ERSAP Framework Command Line Options

Overview

The ERSAP framework provides a command-line interface (CLI) tool called `ersap_shell` for managing and executing data processing workflows. The `ersap_shell` offers a set of commands for configuring, running, and monitoring ERSAP-based processing pipelines. This document provides an overview of the available commands and their options.

Command List

The following commands are available in `ersap_shell`:

- `run` - Start ERSAP data processing
- `edit` - Edit data processing conditions
- `set` - Configure parameters for data processing
- `show` - Display current settings and status
- `save` - Export configuration to a file
- `source` - Execute commands from a script file

For more details on each command, use `help <command>` in the `ersap_shell` prompt.

Running ERSAP Data Processing

`run` Command

The `run` command starts the ERSAP data processing pipeline. It supports two modes:

- `run local` - Runs ERSAP on the local node.
- `run farm` - Runs ERSAP on a distributed computing farm.

Configuring ERSAP Data Processing

`set` Command

The `set` command allows users to configure various parameters for data processing. The key options include:

- `set servicesFile` - Path to the YAML file defining the service composition.
- `set files` - Specify input files to process (e.g., `/mnt/data/files/*.evio`). This sets both `fileList` and `inputDir`.
- `set fileList` - Path to a file containing names of input data files.

- `set inputDir` - Directory containing the input files.
- `set outputDir` - Directory to save processed files.
- `set outputFilePrefix` - Prefix for output filenames (single word, no spaces, preferably with a separator like _ or -).
- `set threads` - Maximum number of processing threads per node.
- `set reportEvents` - Frequency of reporting processed events.
- `set skipEvents` - Number of events to skip from the input file.
- `set maxEvents` - Maximum number of events to process.
- `set logDir` - Directory to store log files.
- `set feHost` - IP address for the front-end DPE.
- `set fePort` - Port number for the front-end DPE.
- `set session` - A single-word identifier for the data processing session.
- `set description` - A single-word description of the data processing task.
- `set javaMemory` - JVM memory size for DPE (in GB).
- `set javaOptions` - JVM options for DPE (overrides `javaMemory`).
- `set monHost` - IP address of the DPE monitoring server.

Displaying Configuration and Status

`show` Command

The `show` command retrieves various aspects of the current ERSAP configuration and status. Available options include:

- `show config` - Display current configuration settings.
- `show services` - Display the service YAML configuration.
- `show files` - List input files.
- `show inputDir` - List files in the input directory.
- `show outputDir` - List files in the output directory.
- `show logDir` - List log files directory.
- `show logDpe` - Display front-end DPE logs.
- `show logOrchestrator` - Display orchestrator logs.
- `show farmStatus` - Display the status of farm-submitted jobs.
- `show farmSub` - Display farm job submission file.
- `show version` - Display the ERSAP version.

Saving and Loading Configurations

`save` Command

The `save` command exports the current ERSAP configuration to a specified file:

```
save <file_path>
```

`source` Command

The `source` command reads and executes a set of commands from a script file:

```
source <file_path>
```

Summary

The ERSAP command-line interface provides a flexible and efficient way to configure and execute data processing pipelines. By leveraging commands like `set`, `run`, `show`, `save`, and `source`, users can effectively manage their ERSAP workflows.