PAUL SCHERRER INSTITUT

PSI

Java Channel Access - CA

https://github.com/
channelaccess/ca

Outline

- Acknowledgements
- Features
- Getting Started
- Commands
- Documentation
- Examples
- Issues / Contribute / Feedback

Acknowledgements

- **Matej Sekoranja** (Cosylab) - Implementation
- **Tom Slejko** (Cosylab)

Features

- Java style Channel Access library
- No prerequisites other than **Java 8**
- Use of modern Java 8 concepts
- (Developer) Performance
- Easy deployment of the library (Maven, Gradle, ...)
    - Available on Maven Central

Features

- Simplicity
- Use of Java type system
- Synchronous and asynchronous operations for get, put, connect
- Efficient handling of parallel operations without the need to use threads
- Chaining of actions/operations, e.g. set this, then set that, … (Reactive Programming)
- Easily get additional metadata to value: Timestamp, Alarms, Graphic, Control
- Support of all listeners ChannelAccess supports: ConnectionListener, AccessRightListener, Value Listener (Monitor)

Basics

- Context
  - Used to create channels
- Channel
  - Class/Object representing a Channel Access channel (connection)
- Future
  - Handle for a result that will be available in the future
  - For asynchronous operations (suffix **Async**)

- Channels
  - Utility class to facilitate various (non standard Channel Access) operations

Basics

- The library always provides synchronous and asynchronous operations.
- Asynchronous operations just have the postfix **`Async`**
- Asynchronous operations always return a `CompletableFuture`

Getting Started - Get The Library

Manual

- Download latest release: https://github.com/channelaccess/ca/releases

Gradle

- Add following line to the dependency section of your build.gradle

```
compile 'org.epics:ca:1.0.0'
```

Maven

- Add following lines to the depency section of your Maven pom.xml

```
<dependency>
  <groupId>org.epics</groupId>
  <artifactId>ca</artifactId>
  <version>1.0.0</version>
</dependency>
```

Getting Started - Use

```
try (Context context = new Context())
{
     Channel<Double> channel =
          context.createChannel("MY_CHANNEL",Double.class);
     channel.connect();
     System.out.println(channel.get());
     channel.close();
}
```

Getting Started - Context

Specify context properties

```
Properties properties = new Properties();
properties.setProperty(Context.Configuration.EPICS_CA_ADDR_LIST.toString(),"sls-cagw");
properties.setProperty(Context.Configuration.EPICS_CA_SERVER_PORT.toString(),"5062");


Context context = Context(properties);
```

Getting Started - Channel

Synchronous

```
Channel<Double> channel = context.createChannel("MY_CHANNEL", Double.class);
channel.connect();
```

Asynchronous

```
Channel<Integer> channel1 = context.createChannel("adc02", Integer.class);
Channel<String> channel2 = context.createChannel("adc03", String.class);
// Wait for all channels to be connected
CompletableFuture.allOf(channel1.connectAsync(), channel2.connectAsync()).get();
```

… with Timeout

```
channel.connectAsync().get(1, java.util.concurrent.TimeUnit.SECONDS);
```

Getting Started - Channel

- If you want a generic type of channel use:

```
Channel<Object> channel = context.createChannel("MY_CHANNEL", Object.class);
```

- When doing a get/put you will get the correct native type

Getting Started - Commands

Get

```
channel.get();
channel.getAsync();
channel.get(Graphic.class);
```

Put

```
channel.put(1.0);
channel.putAsync(1.2);
// Best effort put
channel.putNoWait(3.11);
```

Getting Started - "Metadata Types"

- Timestamped
- Alarm
- Graphic
- Control

Getting Started - Monitor

Add Monitor

```
Monitor<Double> monitor = adc.addValueMonitor(value -> System.out.println(value));
Monitor<Timestamped<Double>> monitor = channel.addMonitor( Timestamped.class,
    value -> { if (value != null) System.out.println(new Date(value.getMillis()) +
    " / " + value.getValue()); }
    );
```

Remove Monitor

```
monitor.close()
```

Getting Started - Listeners

Connection Listener

```
Listener listener = channel.addConnectionListener(
    (channel, state) -> System.out.println(channel.getName()+" is connected? "+state));
```

Access Rights Listener

```
Listener listener = channel.addAccessRightListener(
    (channel, rights) -> System.out.println(channel.getName()+" is rights? "+rights));
```

Remove Listener

```
listener.close()
```

More Stuff …

https://github.com/channelaccess/ca

https://github.com/
channelaccess/ca

Documentation

https://github.com/channelaccess/ca

https://github.com/
channelaccess/ca

Examples

https://github.com/channelaccess/ca/blob/master/src/test/java/org/epics/ca/test/Example.java

Issues / Contribute / Feedback

- **Issues** / **Bugs**: https://github.com/channelaccess/ca/issues
- **Contribute** by cloning the repository on github.com and file your pull request !
- General feedback, useful code snippets, examples : simon.ebner@psi.ch

PAUL SCHERRER INSTITUT

Details

https://github.com/channelaccess/ca

Contact

Simon Ebner
Paul Scherrer Institute
WBGB/001
5232 Villigen PSI

simon.ebner@psi.ch

PAUL SCHERRER INSTITUT

PSI

Questions ?

Colors