Featherweight Getting Started

**Revision History**

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

Contents

[1 Specs 3](#_Toc468113899)

[2 Add to a project 3](#_Toc468113900)

[3 Where to start 3](#_Toc468113901)

[3.1 Successful projects for reference 3](#_Toc468113902)

[4 Actors 3](#_Toc468113903)

[5 Connector 4](#_Toc468113904)

[6 Job 4](#_Toc468113905)

[7 Job Sequence 4](#_Toc468113906)

[8 Baked jobs 4](#_Toc468113907)

[8.1 FTW: Initialize 4](#_Toc468113908)

[8.2 FTW: Event Handler 4](#_Toc468113909)

[8.3 FTW: Error Handler 4](#_Toc468113910)

[8.4 FTW: Shutdown 5](#_Toc468113911)

[9 Endpoint Address Strings 5](#_Toc468113912)

[9.1 Inproc 5](#_Toc468113913)

[9.2 TCP 5](#_Toc468113914)

[9.3 Examples 6](#_Toc468113915)

[10 Best practices 6](#_Toc468113916)

# 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
* DRAP

# Actors

Think of an actor like a scalable action engine.

An interface to a specific resource such as:

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

Takes requests, cannot be directly commanded to do something

# Connector

The interface between an actor and a caller

Connector VIs are synonymous with action engine actions

# Job

An internal action within an actor

Can be queued up within an actor, but not by external callers

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

# Job Sequence

A predefined sequence of jobs

# Baked jobs

These jobs are included in the actor template and should not be removed.

## FTW: Initialize

Called after critical startup jobs

Responds to launcher that the actor is fully initialized

Returns a string to the launch caller that can be used to send launch information to the caller, such as resolved addresses for endpoints.

## 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

# Endpoint Address Strings

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

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

Address strings are specified with a qualifier, similar to URLs.

Address 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.

### Formats

tcp://<domain>[:port][/serviceName]

| Format | Examples | Description |
| --- | --- | --- |
| tcp://<domain>:<port> | tcp://localhost:45678  tcp://169.254.1.2:34567 | Specific IP address and port |
| tcp://<domain>/<serviceName> | tcp://localhost/\*  tcp://169.254.1.2/analogInput | A named service will query the NI Service Locator for the port number registered with the service name and use that port. The port does not need to be specified. |
| tcp://<domain>:<port>/<serviceName> | tcp://169.254.1.2:56789/analogInput | This format will only be output by a TCP endpoint when a service name is used. This should not be used as an input for connecting other endpoints. |

## Examples

| Input String | Resolved String |
| --- | --- |
| inproc://\* | inproc://6039B8E0AF1136A009EFF29CA4EA4B95 |
| inproc://pid-loop-\* | inproc://pid-loop-FC6EC6CC88D25DB46AD8B440D76B54B8 |
| tcp://localhost:51213 | tcp://localhost:51213 |
| tcp://169.254.2.3/\* | tcp://169.254.2.3:56789/A6B5A0C0D72D010812C706B847D4CE60 |

# Best practices

* Use 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 initialization actions in jobs.
* Perform actor launch critical jobs before FTW: Initialize and use **FTW-Actor-OnError-BeginShutdownSequence.vi** to shut down if critical jobs error.
* Use local message worker for periodic actions