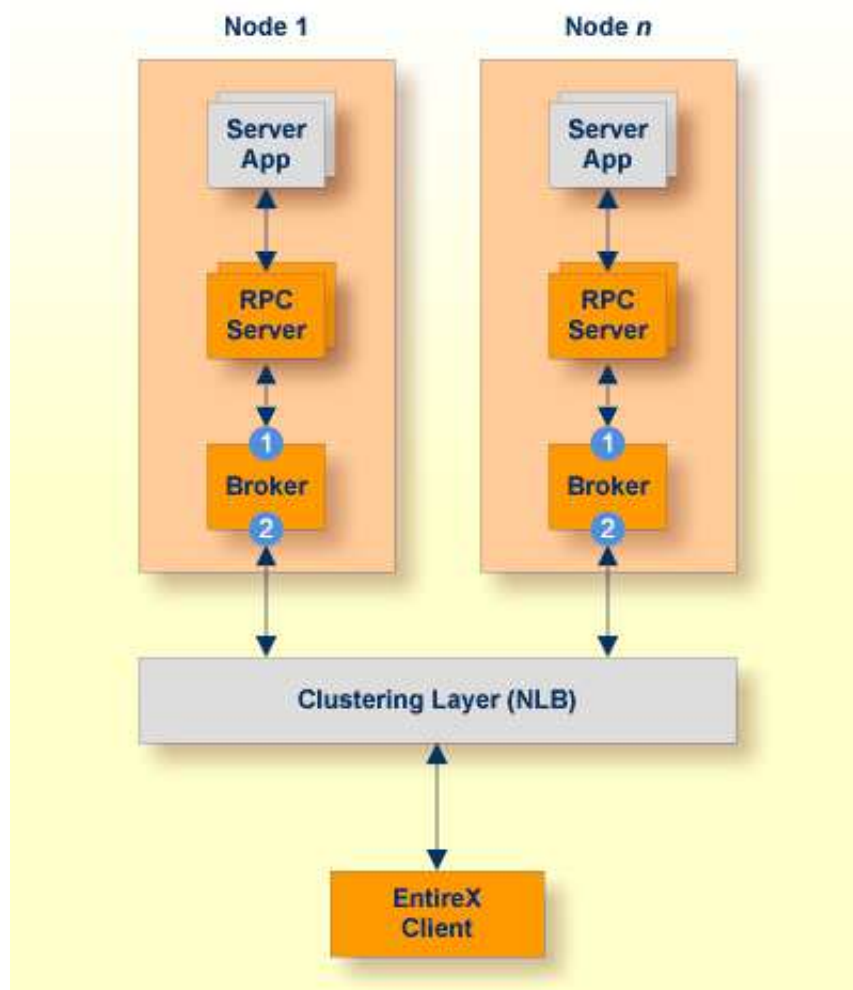


# Setting up Your EntireX Environment for High Availability with Other Clustering Technologies

This chapter covers the following topics:

- Setting Broker Attributes
  - Broker Cluster Considerations
  - Configuring Redundant RPC Servers
  - Managing Brokers and RPC Servers
- 

## Setting Broker Attributes



Segmenting dynamic workload from static server and management topology is critically important. Using broker TCP/IP-specific attributes, define two separate connection points:

- One for RPC server-to-broker and admin connections. **1**
- The second for client workload connections. **2**

See *TCP/IP-specific Attributes* under *Broker Attributes*. Sample attribute file settings:

|          | Sample Attribute File Settings          | Note  |
|----------|---|---|
| <b>1</b> | PORT=1972                               | In this example, the HOST is not defined, so the default setting will be used (localhost).                          |
| <b>2</b> | HOST=10.20.74.103 (or DNS)<br>PORT=1811 | In this example, the HOST stack is the virtual IP address. The PORT will be shared by other brokers in the cluster. |

## Broker Cluster Considerations

We recommend the following:

- Share configurations - you will want to consolidate as many configuration parameters as possible in the attribute setting. Keep separate yet similar attribute files.
- Isolate workload listeners from management listeners.
- Monitor Brokers through SMH.

## Configuring Redundant RPC Servers

In addition to broker redundancy, you also need to configure your RPC servers for redundant operations. We recommend the following best practices when setting up your RPC servers:

- **General Hints**
  - Make sure your definitions for CLASS/SERVER/SERVICE are identical across the clustered brokers. Using identical service names will allow the broker to round-robin messages to each of the connected RPC server instances.
  - For troubleshooting purposes, and if your site allows this, you can optionally use a different user ID for each RPC server.
  - RPC servers are typically monitored using SMH as services of a broker. Optionally, for example for troubleshooting purposes, the RPC servers can be configured with a unique TCP port number for SMH.

**Note:**

SMH port is not supported by the Natural RPC server.

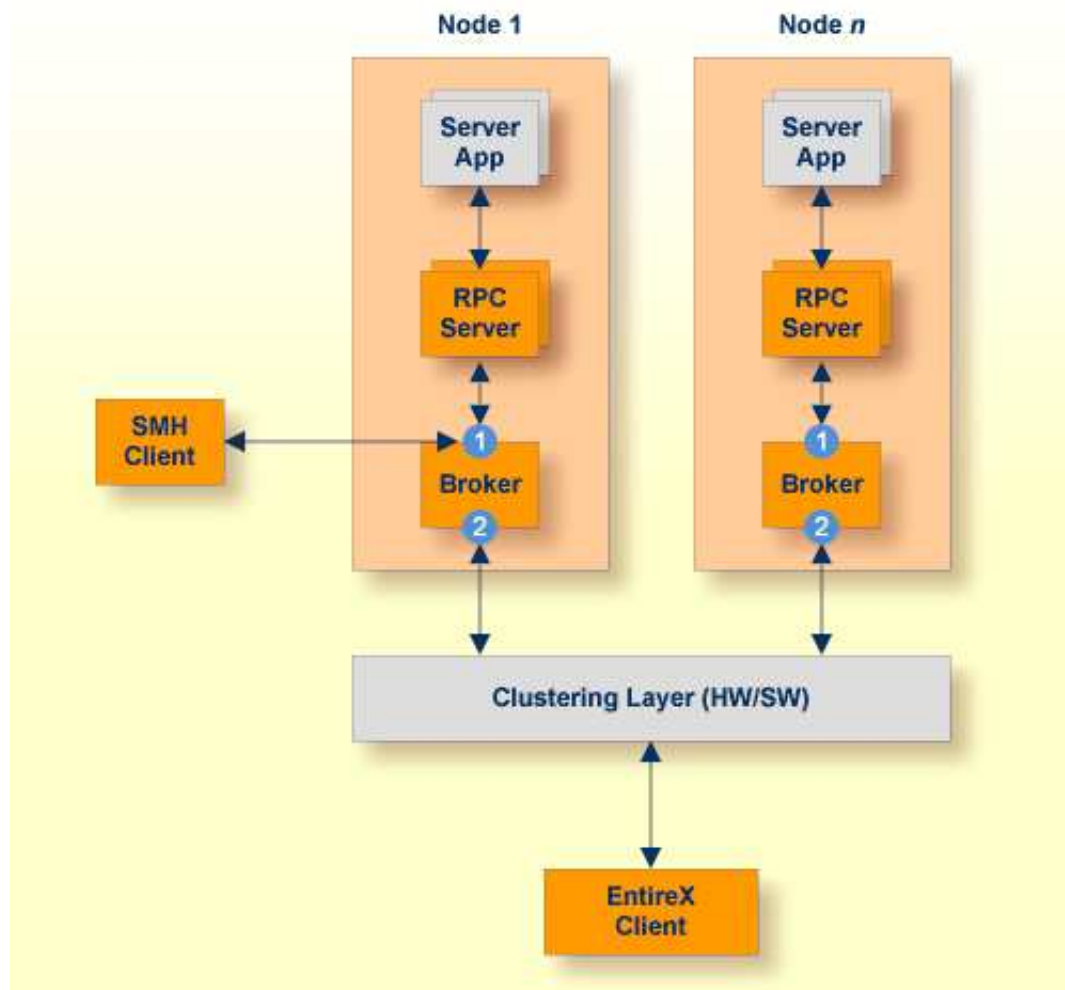
- Establish the broker connection using the static `Broker name:port` definition.
- **Natural RPC Server**
  - Maintain separate parameter files for each Natural RPC Server instance.

## Managing Brokers and RPC Servers

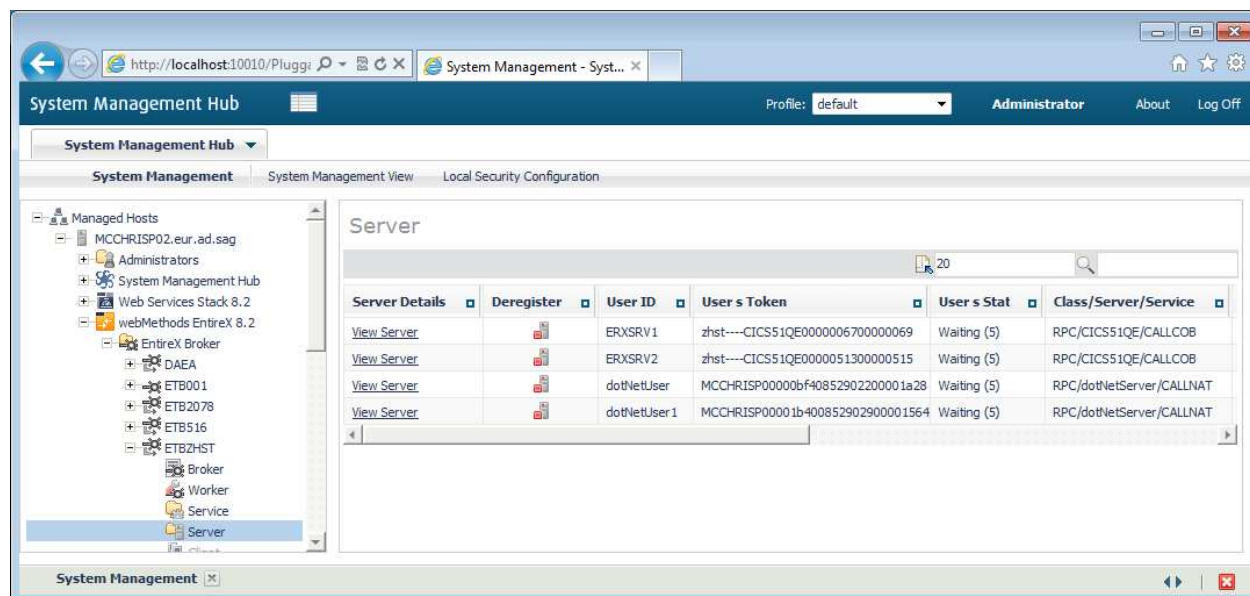
- Using System Management Hub
- Lifecycle Management of Brokers
- Lifecycle Management of RPC Servers

### Using System Management Hub

Use the System Management Hub to monitor the status of the broker and RPC server instances using their respective `address:port` connections. Set up each connection with logical instance names.



The following screen shows two pairs of redundant RPC servers registered to the same broker from the Server view:



- Each broker requires a static TCP port for RPC server and management communications.
- This must be maintained separately from the broker's virtual IP address configuration.

## Lifecycle Management of Brokers

An important aspect of high availability is during planned maintenance events such as lifecycle management, applying software fixes, or modifying the number of runtime instances in the cluster. Using a virtual IP networking approach for broker clustering allows high availability to the overall working system while applying these tasks.

Broker administrators, notably on UNIX and Windows systems, have the need to start, ping (for Broker alive check) and stop Broker as well as RPC servers from a system command-line, prompt or from within batch or shell scripts. To control and manage the life cycle of brokers, the following commands are available:

- `etbsrv BROKER START <broker-id>`
- `etbsrv BROKER PING <broker-id>`
- `etbsrv BROKER STOP <broker-id>`
- `etbsrv BROKER RESTART <broker-id>`

If only one user is to be permitted to execute commands, enter the command

```
etbsrv SECURITY ENABLE TRUSTED-USER=YES
```

The trusted user can then execute commands without any additional authentication. There can only be one trusted user. To change the trusted user, modify the property `install.user` in file `entirex.config`.

See *Administration Service Commands* under UNIX | Windows for more information.

## Lifecycle Management of RPC Servers

### ➤ To start an RPC server

- See *Starting the RPC Server* for UNIX | Windows | Java | .NET | XML/SOAP | Micro Focus | IMS Connect | CICS ECI | WebSphere MQ.

### ➤ To ping an RPC server

- Use the following Administration Service command:

```
etbsrv BROKER PINGRPC <broker-id> <class/server/service>
```

Return code 0 means the broker is running; any other value means the broker has stopped.

### ➤ To stop an RPC server

- See *Stopping the RPC Server* for UNIX | Windows | Java | .NET | XML/SOAP | Micro Focus | IMS Connect | CICS ECI | WebSphere MQ.

You can also use the command-line utility `etbcmd`. Example:

```
etbcmd -b <broker-id> -d SERVICE -o IMMED -m <class/server/service>
```