

Breaking Changes

Russell Winder

Version 1.0, 2015-10-01

Table of Contents

Breaking Changes.....	1
1.2.0	1
1.1.0	1
1.0.0	1
0.12	1
0.11	1
0.10	1

Breaking Changes

1.2.0

- No breaking changes

1.1.0

- Deprecated *foldParallel()* in favor of *injectParallel()* on parallel collections

1.0.0

- Removed all deprecated methods and classes
- The *stop()* method on Dataflow operators has been renamed to *terminate()* to preserve naming consistency accross concepts
- The *_reportError()* method on Dataflow operators has been replaced with *addErrorHandler()*
- The *RightShift (>>)* operator of *DataflowVariables* and channels now calls *then()* instead of *whenBound()* and so can be chained

0.12

- Deprecated the *makeTransparent()* method (use *makeConcurrent()* instead)
- Removed deprecated actor classes - *AbstractPooledActor*
- Created *BlockingActor* for fast thread-bound actors

0.11

- Deprecated *AbstractPooledActor*
- Actor timeout doesn't terminate the actor, but passes a *TIMEOUT* message to the message handler
- Created *DefaultActor* as a replacement for *_AbstractPooledActor_* with the following differences
- removed the *receive* method for blocking message read
- messages are not enhanced with the *reply()* method nor the *sender* property
- The *react()* method doesn't throw controlled exception to clear the stack. It is the user responsibility to return from the actor body in order to allow the react block to be scheduled

0.10

Renaming Hints

- Parallelizer → GParsPool
- Asynchronizer → GParsExecutorsPool
- doParallel() → withPool()
- withParallelizer() → withPool()
- withExistingParallelizer() → withExistingPool()
- withAsynchronizer() → withPool()
- withExistingAsynchronizer() → withExistingPool()
- orchestrate() → runForkJoin()
- ActorGroup → PGroup
- PooledActorGroup → DefaultPGroup
- NonDaemonActorGroup → NonDaemonPGroup
- Safe → Agent