Featherweight Getting Started

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Author | Date | Comments |
| A | E. Metzler |  | Original |
|  |  |  |  |

# Specs

* LV 2013+

# Add to a project

1. Add a property to your project source directory
2. Add an extern to the branch of choice to a local directory called Featherweight. Only the bold part of the URL is required if adding to a Viewpoint project in SVN.
   1. [https://sc.viewpointusa.com**/svn/featherweight/trunk**](https://sc.viewpointusa.com/svn/featherweight/trunk) - releases only, most stable
   2. https://sc.viewpointusa.com**/svn/featherweight/tags/x.x.x** – a specific release, most stable
   3. [https://sc.viewpointusa.com**/svn/featherweight/branches/develop**](https://sc.viewpointusa.com/svn/featherweight/branches/develop) - newest features, very stable
   4. https://sc.viewpointusa.com**/svn/featherweight/branches/feature-xxxx** – feature in development, not stable
3. Commit
4. Update

# Where to start

[\\ROCHNAS02\Data\Engineering\In House Training\Presentations\_Brown Bag\Featherweight\Featherweight BBL.pptx](file:///\\ROCHNAS02\Data\Engineering\In%20House%20Training\Presentations_Brown%20Bag\Featherweight\Featherweight%20BBL.pptx)

Check out examples for very simple examples of implementing actors, connectors

## Successful projects for reference

Exelis WBIF

GWave

MITRE

MEPPI

VCTDAQ

# Actors

An interface to a specific resource such as:

* An instrument
* A database
* A piece of equipment
* A file

Kind of like a scalable AE

Takes requests, cannot be directly commanded to do something

# Connector

The interface between an actor and a caller

# Job

An internal action to an actor

Can be queued up within an actor

Can easily be replaced with VIs, but reduces file volume for simple actors

# Job Sequence

A predefined sequence of jobs

# Baked jobs

## FTW: Initialize

Called after critical startup jobs

Responds to launcher that the actor is fully initialized

## FTW: Event Handler

Waits for a request or user input

Default job when job queue is empty

## FTW: Error Handler

Executes when a previous job has an outgoing error

Logs the error, performs any other custom error handling, then clears the error and continues with the next job in the queue

## FTW: Shutdown

The last job that is executed when an actor is shutting down

Will stop the while loop, so no subsequent jobs will execute

# Best practices

Requests for asking an actor to do something

Publish data when an actor needs to tell the world about itself (status/state usually) as opposed to queries

Use actor init for deserializing configuration string only

Perform actor launch critical jobs before FTW: Initialize and use **FTW-Actor-OnError-BeginShutdownSequence.vi** to shut down if critical steps error

# Connection Strings

Connection strings define the transport mechanism that is used for requests or published messages.

The available transports are In Process (inproc) and TCP.

Connection strings are specified similar to URLs with a qualifier.

Connection strings can be randomized using wildcards, “\*”. Any asterisk will be replaced by a 32-character random string.

## Inproc

Inproc is used for *intra*-process communication. Inproc is much faster than TCP, but it is limited to wherever a LabVIEW queue can be used, so not between executables or between machines.

### Format

inproc://<anything>

## TCP

TCP is used for *inter*-process communication.

### Format

tcp://<IPaddress>[:port][/namedPort]

## Examples

|  |  |
| --- | --- |
| Input String | Resolved String |
| inproc://\* | inproc://6039B8E0AF1136A009EFF29CA4EA4B95 |
| inproc://pid-loop-\* | inproc://pid-loop-FC6EC6CC88D25DB46AD8B440D76B54B8 |
| tcp://localhost:51213 | tcp://localhost:51213 |
| tcp://localhost/\* | tcp://localhost:56789/A6B5A0C0D72D010812C706B847D4CE60 |