



Requirements Specification

1. Scope

1.1 Overview

The Compression Utility provides a mechanism for defining and executing compression/decompression algorithms using java. The component is designed to enable additional algorithms to be added with relative ease. The initial implementation supports the LZ77 compression algorithm.

1.2 Existing Requirements

1.2.1 Compression Utility

- Defines a component to support compression/decompression.
- Multiple compression algorithms can be supported. However, additional algorithms do not need to be "pluggable" in that they are configurable at runtime or through a property file.

1.2.2 Compression Algorithm

- The designer must define the compression/decompression algorithm.
- The first compression algorithm implemented will be the LZ77 compression and decompression algorithm.

1.3 Enhancement Requirements

1.3.1 Deflate Algorithms

- Define the compression/decompression deflate algorithms.
- The algorithm must support WinZip.

Documentation for the deflate algorithm can be found here:

<http://www.gzip.org/zlib/rfc-deflate.html>

<http://opensource.franz.com/deflate/>

Field Code Changed

Field Code Changed

1.3.2 Batch Compression/Decompression

- Provide the ability to support the zipping and unzipping of multiple files.
- Provide the ability to zip the contents of a specified root directory.

1.3.3 Relative Paths

- Preserve the relative path for each file being compressed.

1.4 Example of the Software Usage

The Compression Utility will be used in the TopCoder zip utility to compress/decompress files used in the zip application.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.



2.1.2 External Interfaces

The component should support a variety of input and output types including streams, File objects and StringBuffer.

2.1.3 Environment Requirements

- Development language: Java1.4
- Compile target: Java1.2, Java1.3, Java1.4

2.1.4 Namespace

com.topcoder.util.compression

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 Third Party Component, Library, or Product Dependencies:

None.

3.2.3 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.