

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.

The IPServer 2.0 component is an enhancement upon the IPServer 1.0 component. This version of the component will allow messages to be sent using any type of serialization.

1.2 Logic Requirements

1.2.1 Message Factory

Currently the component uses a generic message that is passed as both request and response messages. The new component must use a factory to return any type of message at runtime. Messages returned by the factory may use either `java.io.Serializable` or TopCoder's custom version of serialization or some other form of serialization.

1.2.2 Request Messages

Throughout the IPServer the *Message* class is used. The new version of the component must use a generic message that may implement any kind of serialization; it will not be limited to `java.io.Serializable`.

1.3 Example of the Software Usage

The IP Server will be used to manage such things as file upload and download requests that must have good performance on the TopCoder website.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

None.

2.1.3 Environment Requirements

- Development language: Java1.4
- Compile target: Java1.4

2.1.4 Package Structure

com.topcoder.processor.ipserver

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

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 1.0

Serialization 1.0

****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:

3.2.4 QA Environment:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003

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

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.