Passage du Cardinal 1  
1700 Fribourg  
Switzerland

Iterator Actor Documentation

|  |  |
| --- | --- |
| Document responsible: | Albert Adiyatullin |
| Last changed | 14.02.2018 |
| Description: | This document contains the documentation of the Iterator Actor framework. |

Table of Contents

[Description 2](#_Toc505701646)

[API 3](#_Toc505701647)

[Property: Settings 3](#_Toc505701648)

[Method: Start Iterations 3](#_Toc505701649)

[Method: Stop Iterations 3](#_Toc505701650)

[Message: Report Error 3](#_Toc505701651)

[Other methods 3](#_Toc505701652)

[Check Settings 3](#_Toc505701653)

[Prepare 4](#_Toc505701654)

[Iterate 4](#_Toc505701655)

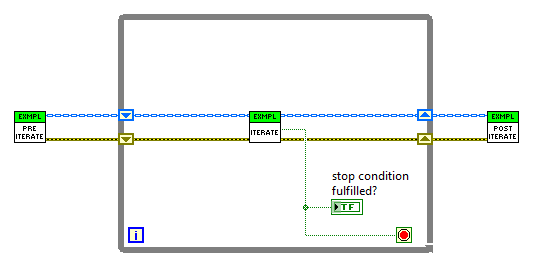
[Conclude 4](#_Toc505701656)

[Error codes 4](#_Toc505701657)

# Description

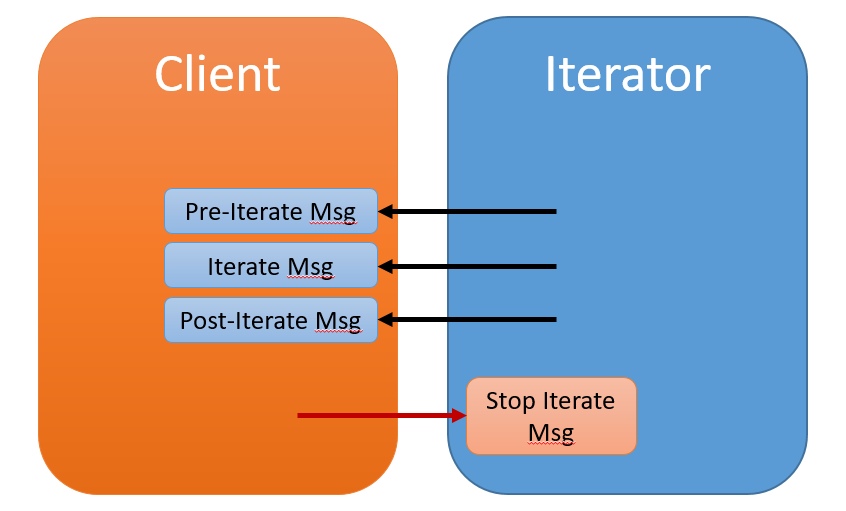
The Iterator Actor framework provides a possibility to stop the action loops in the Actor framework on demand.

If some actor method is called by a message and contains a loop like this:



then it cannot be stopped on demand, because any new message will be processed only after this method will complete.

The Iterator Actor is launched by the main (Client) actor and has messages to start the Pre-Iterate, Iterate, and Post-Iterate methods. Therefore, it can launch these methods, and at the same time be stopped by the Client actor.



# API

The Iterator Actor has only two API methods called using standard messages, and one Property called via a property node or a standard message.

## Property: Settings

Defines the settings of the Iterator.

The Client should provide:

* Three message classes that call your Preparation (optional), Iteration, and Conclusion (optional) methods. All three classes should inherit from the Iterator Reply Msg with only the method Do Core overridden. This is necessary for checking the timeouts of all the actions.

N.B. Classes for Preparation and Conclusion are optional: if not defined, the Iterator will skip them and go to the next step.

* Cluster with Iterator settings:
  + Timeouts for Preparation, each of Iteration, and Conclusion phase
  + Periodicity of Iterations.
  + Way to stop the Iterations:
    - None (stopped only by Stop Iterate message)
    - After a given number of Iterations (should be provided)
    - After a certain time (should be provided)
    - If the previous Iteration returns FALSE
* (optional) A message class that receives the errors returned by the Iterator. It should inherit from the Iterator Report Error Msg class.

## Method: Start Iterations

Launches the sequence of calls starting from the Prepare stage. Does not have any input parameters.

## Method: Stop Iterations

Stops the iterations and moves to the Conclude stage. Does not have any input parameters.

## Message: Report Error

Returns the resulting Iterator error to the Caller. Also returns the Boolean indicating if the Stop Iterations method has been called during the sequence.

# Other methods

As soon as the Client configures the Iterator, and all the settings are correct, Iterator starts the sequence. It is performed in the Helper Loop of the Actor Core, and is split in 5 subVIs for the sake of simplicity.

N.B. Since these VIs are executed in the Helper Loop, they do not address the Actor cluster.

## Check Settings

Checks the Settings and returns an error if something is wrong.

* all timeouts should be positive or -1 (in case there is no timeout)
* iteration period should be positive and not shorter than the iteration timeout
* depending on the Stop condition, the number of iterations or the time before stop should be non-negative

## Prepare

Sends the Prepare message to the Caller and waits for a response. Depending on the response, proceeds or not with the iterations.

## Iterate

Sends the Iterate message to the Caller until they are stopped by the Stop Iterate message, or any of the Stop conditions.

## Conclude

Sends the Conclude message to the Caller. After this message, the Event

After all the iterations, the Helper loop sends a Report Error message, reporting the end of the sequence. If there is no error, an empty error cluster will be sent.

At the very end of the sequence, the Helper loop flushes the queue of incoming Events. Therefore, the Start Iterations commands received during the sequence of iterations will not be executed.

# Error codes

7900: Input parameters are not valid

7901: Preparation call timed out

7902: Iteration call timed out

7903: Conclusion call timed out (warning)