CIShell Basics

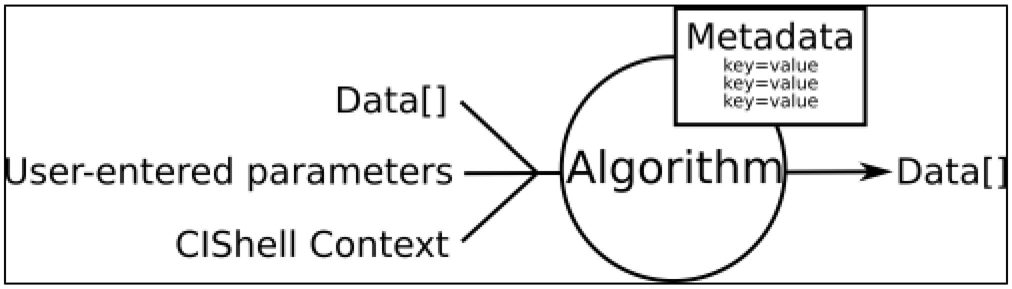
Prerequisites

None.

CIShell Basics

The CIShell Platform has been specifically design around the idea of the algorithm. It is the central and most important concept. Algorithms are fully defined and self-contained bits of execution. They can do many things: data conversion, data analysis, and even spawn whole external programs if needed. Algorithms are well defined black boxes, which can contain either Java code or any program which can be compiled. Creating new algorithms is the primary method to extend CIShell tool's functionality, or creating new CIShell-based tools.

CIShell is based on OSGi, which is a plugin and service based framework. Practically this means that OSGi functionality is divided into plugins or bundles (Java jar files with some additional special files), each of which contains code to create some number of services at runtime. These services are the main actors in the OSGi environment. In CIShell almost all services are algorithms, which means they conform to a certain interface, allowing algorithms to interoperate with the CIShell environment and each other in a well-defined way. Specifically, algorithms accept Data[], user-input parameters, and a CIShellContext. They output Data[].



CIShell provides an environment which makes it easy for users to interact with a set of algorithms in the form of an executable tool. This environment includes a Menu Manager which allows users to invoke algorithms, a Data Manager which serves as workspace to hold data while users run a series of algorithms on it, a scheduler which monitors algorithms as they run, a conversion service which converts data between various types so algorithms can operate on data in the format of their choice, and several other services. Since all of this is provided by default, developers can maintain focus on the algorithms they wish to implement without having to reinvent all the supporting infrastructure.

Currently the CIShell environment is implemented as an Eclipse-based desktop tool, but the CIShell interfaces are defined in such as way that the environment could be implemented in a variety of ways, including as a web-based service.

Since CIShell is so closely tied to OSGi, many references will be made to OSGi in the developer documentation. To fully understand the details of how CIShell works, it is often necessary to understand certain aspects of OSGi, however most developers should be able to begin working with CIShell without understanding OSGi.

Tutorial 0: Setting Up the Development Environment