BiGZiP.Com

Implementing the Plugin API Within BiGZiP

Introduction

- Purpose.
  - To show case BiGZiP and its Plugin API.
- Process.
  - Demonstrating BiGZiP’s functionality, reviewing the Plugin architecture and developing an example plugin.
- Product.
  - An understanding of how to develop plugins and what BiGZiP has to offer.

What is BiGZiP

- Java based compression utility

- Implements a simplistic / proven UI

- Support for:
  - Multiple Look’n Feels
  - Zip archives
  - 3rd party plugins
  - 100% Pure implementation

- Has a SEXY logo and animated icon

The User Interface

- Generic Screen and Toolbar/Menu
- Specific Plugin Toolbar/Menu
- Auto Plugin loading with manual fall back
- Plugin Configuration Screen
- Plugin Helpset integration

The Zip Plugin

- Open
  - Will read zip and self-extraction zip files
  - Reports the zip archive comment if present
- Add
  - Full Zip Archive support including multiple additions to an existing archive.
  - Multiple additions in 1 step via a mini-browser and a job queue
  - Support for comments attached to each entry
- Extract, Delete and View (TBA)

API Structure

- com.bigzip
- com.bigzip.plugin
- com.bigzip.plugin.compression
- com.bigzip.plugin.compression.data
- com.bigzip.plugin.compression.engine
- com.bigzip.plugin.compression.io
- com.bigzip.plugin.support
- com.bigzip.plugin.support.table
- com.bigzip.plugin.support.ui

com.bigzip.plugin.*

- Building the framework

- ArchiveAction.java
- ArchiveEntry.java
- ArchiveModel.java
- BiGZiPPlugin.java
- ColumnInfo.java
- DataSorter.java

Identifying Plugin to the UI

- All Plugins must implement ***BiGZiPPlugin***
  - Provides the associations and icon
  - General Plugin information
  - Allocates a help set
  - Feeds an instance of a settings object and a user registration object
  - Handles opening and creating the plugin’s archive type.
  - Should be placed in com.bigzip.archive.xxx
  - ./com/bigzip/resources/bigzip_plugin.config

Communicating with the UI

- UI coupling is achieved by sub-classing ***ArchiveModel***
- Plugin actions and functions are passed to the UI with an implemented event handler.
- Customization of the entry table by adding more columns and column types.
- General archive information like total size and number of entries.

Plugin Events/Actions

- All UI integrated actions must implement the ***ArchiveAction*** class

- Provides the toolbar and menu customization as well as support for animation

- Links into SwingWorker to correctly handle the appropriate thread

- Provides access to the selected entries in the table

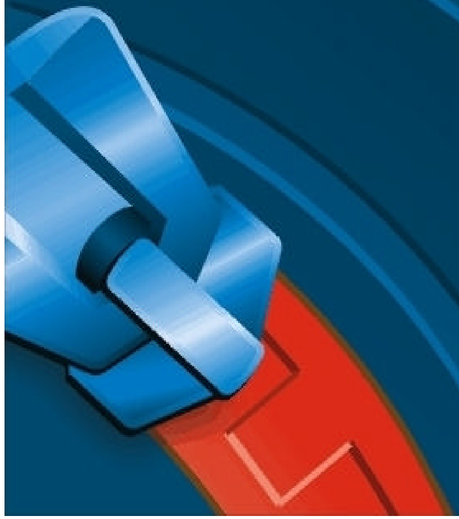
Inserting Entries into the Table

- ArchiveModel inserts entries into the table by passing instances of ***ArchiveEntry***
- Stores important fields for the UI
  - Size, Uncompressed Size
  - Entry ID
  - Entry Name
  - Attributes (Compressed and Directory)
- Stores custom information in an array of objects

Table Customization

- ***ColumnInfo*** is used for determining the data type and rendering of the column in the table and is parsed via ArchiveModel

- ***DataSorter*** is used to govern how each entry should be sorted within the column



Support

- Table Support
  - DateColumn.java
  - DateRenderer.java
  - DateSorter.java
  - DoublePrecisionColumn.java
  - DoublePrecisionRenderer.java
  - DoublePrecisionSorter.java
  - NumberColumn.java
  - NumberRenderer.java
  - NumberSorter.java
  - PercentColumn.java
- UI Support
  - DirectoryChooser.java
  - FileQueue.java
  - FileQueueAccessory.java
  - FileQueueEntry.java
  - StatusBarHelper.java
  - SwingWorker.java
  - WindowUtilities.java

com.bigzip.plugin.compression.io

- AbstractStream.java
- ArchiveStream.java
- CheckedStream.java
- CompressionStream.java
- IOStream.java
- NullCompression.java
- NullDecompression.java

IO Hierarchy and Format

- AbstractStream is based on the RandomAccessFile

- All IO streams must implement the AbstractStream

- The parent streams parsed to the instantiated stream must also implement the AbstractStream (as in the case of the IO streams in the java package)

Com.bigzip.plugin.compression.engine

- Required only if the compression stream is used
- Based on the GZIP standard

- Compressor.java
- Decompressor.java

Conclusion

- Fully JavaDoc’ed API is available on the Website
- An implemented example will be made available very shortly
- Developer site is being constructed
- Email developer@bigzip.com
- Website www.bigzip.com