

IP Server Requirements Specification

1. Scope

1.1 Overview

The IPServer component creates a TCP/IP listener service on a specific port. The service will handle any type of request matching the interface defined by the server. The server will analyze the request and will determine which handler to instantiate to process the request. Once the handler has completed the request, a response will be delivered back to the IPServer which will deliver the response back to the appropriate socket. Handlers implement an interface defined by this component and are added to the component through configuration.

1.2 Logic Requirements

1.2.1 Max Connections

The maximum number of connections the server supports will be configurable. At a minimum the server should expect two thousand (2000) simultaneous connections. Please take into consideration this number will depend on the OS itself, # of file descriptors and other factors. The maximum connections will be set for the entire server.

1.2.2 Connections

Connections will be persistent.

1.2.3 Keep Alive

Optionally the server will support Keep Alive packets for long running transactions. Designers may use the heartbeat component to implement this functionality.

1.2.4 Connection made

Once a connection to the server has been made, an event will occur to hand the request to a handler. The IPServer will define an interface for all handlers. Active handlers will be configurable.

1.2.5 Handler identifier

Every request processed by the server must contain a handler identifier. This will provide the server a way to instantiate the correct handler.

1.2.6 Connection identifier

Every request processed by the server must contain a connection identifier. As soon as the connection is made the server will provide a unique identifier for the specific connection. This connection identifier will also be delivered to the handler.

1.2.7 Max Connections Per Handler

In addition to configuring available handlers, the maximum number of connections per handler will be configurable.

1.2.8 Send response

Once a handler has performed the appropriate logic, it will notify the IPServer with a response payload and the connection identifier. The IPServer will be able to send the information back to the correct connection based on the connection identifier.

1.2.9 JDK 1.4

The JDK 1.3 requirement has been removed. Once a handler has performed the appropriate logic, it will notify the IPServer with a response. This server will not have one thread per client, it will have a number of event handlers to be in pool of threads that is configurable. This means the use of Non-Blocking I/O.

1.3 Required Algorithms

None required.

1.4 Example of the Software Usage

The IP Server will be used to manage such things as chat request inside of the TopCoder applet.

1.5 Future Component Direction

None specified.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None required.

2.1.2 External Interfaces

None required.

2.1.3 Environment Requirements

- Development language: Java
- Compile Targets: JDK 1.4

2.1.4 Namespace

com.topcoder.processor.ipserver

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- None

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

None.

3.2.2 TopCoder Software Component Dependencies:

Heartbeat

****Please review the [TopCoder Software component catalog](#) for existing components that can be used in the design.**

3.2.3 Third Party Component, Library, or Product Dependencies:

None.

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

XML documentation must provide sufficient information regarding component design and usage.