users, routers, and channel groups; or Channel Manager (CM), which provides channel services, mapping subjects and attributes for groups of channels

Each topology must have an AIS, a router and a channel manager.

Routers may have links among themselves as well as to the AIS. In establishing your topology, consider carefully the hardware you wish to configure and the services you plan to develop and maintain:

- What are the physical locations of the routers and servers?
- What areas will the routers serve?
- How much traffic will the area generate?
- How many subscribers are in each area?
- What kind of channels do you plan to create for your network?
- Are the routers one or two stage routers?
- What are the IP addresses for your hardware?
- Which machine will be designated as the Aquila Information Server (AIS)?

NetInjector Administration

Overview

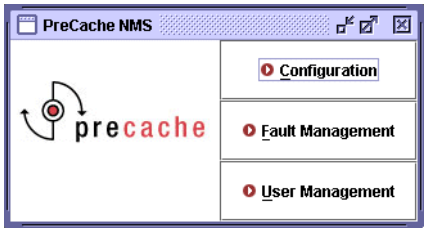
— a quick tour of the management console

The *Management Console* includes three interfaces:

Configuration: Create and define your network topology.

Fault Management: Monitor routers and AIS.

User Management: Manage customers and operators.



Each of these interfaces is closely linked with your NetInjector publisher and subscriber applications.

- The configuration of the network contributes to the process of creating and distributing notifications.
- The fault management of the network affects network usage and reliability.
- Customer and operator account management directly affects customer satisfaction and operational profitability, both crucial to the successful deployment of applications.

In this chapter, we will review the basics of each interface. As the NetInjector administrator must first create and define a network, we focus our attention especially on the Configuration interface.

Interface basics

When you first run the *PreCache NetInjector Management Console*, the interface will present an empty workspace.

The interface's features are activated by both mouse clicks and buttons:

- When working within the topology views, the *Management Console* presents command options on clicks.
- In all other modes, the interface requires input and executes commands associated with your input when you click an appropriate button.

The Management Console operates directly on your network. All operations are instantly updated whenever you save or commit your work.

Note: The **Save** command stores the topology view locally. Whether you save your view or not, changes to the topology are implemented as you create, remove, set, commit, or otherwise perform tasks.

Configuration

— creating a network topology

The configuration of a *NetInjector* network requires that routers and servers maintain a data structure for channels. This data structure ultimately enables the application developer to produce lightweight, content-based applications for publishers and subscribers.

The configuration of your network requires you to define parameters for four operational levels:

Network Topology: *the physical network*

Channel Service: *the information base for channels*

Routers: *assigning the physical hardware*

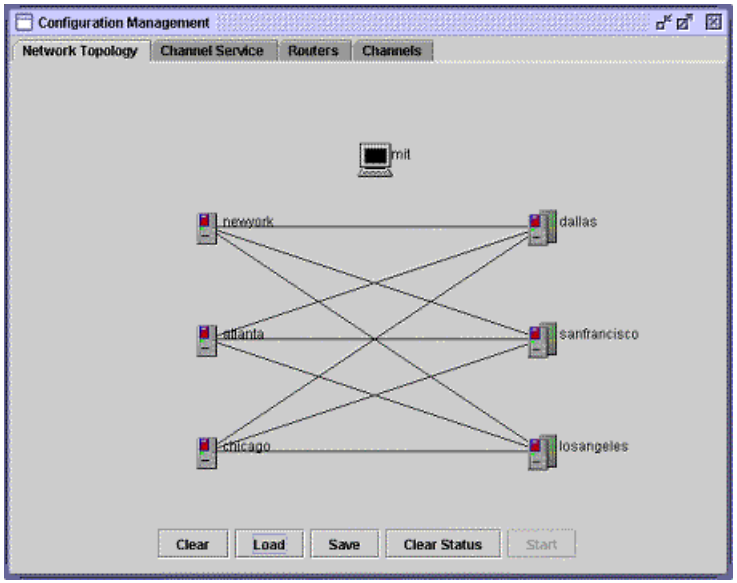
Channels: *the namespace for subjects and attributes*

Network Topology

— the physical network

NetInjector routers move notifications along from point to point.

Your NetInjector environment establishes links among routers, and manages these intelligently, so that notifications move efficiently.



The Network Topology interface provides a view of your physical network; you can add routers and servers, as well as establish links among them.

Create a router or server

Your topology begins with the placement of routers and servers. As you develop your network, you can add routers to support new or expanding services.

1. Open the **Configuration** window.
2. Click in the workspace to create a router or server.
3. At the prompt, create either a standard router, a channel server, or an AIS server.

   Note: The default setting for **server** is *channel server*. Select only one server to be an AIS.

4. Enter the following information at the prompts

   router name: the physical name of the machine

   IP address: the machine's static IP address

   stage: select stage one for publishing, stage two for subscribing

Modify a router

Your configuration must conform to the physical environment represented by the topological view. If you change the hardware's name or IP address, or entered the information inaccurately, you can modify the node in the topology, or remove it entirely.

1. Left-click a router to access its properties.
2. Click **Properties** to change the router's name or IP address.
3. To delete a router and its links, click **Remove**.

Create the network

Once you have defined the routers and servers for your network, you can create links among them. Create your links between stage one and stage two servers.

For example, in the foregoing network topology, publisher routers appear on the left; subscriber routers are on the right. The routers are linked such that publishers served by *newyork*, *chicago*, and *atlanta* can publish notifications to subscribers served by *dallas*, *sanfrancisco*, and *losangeles*.

Note: To maximize performance and ensure stability, do not create links between routers serving the same stage.

1. Right-click a router to highlight it.
2. Right-click another router to link to the highlighted one.

Manage the network

You can also use the **Network Topology** view of your network for fault management.

1. Left-click a router to check router properties and node status.
2. Select **Properties** from the prompt box to check the name and IP address of a router.
3. Select **Node Status** from the prompt box to check the availability of the router.

For daily router maintenance, use the *Fault Management* interface to check router status and to stop processes.

Channel Service

— the information base for channels

The **Channel Services** interface allows you to manage the information base of the pipe for data flows within channels.

Channels exist as tunnels across the network. Each channel is essentially a tunnel: publishers and subscribers tap into the tunnel whenever they want to tap into the information flow within the channel.

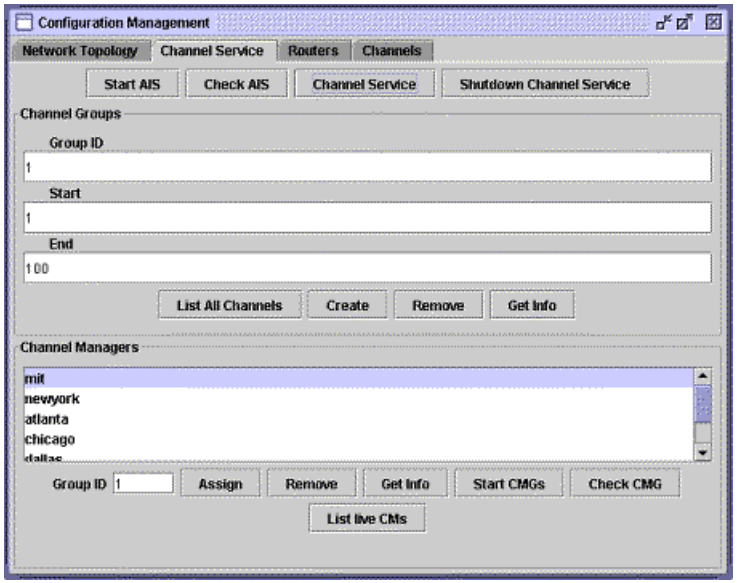
Channel Service include Aquila Information Server (AIS) and Channel Manager (CM).

Aquila Information Server (AIS): manages network communications, mapping users, routers, and channel groups

Channel Manager (CM): provides channel services, mapping subjects and attributes for groups of channels

The process of configuring Channel Service consists of:

1. Starting AIS and Channel Service.
2. Defining Channel Groups.
3. Designating a Channel Manager.



The Channel Services interface facilitates the tasks of creating and maintaining channel groups for your network.

AIS

AIS matches routers and users by zip code, so that each channel's notifications flow efficiently through the network. AIS maintains the database of the network topology, including router nodes, zip code information and channel managers.

Note: The task of assigning zip codes to routers, and routers to zip codes, is performed within the Routers interface.

When you start AIS, *NetInjector* initializes the network according to your topology. AIS must be running when you start routers or channel managers.

Start AIS: starts (or restarts) AIS

Check AIS: checks whether AIS is running or not

Channel Service: exposes the channel group for the channel manager for editing purposes

Shutdown Channel Service: stops AIS and Channel Managers

Channel Groups

A NetInjector topology is comprised of a set of channels. Each channel is part of the group of channels within a network.

Each channel must belong to a group. You may create more than one group for a topology. Channels may not be assigned to multiple groups.

Create a channel group

Creating a channel group requires that you know how many channels you plan to create, as well as the existence of other groups.

group ID: an administrator-assigned identifier for the group

start/end: the numbers of the first and last channels in the group; channel numbers do not need to run consecutively. You may create a group for more channels than you actually build.

Before building your group, load the topology and start AIS.

1. Click **Channel Service**.
2. Enter a *Group ID* for the set of channels you are creating.
3. Enter *Start* and *End* values for the group.

   Note: You cannot modify these values once you create the group.

4. Click **Create** to add the group of channels to your topology.

Delete a group

You can remove a group from the topology.

Remove: removes channel services and all designated channels for the group from the topology

1. Click **Channel Service**.
2. Enter a *Group ID* for group you wish to remove.
3. Click **Remove** to delete the group and all its channels from the topology.

Caution: Removing a group deletes all of the channels within the group, including all of their subjects and attributes. As these changes are updated automatically, you should exercise caution when removing a group from a live environment.

Query channel group properties

You can view channel group properties and information without affecting the network.

List all Channels: a list of the channel numbers in the group; a useful reference when creating new groups

Get Info: lists the channel group's start and end values

Channel Manager

Each group of channels must have its own Channel Manager; a channel manager can only serve one group. The Channel Manager handles internal communications regarding subjects and attributes between the network and the channel.

Create a Channel Manager

After creating a group, assign the role of Channel Manager to one of your routers or servers.

Note: You can start the channel management service only after starting AIS.

1. Select a node from the list.
2. Click **Assign** to create a Channel Manager for the group.
3. Click **Start CMGs** to start the Channel Manager for the group.

   Note: Start CMGs restarts all Channel Managers for your topology, not merely the Channel Manager for the specific group listed in the interface.

Modify a Channel Manager

You can also check the status of the Channel Managers for your topology, reassign the task, or remove the assignment.

Remove: deletes the Channel Manager assignment from the selected host

Get Info: identifies the Channel Manager for the group

Check CMG: status of the Channel Manager

List Live CMs: indicates which Channel Managers are actively running in your network

To reassign Channel Manager to a different host:

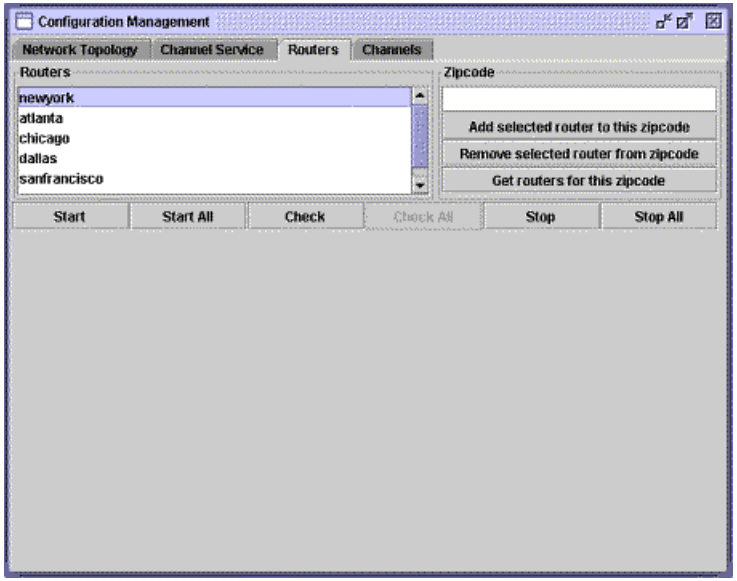
1. Select the node running Channel Manager from the list.
2. Click **Remove**.
3. Select the node to which you plan to reassign the service.
4. Click **Assign** to create a new Channel Manager for the group.
5. Click **Start CMGs** to start the Channel Manager for the group.

Routers

— assigning the physical hardware

Each router serves a geographical area or set of areas. Conversely, each region may be served by more than one router or server.

AIS handles the task of mapping data flows for a geographical region's subscribers to particular routers, and for the flow of data from routers to particular regions. The routers themselves are not aware of the zip code assignment.



You can manage the association between routers and zip codes, as well as start and stop routers.

Associate a router with a zip code

Zip code assignments may be made after starting AIS.

1. Select a node from the list.
2. Enter a zip code.
3. Click **Add selected router to this zip code** to associate the router and zip code, and to update AIS's assignment list.

Manage routers

You can start and stop routers, as well as manage zip code associations.

Note: You cannot start a router until it has been entered as an operator in the User Management interface.

Remove selected router from zip code: removes the selected router from the Channel Manager's assignment list.

Get routers for this zip code: lists the routers associated with the zip code

Start: restarts the selected router to reflect the new zip code assignments

Start All: restarts all routers in the loaded topology to reflect the new zip code assignments

Stop: terminates service on a specific router

Stop all: terminates service for all routers in the loaded topology

AIS must be running to start, stop, or shutdown routers.

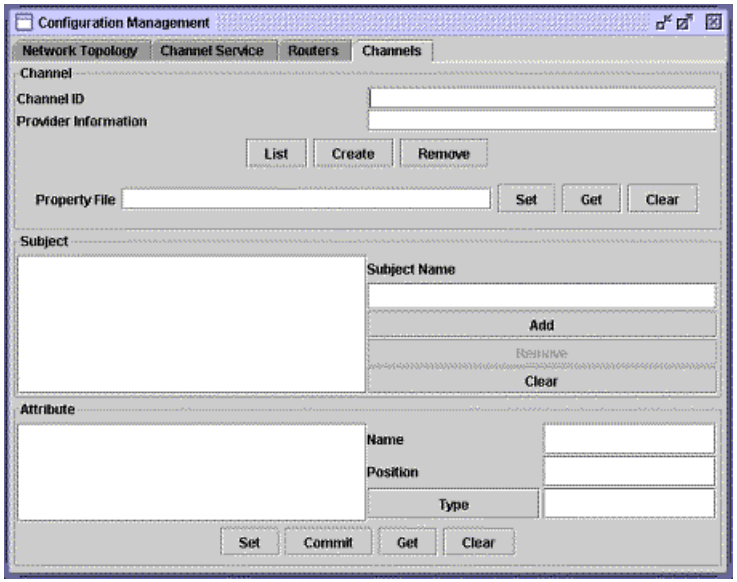
For daily router maintenance, use the Fault Management interface to check router status and to stop processes.

Channels

— the namespace for subjects and attributes

Channels are a namespace for subjects; content within a channel is organized by subject. Each channel has characteristic attributes.

Proxy connections to *NetInjector* access channels. The event agent, after opening a proxy connection to a channel, publishes and subscribes to the data stream within the channel.



The Channel interface of the PreCache NetInjector Administration exposes the data structure for each channel. This data structure should be designed with the event agent application developer.

Channel Information

The configuration of the channel and its subjects must conform to the conventions of the application design for notifications. So, too, the application must conform to the data structure for the channel, as created within the **Channels** interface.

Create a channel

Before creating a channel, you will need to know the following:

Channel ID: a numeric token for the channel, assigned by the administrator

Provider Information: comments about the data provider

1. Enter a unique **ID** for the Channel.
2. Add **Provider Information**.
3. Click **Create** to add the Channel.

Subjects

Each category of information within a channel is a subject. Subjects may have subcategories. A *Hobby* channel could have many subjects:

- reading
- arts and crafts
- gardening
- woodworking
- music
- collecting

A particular subject would be represented within the channel as a set of subjects and subcategories:

- woodworking
  - woodworking.projects
    - woodworking.projects.interior
    - woodworking.projects.exterior
  - woodworking.wood
    - woodworking.wood.hardwood
    - woodworking.wood.softwood

The richness of each channel depends on the proper design of the subjects within the channel. This aspect of your topology should be carefully designed before committing the changes to the network.

Create a subject

Following your naming conventions for subjects, you can create subjects for your channel.

Subject: a category of content

Subject Name: the name of the subject, used to identify the subject and its subcategories

To create a subject:

1. Enter a *Subject* name.
2. Click **Add** to include the subject in the channel.
3. Add subcategories, as needed.

   subject.subcategory.subsubcategory

4. The subject will be available when you **Commit** your work.

Attributes

Each channel has certain common attributes. The attributes defined at this stage facilitate the filtering of data within the network. As such, these *predefined attributes* are crucial to the success of your *NetInjector* deployment.

These attributes are essential to filtering content, whether you are returning simple text messages or developing rich content:

- Attribute filters enable you to create richly detailed content without building a finely-tuned subject hierarchy.
- Attribute filters facilitate rapid publication of content to narrowly-defined audiences.

Predefined attributes may be accessed by the application developer by name or position. Each of these parameters, as well as data type, must be entered when the channel is committed to the database.

The administrator and application designer should work closely together to create these predefined attributes for channels.

Note: If the application developer would like to create attributes unique to a particular subject within the channel, the developer should create *discretionary attributes*.

Here are a few predefined attributes for our *Hobby* channel:

- material
- experience level
- craft
- message

The data type for each of these attributes may vary, depending on how the developer needs to call them in the application itself.

Create an attribute

Each channel must have attributes. All of the subjects and subcategories for a channel share the same attributes.

Each attribute must have these three values:

Name: the name of the attribute

Position: the array position of the attribute

Type: the data type

To create attributes for a channel:

1. Enter an attribute *Name*.
2. Enter an attribute *Position*.

Note: Number attributes consecutively, beginning with 1.

3. Select the attribute's data *Type* from the select list.

   Note: You may select any of the first seven types in the list.

4. Click **Set** to assign the attribute to the channel.

5. Click **Commit** to update the channel.

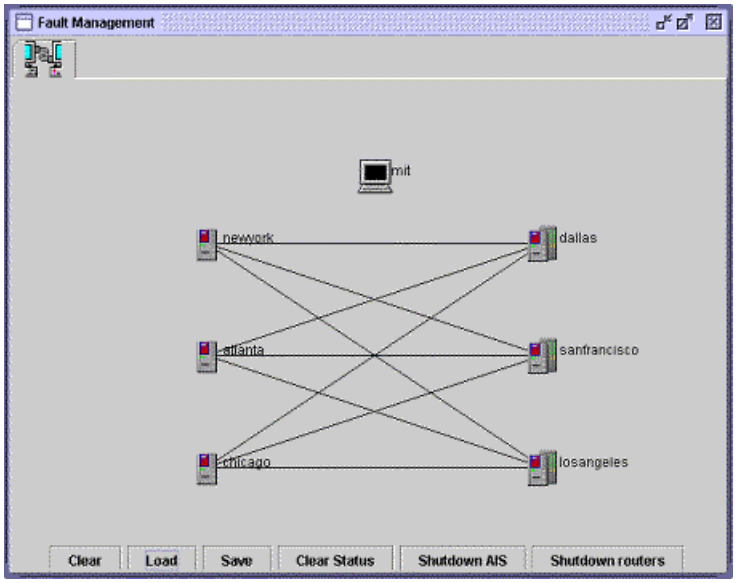
Fault Management

— regular maintenance of your network

The **Fault Management** interface provides a simple means of checking router status and of starting and stopping routers and the AIS server.

Use the Fault Management interface for quickly scanning your network's performance, and for interceding when necessary.

Routers which are not operating properly are highlighted in red.



The Fault Management interface replicates many of the features of the Configuration interface's Network Topology tab, while focusing attention on router status and behavior.

Clear: remove the topology from the workspace

Load: open the existing topology

Save: save any changes made during the session

Clear Status: clears the screen view of the topology

Shutdown AIS: terminate the AIS process

Shutdown routers: terminate all router processes

You can also check the runtime status of any individual router by clicking on it.

User Management

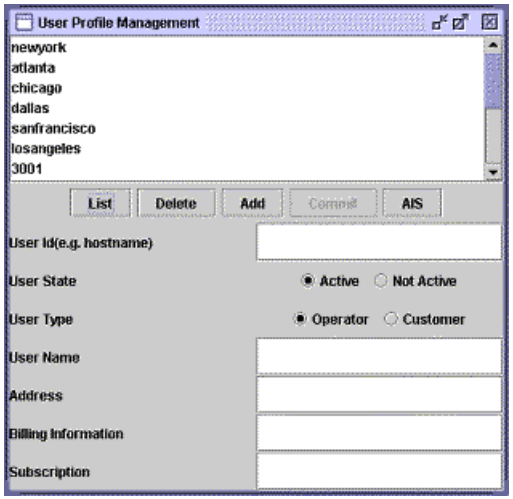
— managing user accounts

User accounts reflect the NetInjector conceptual model: users may be customers or operators.

customers: individual or group subscribers or publishers

operators: routers

Note: Routers must be entered in the user database before they can be started. Be sure to enter the hostname.



The User Management interface facilitates basic record keeping for all users.

List: calls up the list of customers and operators

Delete: remove a customer or operator from the database

Add: add a customer or operator to the database

Commit: save changes to an account

AIS: check the status of the AIS for the user base