Kepler Actor Reference

The following pages contain documentation for the actors that ship with Kepler's standard library. The Table of Contents is arranged by function, and the actors are categorized as they are in the Kepler library (e.g., "Data Input", "Data Output", etc). The pages in this reference are ordered alphabetically by actor name, and actors can be browsed by name via the PDF bookmarks, or with the alphabetical index at the end of the reference.

Logger
MappedLogger
MatrixViewer
MonitorValue
Recorder

....

Director

....

CTDirector
DDFDirector
DEDirector
PNDirector
SDFDirector

....

File System

....

DirectoryMaker.html
FileCopier.html
FileCopy.html

....

General Purpose

....

ArchiveCounter
CompositeActor
Expression
MatlabExpression
RExpression

General Purpose: Grid Function

Authentication
GlobusJob
GlobusProxy
GridFTP

ParameterizedGlobusJob
RunJobGridClient
UpdatedGridFTP

General Purpose: Grid Function: Griddles
GriddlesExec
GriddlesInputFile
GriddlesOutputFile
GriddlesParameter
JCOGGridFTP
JCOGPROXYExec
JCOGWorkflowExec
JGridletCreator

General Purpose: Grid Function: Nimrod
ExperimentMonitor
ExperimentPreparator
ExperimentStarter
ForkResourceAdder

General Purpose: Job Command
JobCreator
JobGetDirectory
JobGetRealJobID
JobManager
JobRemover
JobStatus
JobSubmitter

General Purpose: SSH
SshDirectoryCreator
SshDirectoryListing
SshExecuteCmd
SshFileCopier
SshFileRemover
SshSession

General Purpose: Unit System
BasicUnits

Polygon
Rectangle

Workflow: Workflow Input
Interpolator
Ramp

Workflow: Workflow Input: Constant
Constant
StringConstant

Workflow: Workflow Input: Parameter
ColorParameter
FileParameter
Parameter
ParameterSet
PortParameter
RequireVersion
ScopeExtendingAttribute
SemanticTypeParameter
StringParameter

....
Projects
....

CIPRes
GUIRunCIPRes
Initializer
NexusFileReader
PAUPInfer
PhyloDataReader
RecIDCM3
SubsetChooser
TreeDecomposer
TreeImprover
TreeMerger
TreeParser

TreeToString
TreeVizForester

CPES
SshDirectoryCreator
SshDirectoryList
SshExecuteCmd
SshFileCopier
SshFileRemover
SshSession
SyncOnTerminator

Computational Chemistry
Babel.html
GAMESSInputGenerator
GAMESSLocalRun
GAMESSNimrodRun
MoleculeSelector
OpenBabel
QMViewDisplay

Computational Chemistry: GAMESS Input
DataGroup
EndGAMESSInput
FormattedGroup
KeywordGroup
StartGAMESSInput

Computational Chemistry: Process Utility
FileExistenceMonitor
FileListSequencer
FileLocationChooser
FileNameChooser
GAMESSAtomDataExtractor
GAMESSKeywords
MoleculeArrayProducer
TemporaryScriptCreator

SEEK: Niche Modeling
AscGridValue

ClimateChangeFileProcessor
ClimateFileProcessor
GARPAlgorithm
GarpPrediction
GARPPresampleLayers
GARPSummary
GridRescaler
GridReset

....
Statistics
....

Sampling Distribution
RicianDistributionRandomNumberGenerator

Sampling Distribution: Normal Distribution
GaussianDistributionRandomNumberGenerator
RandomNormal

Sampling Distribution: Uniform Distribution
RandomUniform

Statistical Analysis: Bivariate Analysis
LinearModel

Statistical Analysis: Bivariate Analysis: Correlation Analysis: Kendall Correlation
Correlation

Statistical Analysis: Bivariate Analysis: Correlation Analysis: Pearson Correlation
Correlation

Statistical Analysis: Bivariate Analysis: Correlation Analysis: Spearman Correlation
Correlation

Statistical Analysis: Bivariate Analysis: Regression Analysis
LinearModel

Regression

Statistical Analysis: Graphical Analysis
TimedPlotter

Statistical Analysis: Graphical Analysis: Bar Chart
BarGraph
Barplot

Statistical Analysis: Graphical Analysis: Box and Whisker Plot
Boxplot

Statistical Analysis: Graphical Analysis: Scatter Plot
ArrayPlotter
Scatterplot
XYPlotter
XYScope

Statistical Analysis: Univariate Analysis
Summary

Statistical Analysis: Univariate Analysis: Analysis of Variance
ANOVA

Statistical Analysis: Univariate Analysis: Central Tendency
ArrayAverage
Average
RMean
RMedian

Statistical Analysis: Univariate Analysis: Data Transformation: Rank
ArraySort

Statistical Analysis: Univariate Analysis: Data Transformation: Scaling
ComplexToCartesian
ComplexToPolar
PolarToCartesian
PolarToComplex
Rescaler

Statistical Analysis: Univariate Analysis: Dispersion

ANOVA (org.ecoinformatics.seek.R.RExpression)

Author: Dan Higgins **Version:** Unknown	The ANOVA actor uses R to perform a variance analysis on input data. The actor outputs a graphical representation of its calculations. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. Input factors might include 'date', 'site,' or 'replicate' data. In addition to factors, a variable (usually a continuous variable such as 'height' or 'rain fall') must be specified via the variable port. The actor performs the analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

Method	The summary method. The default method is 'presence.' Use the Constant actor to input another summary method (e.g., mean, max, sum, etc.)
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Variable	The variable to be summarized.
Factor5	Any factor of interest (e.g., 'date' or 'site' data)

graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
Factor4	Any factor of interest (e.g., 'date' or 'site' data)
Factor3	Any factor of interest (e.g., 'date' or 'site' data)
Factor2	Any factor of interest (e.g., 'date' or 'site' data)
Factor1	Any factor of interest (e.g., 'date' or 'site' data)

ASC To RAW (util.AscToRaw)

Author: unknown **Version:** Unknown	The ASCToRaw actor converts ASCII raster files (*.asc) to binary files (*. raw) that can be used as environmental layers (e.g., climate, rainfall, or temperature) in GARP. GARP is a genetic algorithm that creates an ecological niche model representing the environmental conditions where a species would be able to maintain populations. For more information about GARP, see http://www.lifemapper.org/desktopgarp/. The ASCToRaw actor receives either an array of ASCII file names or a single ASCII file to convert (via the inputAscFilenameArrayPort or singleFileNamePort input port respectively.) If a single file is specified, the actor outputs the name of the converted binary file. If an array of files is specified, the actor outputs the name of an XML file (*.dxl) that is used by GARP to summarize a layer list. The actor creates the *.dxl file in addition to converting and saving the input files.

Parameters

outputRawFilename	The name of the converted file. The file name is output as a string when a single ASCII file is input via the singleFilenamePort.
dxlFilename	The name of the *.dxl file. The *.dxl file name is output as a string when an array of ASCII file names is input via the inputAscFilenameArrayPort.
EnvLayerSetIdParameter	The ID of the environmental layer set. The specified ID is used in the generated *.dxl file.
EnvLayerSetTitleParameter	The title of the environmental layer set. The specified title is used in the generated *.dxl file.
scaleRaw	The ASCToRaw actor automatically scales ASCII data values so that they fall within the required binary range (0-255). In some cases, it is not desirable to scale the input file (e.g., if the ASCII file already represents data that is in the range of 0-255). Set the scaleRaw parameter to false to prevent data scaling. Note that a false value is ignored if the ASCII range falls outside 0.0 to 255.0.

Ports

outputValuesPort	An output port that broadcasts a *.dxl file name (if an array of ASCII file names is input via the inputAscFilenameArrayPort) or a single binary file name (if a single ASCII file is input via the singleFilenamePort).
inputAscFilenameArrayPort	An input port that accepts an array of ASCII raster file names. The ASCII files must have the same extent and number of cells. If an array is specified via this port, the actor will convert each file to a binary format that can be used as a GARP spatial layer. The actor will also create a *.dxl file that summarizes the layer list. The name of the *.dxl file is output.
singleFilenamePort	An input port that accepts a single ASCII file name. If a single file is specified via this port, the actor will convert the file to a binary format that can be used as a GARP spatial layer input. The name of the binary file is output.

Absolute Value (ptolemy.actor.lib.AbsoluteValue)

Author: Edward lee
Version: Unknown

The AbsoluteValue actor reads a scalar value (e.g., an integer, double, etc) and outputs its absolute value. The output will have the same type as the input, unless the input is a complex number. If the input is a complex number, the output type is a double with the same magnitude as the input number.

Ports

output	An output port that broadcasts the absolute value of the input.
input	An input port that accepts a scalar value (e.g., an integer, double, complex, etc.)

Accumulator (ptolemy.actor.lib.Accumulator)

Author: Edward Lee
Version: Unknown

The Accumulator outputs the sum of its initial value plus all of the inputs it has received since the last time a true token was received at the reset port. One output is produced each time the actor is fired. The inputs and outputs can be any token type that supports addition. The output type must be greater than or equal to the input type and the type of the init parameter.

Parameters

init	The value produced by the actor on its first iteration. The default value of this parameter is the integer 0.

Ports

output	An output port that broadcasts the sum of the actor's initial value plus the input values. By default, the type of this output is constrained to be at least that of the input.
input	A multiport that accepts values of any type that supports addition.
reset	A multiport that accepts Boolean tokens. If this port receives a true token on any channel, then the accumulator state will be reset to the initial value (specified with the init parameter).

Add Grids (org.ecoinformatics.seek.gis.java_gis.AddGrids)

Author: dan higgins
Version: Unknown

The AddGrids actor reads two or more geospatial image files, merges the files by adding the cell values, and outputs the name of the merged file. The gridFilenameArray port accepts an array of image file names. The extent and cell size of the merged file will match that of the first input file. The name of the merged file is specified via the mergedGridFilename port. The actor's primary purpose is to combine stochastic files to give a spatial distribution where more probable cells have larger values. Thus, cell values are added for all pixels in the input file list. The AddGrids actor is similar to the MergeGrids actor, except that AddGrids actor can be used to merge multiple files, while MergeGrids can merge only two.

Ports

mergedGridFileResult	An output port that broadcasts the file name of the merged raster file (acts as a trigger when addition is complete).
mergedGridFileName	An input port that accepts the name given to the resulting output file
gridFilenameArrayPort	An input port that accepts a string array of file names for the raster files to be added.

Add or Subtract (ptolemy.actor.lib.AddSubtract)

Author: yuhong xiong and Edward lee
Version: Unknown

The AddOrSubtract actor reads values via its two input ports (plus and minus), performs an add and/or subtract operation, and outputs the result. The input ports are multiports, meaning that they can accept multiple inputs. Any values received via the plus port will be added; any values received via the minus port will be subtracted. Either port can be left unconnected. Leave the minus port unconnected to create a simple adder, for example. Both of the input ports are polymorphic, accepting data of multiple types. The actor will automatically resolve the input type to the least upper bound of the presented values. For example, if the plus input port receives a Boolean value on one input channel and an integer on another, the resolved type will be a string, permitting the two inputs to be concatenated as strings. Note that strings cannot be subtracted. If the actor resolves an input type into a type that cannot be subtracted, it will generate an error. The actor outputs the sum or difference and derives an output type based on the input values.

Ports

output	An output port that broadcasts the sum or difference of the inputs. The actor derives the output type based on the type of the inputs.
minus	A muliport that accepts values to be subtracted. The actor automatically infers the input type based on the type of the input values.
plus	A muliport that accepts values to be added. The actor automatically infers the input type based on the type of the input values.

Annotation (ptolemy.vergil.kernel.attributes.TextAttribute)

Author: Edward A. Lee
Version: Unknown

Use the Annotation attribute to annotate a workflow. Double-click the attribute to customize the content, font size, color, and basic formatting (bold and italics).

Parameters

fontFamily	The font face. Select a font from the drop-down menu. The default is "SansSerif".
italic	Select to indicate that the type should be italics. By default, type will not be italicized.
bold	Select to indicate that the type should be bold. By default, type will not be bold.
textColor	The font color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. The default is an opaque black, {0.0, 0.0, 0.0, 1.0}
textSize	The font size. Select an integer from the drop-down menu or specify a new size by typing it directly into the value field. The default is 14.
text	The text of the annotation.

Archive Counter (org.sdm.spa.ArchiveCounter)

Documentation coming soon!

Array Accumulator (org.resurgence.actor.ArrayAccumulator)

Author: Wibke Sudholt **Version:** Unknown	The ArrayAccumulator actor reads an array and outputs a string containing all of the array elements. The characters separating the entries in the output string can be specified with the elementSeparator parameter. The ArrayAccumulator actor is similar to the ArrayToSequence actor, except that the ArrayAccumulator outputs values as a string, while the ArrayToSequence actor outputs elements as a sequence via a single output port. Note: Use the ArrayToElements actor to output individual array values on multiple channels of the output port.

Parameters

Element separator	The character used to separate the elements in the output string. By default, the value is "".

Ports

string	An output port that broadcasts a string consisting of all of the array elements.
array	An input port that accepts an array of elements of any type.

Array Average **(ptolemy.actor.lib.ArrayAverage)**

Author: Mark Oliver, Edward A. Lee
Version: Unknown

The ArrayAverage actor reads an array of values and outputs the average of the values. The elements of the input array must support addition and division by an integer; otherwise, the actor will generate an error.

Ports

output	An output port that broadcasts the average of the input values.
input	An input port that accepts an array of values. The values must support addition and division by an integer; otherwise, the actor will generate an error.

Array Contains Element (org.geon.ArrayContains)

Author: null
Version: Unknown

The ArrayContainsElement actor reads an array and determines whether a specified element is contained in it. The actor outputs a Boolean value: true if the element is contained in the array, false if not. The actor accepts an array via the input port and an element to "look for" in that array. The element is passed via either the element port or parameter.

Parameters

element	An element to "look for" in the array. The value can be specified with either the element port or parameter.

Ports

array	An input port that accepts an array (e.g., {1,2,3,4,5}).
element	An input port that accepts an element to "look for" in the array.
output	An output port that broadcast a Boolean value: true if the element is contained in the array, false if not.

Array Element (ptolemy.actor.lib.ArrayElement)

Author: null
Version: Unknown

The ArrayElement actor reads an array of elements and "looks up" an element specified by a given index number. The specified element is output. The first element in an array has an index of 0 (the second 1, and so on). The ArrayElement actor is similar to the LookupTable actor in function; however, the two actors specify arrays and index numbers in different ways. The ArrayElement actor receives an array via its input port and an index number via a parameter. The LookupTable actor specifies an array with its table parameter and receives an index number via its input port.

Parameters

index	The index number of an element in the array. The index must be an integer. If the index is out of range (i.e., is greater than or equal to the length of the array), the actor will output nothing. The index can be specified via either the index port or parameter.

Ports

input	An input port that accepts an array from which a value is "looked up" and output.
output	An output port that broadcasts the array element identified by the index number.

Array Extract (ptolemy.actor.lib.ArrayExtract)

Author: null **Version:** Unknown	The ArrayExtract actor reads an array and extracts and outputs a specified subarray. The actor begins reading and outputting array elements at the position specified with the sourcePosition parameter or port. The length of the output subarray is specified with the extractLength parameter or port. Subarray elements are assigned a destinationPosition and outputArrayLength. If the outputArrayLength is greater than the extractLength, any entries not supplied by the input will have the value zero (of the same type as the entries in the input array). By default, only the first element of the input array is copied to the output array, which has length one.

Parameters

sourcePosition	The array index at which to start copying. The index is a non-negative integer that defaults to 0 (i.e., the first array element), and must be less than the length of the input array.
extractLength	The length of the source array segment to copy to the output. The length is a non-negative integer that defaults to 1. The sum of the length and the sourcePosition must be less than or equal to the length of the input array.
destinationPosition	The output array index at which to start copying. The index is a non-negative integer that defaults to 0, and must be less than the length of the output array.
outputArrayLength	The total length of the output array. The length is a non-negative integer that defaults to 1 and must be at least equal to the destinationPosition plus extractLength.

Ports

input	An input port that accepts a source array (e.g., {1,4,11,16})
output	An output port that broadcasts the extracted subarray.

Array Length (ptolemy.actor.lib.ArrayLength)

Author: Edward A. Lee
Version: Unknown

The ArrayLength actor reads an array and outputs the length of the array. The array {0,1,2,3,4} has a length of 5, for example.

Ports

output	An output port that broadcasts the length of the input array.
input	An input port that accepts an array of elements of any type.

Array Level Crossing (ptolemy.actor.lib. ArrayLevelCrossing)

Author: null
Version: Unknown

The ArrayLevelCrossing actor reads an array and outputs the index of the first element that is above or below a specified threshold. If no element meets the specified criteria, the actor outputs -1. The actor begins searching the source array at the index specified via the start parameter or port. By default, the actor searches forward from the start position. Deselect the forwards parameter to search backwards. The threshold can be absolute or relative to the value at the starting index. If the threshold is relative, it can be specified on a linear scale or in decibels. If the threshold is relative and the actor is searching for values above the threshold, then values that exceed the value of the starting index by more than the threshold are reported. If the threshold is relative and the actor is searching for values below the threshold, then values that are below the value of the starting index by more than the threshold are reported.

Parameters

forwards	The direction in which to search. The default is forwards. Deselect the parameter to search backwards from the start position.
threshold	The threshold for which to search. The threshold is a double that can be interpreted on an absolute or relative scale depending on the value of the scale parameter. If relative, the threshold can be on either a linear or decibel scale. The default threshold is 0.0.
above	Specifies whether array values should be above or below the specified threshold. The default is below. Uncheck to search for values above the threshold.
scale	Specifies whether the threshold should be interpreted as absolute or relative. If relative, the scale can be linear, in amplitude decibels, or in power decibels. The default value is "absolute". Note: If decibels are used, then the corresponding linear threshold is 10 to the power of (threshold/N), where N is 20 (for amplitude decibels) or 10 (for power decibels).
start	The index from which to start looking for a threshold crossing. The default is the integer 0.

Ports

array	An input port that accepts an array of doubles.

output	An output port that broadcasts the index of the first array element to meet the specified criteria.

Array Maximum (ptolemy.actor.lib.ArrayMaximum)

Author: Mark Oliver **Version:** Unknown	The ArrayMaximum actor reads an array of elements (e.g., {3,12,33,2,4}) and outputs the value and the index of the largest element via its output and index ports, respectively. If the array contains more than one element with the maximum value, the actor outputs the index of the first occurrence of the value. The first element in an array has an index of 0, the second element has an index of 1, the third has an index of 2, etc.

Ports

output	An output port that broadcasts the value of the largest array element.
input	An input port that accepts an array of elements (e.g., {1,2,3,4,5}).
index	An output port that broadcasts the index of the largest array element. For example, if the input array is {1,2,3,4,5}, the port will output 4, which is the index of the largest value in the array. Note that the first element in an array has an index of 0.

Array Minimum (ptolemy.actor.lib.ArrayMinimum)

Author: Mark Oliver and Edward A. Lee **Version:** Unknown	The ArrayMinimum actor reads an array of elements (e.g., {3,12,33,2,4}) and outputs the value and the index of the smallest (i.e., closest to minus infinity) element via its output and index ports, respectively. If the array contains more than one element with the minimum value, the actor outputs the index of the first occurrence of the value. Note: The first element in an array has an index of 0, the second element has an index of 1, the third has an index of 2, etc.

Ports

output	An output port that broadcasts the value of the smallest array element.
input	An input port that accepts an array of elements (e.g., {1,2,3,4,5}).
index	An output port that broadcasts the index of the smallest array element. For example, if the input array is {3,2,1,4,5}, the port will output 2, which is the index of the smallest value in the array. Note that the first element in an array has an index of 0, the second element has an index of 1, etc.

Array Peak Search (ptolemy.actor.lib.ArrayPeakSearch)

Author: null
Version: Unknown

The ArrayPeakSearch actor reads an array and outputs the indices and values of peak values. The actor reads a source array via its input port. The actor will search all array elements that fall between the index specified with the startIndex and the index specified with the endIndex port or parameter. The dip and squelch parameters control the output's sensitivity to noise. Dip specifies the amount that the signal must differ from a local maximum before a peak is detected. Squelch specifies the value below which the input is ignored by the actor. The parameters are specified either as absolute numbers or as relative numbers. If the dip and squelch parameters are absolute numbers, the actor detects a peak if the array elements rise above the specified dip before the peak, or fall below the specified dip after the peak. If the dip and squelch parameters are relative numbers, then the actor detects a peak when the array elements rise by the specified factor relative to a minimum (before the peak), or dip by the given factor relative to a peak (after the peak). Note that dip is relative to the most recently seen peak or valley, and squelch is relative to the global peak in the array. Relative values can be specified as linear (a fraction) or in decibels. Select either absolute or a flavor of relative from the drop-down menu beside the scale parameter. For example, if the dip parameter is 2.0, and the scale parameter has the value "relative linear", then a dip must drop to half of a local peak value to be considered a dip. If the squelch parameter is 10.0 and scale has the value "relative linear", then any peaks that lie below 1/10 of the global peak are ignored. If scale has the value "relative amplitude decibels", then a value of 6.0 is equivalent to the linear value 2.0. If scale has the value "relative power decibels", then a value of 3.0 is equivalent to the linear value 2.0. In either decibel scale, 0.0 is equivalent to 0.0 linear.

Parameters

dip	The amount that the signal must differ from a local maximum before a peak is detected. The default is 0.0. Use the scale parameter to specify whether the dip is interpreted as an absolute threshold or relative to a local peak.
squelch	The value below which the input is ignored by the actor. The default is -10.0. Use the scale parameter to specify whether the squelch value is interpreted as an absolute number or a number relative to the global peak.
scale	Specifies whether the dip and squelch should be interpreted as absolute or relative. If relative, the scale can be linear, in amplitude decibels, or in power decibels. The default value is "absolute". Note: If decibels are used, then the corresponding linear threshold is 10 to the power of (threshold/N), where N is 20 (for amplitude decibels) or 10 (for power decibels).
maximumNumberOfPeaks	The maximum number of peaks to report. Specify an integer. The default is MaxInt.
startIndex	The starting point of the search. If the index is larger than the value of endIndex, the search is conducted backwards (and the results presented in reverse order). If the index is larger than the length of the input array, then the search is started at the end of the input array. The default is the integer 0.
endIndex	The end point of the search. If the index is larger than the length of the input array, then the search is to the end of the array.

Ports

input	An input port that accepts an array of doubles.
peakValues	An output port that broadcasts the values of the peaks. The type is an array of doubles.
peakIndices	An output port that broadcasts the indices of the peaks. The type is an array of integers.

Array Plotter (ptolemy.actor.lib.gui.ArrayPlotter)

Author: Edward A. Lee
Version: Unknown

The ArrayPlotter reads arrays of doubles via its input multiport and plots each received array as a separate data set. The actor displays a plot graph on the screen. By default, the actor updates the display each time it iterates. Note: updating the display can be costly in terms of system resources. Specify the number of actor iterations that should pass between display updates with the iterationsPerUpdate parameter. For example, if iterationsPerUpdate = 2, then every second time the actor fires, it will update the display (i.e., the actor will update its display on the first firing, the third, the fifth, etc.) The actor will consume its inputs on every firing, regardless of the value of the iterationsPerUpdate parameter.

Parameters

xInit	The starting point of the X-axis.
fillOnWrapup	Specify whether to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
startingDataset	The number of the dataset that should be plotted first. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the plot graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).
iterationsPerUpdate	The number of actor iterations that should pass between display updates. The value must be a non-negative integer. The default value is 1.
xUnit	The increment along the X-axis.

Ports

input	A multiport that receives arrays of doubles.

Array Remove Element (org.geon.ArrayRemoveElement)

Author: null Sudholt **Version:** Unknown	The ArrayRemoveElement actor reads an array and removes occurrences of a specified element. The actor outputs an array consisting of the remaining array elements. The actor removes the element specified by the element parameter or port. Note that the actor will remove all instance of the element unless the array contains no other values (in which case, the actor will leave a single occurrence of the element).

Parameters

element	The element to be removed from the array. The value can be specified with either the element port or parameter.

Ports

array	An input port that accepts an array.
output	An output port that broadcasts the resulting array.

Array Sort **(ptolemy.actor.lib.ArraySort)**

Author: unknown **Version:** Unknown	The ArraySort actor reads an array of values and outputs them in either ascending or descending order (e.g., from A to Z or Z to A). The input array must contain strings (i.e., alphanumeric characters) or non-complex scalar values (e.g., integers or real numbers). Any other type of input will generate an error. The actor can be set to remove duplicate entries.

Parameters

ascending	The sort order. By default, the actor sorts the array in ascending order. Uncheck the parameter to sort in descending order.
allowDuplicates	Optionally remove duplicate array elements. By default, the actor will remove duplicates.

Ports

output	An output port that broadcasts an array of values sorted by ascending or descending order.
input	An input port that accepts an array of strings or non-complex scalar values.

Array To Elements (ptolemy.actor.lib.ArrayToElements)

Author: unknown
Version: Unknown

The ArrayToElements actor reads and disassembles an array. The actor outputs the disassembled values via a multiport output. The actor can accept an array with any element type (e.g., int, double, etc.). Each time the actor fires, it reads and disassembles one array and outputs the individual values. The actor's output port is a multiport, meaning that it can broadcast multiple channels of data. If the width of the output port is less than the number of array elements, then the extra output tokens are discarded (i.e., only the first elements in the array are output via the available channels). The actor is similar to the ArrayToSequence actor, except that it outputs individual values on multiple channels of the output port. The ArrayToSequence actor outputs values as a sequence via a single output port.

Ports

output	An output port that broadcasts the disassembled array values on multiple channels.
input	An input port that accepts an array consisting of elements of any type.

Array To Sequence (ptolemy.domains.sdf.lib.ArrayToSequence)

Author: unknown
Version: Unknown

The ArrayToSequence actor outputs the individual values of an array as a sequence of tokens. The actor can accept an array with any element type (e.g., int, double, etc.). Each time the actor fires, it reads and disassembles one array and outputs a sequence of values. The arrayLength parameter can be used to specify the length of the source array. If the enforceArrayLength parameter is selected, then the actor will only accept arrays of the specified length. The enforceArrayLength parameter should be selected if using an SDF director. If using other directors, such as DE or PN, the enforceArrayLength parameter can be deselected, in which case the arrayLength parameter is ignored. The ArrayToSequence actor is similar to the ArrayToElements actor, except that it outputs values as a sequence via a single output port. The ArrayToElements actor outputs individual values on multiple channels of the output port. Use the ArrayAccumulator actor to convert an array into a single string, where each value is separated by a specified separator.

Parameters

arrayLength	The size of the input array. The default is 1.
enforceArrayLength	Specify whether or not to enforce the arrayLength parameter. By default, the arrayLength is enforced, and the actor will only accept arrays of the specified length.

Ports

output	An output port that broadcasts a sequence of the disassembled array values.
input	An input port that accepts an array of elements of any type.

AscGrid Value (org.ecoinformatics.seek.gis.java_gis.AscGridValue)

Author: Dan Higgins **Version:** Unknown	The AscGridValue actor interpolates a parameter value (e.g., elevation or rainfall) for a specified geospatial point. The x and y coordinates (longitude and latitude) are input as well as a data file in ASC Grid spatial raster format (*.asc). The actor outputs the interpolated value. The actor uses the interpolation algorithm specified in its parameters (either Nearest Neighbor or Inverse Distance Weighted) to calculate the value at the indicated point. If the input point is outside of the data set's geographical extent, or if the point corresponds to "missing data" points, the AscGridValue actor will output a 'nil' value.

Parameters

useDisk	Select this parameter to use disk memory for storing grid data. This option in much slower than using only RAM memory (the default) but allows for very large rasters.
algorithm	The interpolation algorithm to use for calculating the value (either 'Nearest Neighbor' or 'Inverse Distance Weighted')

Ports

xLocation	The longitude value of the geospatial point (e.g., -100.0). If the value is outside of the extent of the data set, or if the point corresponds to "missing data" points, the AscGridValue actor will output a 'nil' value.
yLocation	The latitude value of the geospatial point (e.g., 50.0). If the value is outside of the extent of the data set, or if the point corresponds to "missing data" points, the AscGridValue actor will output a 'nil' value.
input	The file name of an ASC Grid spatial raster (*.asc).
output	The interpolated parameter value for the specified point.

Authentication (org.kepler.authentication.test.AuthActor)

Author: zhije guan **Version:** Unknown	The Authentication actor prompts the user to log in to a Grid Account Management Architecture (GAMA) server. The actor outputs the user's credentials as a string that can be used by other workflow actors. GAMA is a system for securely creating and managing Grid accounts. For more information about GAMA, see http://grid-devel.sdsc.edu/gama.

Ports

output An output port that broadcasts the user's credential as a string.

Average (ptolemy.actor.lib.Average)

Author: Edward A. Lee, Jie Liu **Version:** Unknown	The Average actor outputs the average of the values it receives via its input port. If the actor receives a "true" token on its reset port, then it will reset to 0, and the averaging will begin anew. The actor outputs the current average each time it fires. The inputs and outputs can be any token type that supports addition and division by an integer. The output type is constrained to be the same as the input type. Note that if the input is an integer, then the output is an integer, which may not be expected. Set the input and output port to double to force the result to be a double. The actor will automatically resolve the input type. If the type does not support addition and division, the actor will generate an error.

Ports

output	An output port that broadcasts the current average of the input values.
input	An input port that accepts values of any type that supports addition and division by an integer.
reset	An input port that accepts Boolean tokens. If this port receives a true token, then the actor's state will be reset to 0.

Babel (org.resurgence.moml.Babel)

Author: unknown
Version: Unknown

The Babel actor is a composite actor that uses Babel software to convert chemical modeling data from one file format to another. The actor reads a chemical structure via its input port and outputs the structure in a specified new format (e.g., SMILES, or MDL MOL). To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. Input can be passed to the actor by the MoleculeSelector actor, which is used to select molecular structures from a local file system. Double-click the actor to customize the internal parameters (e.g., the output file type and the output file directory). Babel is an application designed to convert file formats used in molecular modeling and computational chemistry. For more information about Babel, see http://smog.com/chem/babel/.

Parameters

BabelBinary	The directory in which the Babel application resides (e.g., /usr/local/bin/babel)
OutputFileDirectory	The local directory into which to place the output files (e.g., /tmp/openbabel)
AdditionalOptions	Additional options to perform (e.g., -t to denote that all input files describe a single molecule). For a list of options, see http://openbabel.sourceforge.net/wiki/Tutorial:Basic_Usage
OutputFileType	The desired output format (e.g., gamin or box). Select a format from the drop-down menu.

Ports

inputFormatHandle	An input port that accepts the path of a file to be converted.
outputFormatHandle	An output port that broadcasts the path of the converted file.

Bar Graph (ptolemy.actor.lib.gui.BarGraph)

Author: Edward A. Lee
Version: Unknown

The BarGraph actor reads arrays of doubles and displays a bar graph of the input on the screen. By default, the actor updates the screen display each time it iterates. The actor does not save the bar graph. To create and save a simple bar plot, use the BarPlot actor. Note: updating the display can be costly in terms of system resources. Specify the number of actor iterations that should pass between display updates with the iterationsPerUpdate parameter. For example, if iterationsPerUpdate = 2, then the actor will update the display every other time it fires instead of every time. The actor will consume its inputs on every firing, regardless of the value of the iterationsPerUpdate parameter.

Parameters

xInit	The start point of the X-axis.
fillOnWrapup	Specify whether to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
startingDataset	The initial dataset number. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the bar graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).
iterationsPerUpdate	The number of actor iterations that should pass between display updates. The value must be a non-negative integer. The default value is 1.
xUnit	The increment along the X-axis.

Ports

input	A multiport that receives arrays of doubles.

Barplot (org.ecoinformatics.seek.R.RExpression)

Author: Dan Higgins **Version:** Unknown	The Barplot actor creates and saves a simple barplot graph. The actor outputs the path to the barplot graph and (optionally) display the graph itself. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor accepts a single array of values and an optional array of corresponding names. To graph multiple arrays, use the BarGraph actor. The actor creates the bar plot and saves it to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
Values	An input port that accepts an array of values to plot.
Names	An input port that accepts an optional array of names that correspond to the input values.

BasicUnits (ptolemy.data.unit.UnitSystem)

Author: Xiaojun Liu
Version: Unknown

BasicUnits defines a unit system that consists of a set of base and derived units. The base units in the BasicUnits system are meters and seconds. The system defines centimeters, millimeters, and inches in terms of meters; milliseconds are defined in terms of seconds. To view or edit the defined units, right-click the BasicUnits icon and select "Configure Attribute" from the drop-down menu.

Bernoulli (ptolemy.actor.lib.Bernoulli)

Author: Edward A. Lee **Version:** Unknown	The Bernoulli actor generates and outputs a sequence of random Boolean values. The generated values are independent and identically distributed. The probability of true is given by the parameter trueProbability. The seed parameter controls the random number generation. Note that if the parameters are changed during workflow execution, there is a one iteration delay before the changes take effect.

Parameters

seed	The seed that controls the random number generation. A seed of zero (the default) means that the seed is derived from the current system time and a Java hash code (i.e., System.currentTimeMillis() + hashCode()). With extremely high probability, the default seed will ensure that two distinct actors will have distinct seeds. However, current time may not have enough resolution to ensure that two subsequent executions of the same model have distinct seeds. The parameter contains a long token, initially with value 0.
resetOnEachRun	Select to reset the random number generator each time the workflow is run. By default, the generator does not reset.
trueProbability	The probability of true. The default is 0.5.

Ports

output	An output port that broadcasts a sequence of random Boolean values.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Binary File Reader **(org.geon.BinaryFileReader)**

Author: efrat jaeger **Version:** Unknown	The BinaryFileReader reads a local file path or URL and outputs an array of bytes. The actor can read both binary and ASCII file formats. Kepler contains several actors used to read and output files in different ways. To read and output a file as a single string, use the FileReader or SimpleFileReader actor. To read and output a file line by line as a series of strings, use the LineReader actor. To read and output a specified segment of a file as a string, use the SegmentFileReader.

Parameters

fileOrURL	The file name or URL of the file to be read. See FileParameter for more information about specifying file names.

Ports

output	An output port that broadcasts an array of bytes representing the contents of the read file.
endOfFile	An output port that indicates whether or not the end of the file has been reached. If the end of the file has been reached, the port will produce a true value. Otherwise, the value is false.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
fileOrURLPort	An input port that accepts the file name or URL of a file to be read. When the port is connected, the actor reads the file sent by the previous workflow step. The file name or URL can also be specified using the fileOrURL parameter.

Binary File Writer (org.geon.BinaryFileWriter)

Author: Efrat Jaeger **Version:** Unknown	The BinaryFileWriter actor receives an array of bytes and writes it to a file. The actor outputs the file path of the generated file. The BinaryFileWriter will write to the file specified with the fileName parameter. The written data can be appended to the specified file (if it exists), or an existing file can be overwritten--with or without confirmation--depending on the settings of the append and confirmOverwrite parameters.

Parameters

append	Specify whether to append the generated file to the existing, specified file. By default, the actor will overwrite any preexisting file.
confirmOverwrite	Specify whether the actor should request confirmation before overwriting a file. By default, the actor will not ask for confirmation.
fileName	The name of the file to which to write. Type in a file name or use the Browse button to select a destination file. See FileParameter for more information about specifying file names.

Ports

input	A multiport that receives an array of bytes.
filePath	An output port that broadcasts the file name of the generated file.

Boolean Accumulator (org.resurgence.actor.BooleanAccumulator)

Author: Wibke Sudholt
Version: Unknown

The BooleanAccumulator actor reads a sequence of Boolean values and outputs one Boolean value from their combination. The actor uses the logical operator ("and" or "or") specified with the logicalOperator parameter to combine the input values. If the "and" operation is selected, the output will be false if at least one input is false. If the "or" operation is selected, the output will be true if at least one input is true.

Parameters

Logical operator	Specify either "and" or "or". The actor uses the logical operator when combining the input values. If the "and" operation is selected, the output will be false if at least one input is false. If the "or" operation is selected, the output will be true if at least one input is true.
number	The number of Boolean values in the input sequence. The default is 1. The number may be specified via the number port or parameter.

Ports

combination	An output port that broadcasts a Boolean value based on the combination of the input values.
number	An input port that accepts the number of Boolean values in the input sequence. The default is 1. The number may also be specified with the number parameter.
booleans	An input port that accepts a sequence of Boolean values.

Boolean Multiplexor (ptolemy.actor.lib.BooleanMultiplexor)

Author: Steve Neuendorffer
Version: Unknown

The Boolean Multiplexor uses a Boolean value to determine which of two input values to output. The actor is useful when creating workflow control structures, which allow workflows to branch, for example. The actor reads values of any type via its trueInput and falseInput port. In addition, the actor receives a Boolean "select" token, which it uses to determine which input value to select and output. The actor outputs the selected value each time it iterates. The actor must receive tokens on all ports. Each time the actor fires, it reads one token from each of its input ports and outputs the selected token.

Ports

select	An input port that receives a Boolean token (true or false) that is used to select and output either the trueInput or falseInput value.
falseInput	An input port that receives tokens of any type. The value will be output if the "select" token is false.
output	An output port that broadcasts the selected input value.
trueInput	An input port that receives tokens of any type. The value will be output if the "select" token is true.

Boolean Switch (ptolemy.actor.lib.BooleanSwitch)

Author: Steve Neuendorffer
Version: Unknown

The BooleanSwitch actor reads a value of any type, as well as a Boolean token that is used as a control. If the Boolean token is true, the actor outputs the received value to the trueOutput port; if the Boolean token is false, the actor outputs the received value to the falseOutput port. If the actor has never received a value on the control port, then the actor will output to the falseOutput port. The actor only works under certain directors. It will not work under an SDF Director, but it will under a PN Director, for example.

Ports

input	An input port that accepts tokens of any type.
falseOutput	An output port that broadcasts the input token when the control is false.
trueOutput	An output port that broadcasts the input token when the control is true.
control	An input port that accepts a Boolean token used to select which output port (trueOutput or falseOutput) to broadcast.

Boolean To Anything (ptolemy.actor.lib.conversions. BooleanToAnything)

Author: Wibke Sudholt
Version: Unknown

The BooleanToAnything actor converts a Boolean token (i.e., true or false) into any data type and value. The actor reads a Boolean value and assigns a new value based on the specified conversion values. The actor outputs the value of the trueValue parameter if the Boolean input is true. The actor outputs the value of the falseValue parameter if the Boolean input is false. The type of the output matches that of the specified conversion values.

Parameters

trueValue	The value to output when a "true" input is read.
falseValue	The value to output when a "false" input is read.

Ports

output	An output port that broadcasts the converted data. The actor will output the value of the falseValue parameter if the input value is false, or the value of the trueValue parameter if the input value is true.
input	An input port that receives a Boolean token.

Boxplot (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin **Version:** Unknown	The Boxplot actor creates and saves a boxplot. The actor reads an array of values and, optionally, an array over which the values are divided (an array of dates, for example). The actor outputs the path to the saved boxplot and (optionally) displays the graph. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor creates the boxplot and saves it to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory, by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to allow downstream R-actors to retrieve the workspace later in a workflow.

Ports

graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Variable	An input port that accepts an array of values to plot.
Group	An input port that accepts an array over which the input values are divided (an array of dates, for example). The port is optional.

Browser Display **(org.geon.BrowserDisplay)**

Author: Efrat Jaeger
Version: Unknown

The BrowserDisplay actor reads a file name or URL and displays the file in the user's default browser.

Ports

trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
inputURL	An input port that accepts the file name or URL of a file to be read.

BrowserUI (org.sdm.spa.BrowserUI)

Author: Ilkay Altintas, Efrat Jaeger and Kai Lin
Version: Unknown

The BrowserUI actor displays text/HTML in the system's default browser, allowing users to interact with the content during workflow execution. The actor outputs arrays of user-selected "(name, value)" pairs in XML format. The actor can also be used to simply display a file in a Web browser, which is useful for displaying the output of legacy applications. Select the useForDisplay parameter to use the BrowserUI actor as a browser viewer with no output port. The actor accepts either a URL/file path or a string of HTML/text. If content is input as a string, select a file type (e.g., html or xml) with the fileExtensionParameter. The BrowserUI actor is often used in conjunction with the SRBCreateQueryConditions and SRBCreateQueryInterface actors to create a set of user-specified query conditions.

Parameters

file extension	The extension of the input content: .html, .xml, .txt, .xsl, or .svg. Specify this parameter when content is input via the fileContent input port.
use for display	Select this setting to use the actor solely for display. The output port will be turned off. By default, the setting is off.
portConfiguration	The output port configuration. The actor will create the described ports. Each port should be represented on a separate line as portName portType (e.g., height int).
hasTrigger	Specify whether to activate a trigger port or not. The default is off.
fileOrURL	The file name or URL of a file to be read. The value can also be specified via the fileOrURL port.

Ports

connected	This port is used only for scheduling the actor. Activate the hasTrigger parameter to display the trigger port.
trigger	The port to trigger the actor in case there are no other input ports. By default, this port is hidden.
xmlOutput	An output port that broadcasts an array of user-selected xml "(name, value)" pairs.
fileOrURL	An input port that accepts the file name or URL of a file to be read. The value can also be specified with the fileOrURL parameter.

fileContent	An input port that accepts a string of content to display. Select a file extension that matches the content (e.g., .html,.xml or .svg) with the file extension parameter.

CGSUnitBase (ptolemy.data.unit.UnitSystem)

Author: Xiaojun Liu
Version: Unknown

CGSUnitBase ("Centimeter, Gram, Second") defines a unit system that consists of a set of base and derived units. To view or edit the defined units, right-click the CGSUnitBase icon and select Configure Attribute from the drop-down menu.

CV Hull to Raster (org.ecoinformatics.seek.gis.java_gis.CVHull2Raster)

Author: Dan higgins **Version:** Unknown	The CVHullToRaster actor reads a list of (x,y) points representing a convex hull polygon. The actor creates a "mask": points within the convex hull are set to a value of 1 and points outside the hull have a value of "NO_DATA." The actor saves the mask and outputs it as a raster file. The hullFileName port accepts a a space-delimited text file containing one pair of (x,y) convex hull points per line. The ConvexHull actor can be used to generate this input file. Use the rasterFileName input port to specify a name for the output raster file. The size and resolution of the generated raster file can be specified with the numRows, numCols, xllCorner, yllCorner, and cellSize parameters.

Parameters

useDisk	A Boolean setting to determine whether or not to use a disk for storing data rather than putting all data in RAM arrays during processing. This option in much slower but allows for very large raster files.
xllcorner	A double token representing the x-value of the lower-left corner of the raster. If the parameter is empty, the x-value is set to the minimum x in the convex hull.
numrows	An integer representing the number of rows for the raster.
numcols	An integer representing the number of columns for the raster.
cellsize	A double token representing the cell size used by the raster (assumed square). If the parameter is empty, the actor automatically generates a raster with 50 x-direction cells, and y-direction cells to match the y-extent of the convex hull.
yllcorner	A double token representing the y-value of the lower-left corner of the raster. If the parameter is empty, the y-value is set to the minimum y in the convex hull.

Ports

rasterFileResult	An output port that broadcasts the path of the generated raster file, which the actor creates and saves in *.asc format.
rasterFileName	An input port that accepts a name to be given to the output raster file.
hullFileName	An input port accepting a space-delimited text file containing (x,y) convex hull data points (one pair of points per line).

Cartesian To Complex (ptolemy.actor.lib.conversions.CartesianToComplex)

Author: Michael Leung, Jie Liu, Edward A. Lee, Paul Whitaker
Version: Unknown

The CartesianToComplex actor converts a Cartesian pair (x,y) to a single complex token (e.g., 2 + 3i), which it outputs. Each time the actor fires, it consumes exactly one token from each of its two input ports (x and y) and outputs a complex token. The x-input becomes the real part of the complex output and the y-input becomes the imaginary part. If either input port is empty, the actor outputs nothing.

Ports

x	An input port that accepts the x-coordinate of the Cartesian pair. The type is double.
output	An output port that broadcasts the complex token.
y	An input port that accepts the y-coordinate of the Cartesian pair. The type is double.

Cartesian To Polar (ptolemy.actor.lib.conversions. CartesianToPolar)

Author: Michael Leung, Edward Lee, Paul Whitaker
Version: Unknown

The CartesianToPolar actor converts a Cartesian pair (x and y) to polar coordinates (magnitude and angle). The angle is returned in radians.

Ports

x	An input port that accepts the x-coordinate of the input pair, which has type double.
magnitude	An output port that broadcasts the magnitude component of the polar value, which has type double.
angle	An output port that broadcasts the angle component of the polar value, which has type double.
y	An input port that accepts the y-coordinate of the input pair, which has type double.

ClimateChangeFileProcessor (util. ClimateChangeFileProcessor)

Author: dan higgins
Version: Unknown

The ClimateChangeFileProcessor actor converts climate change data from the Intergovernmental Panel on Climate Change (IPCC) into a more generalized raster format that can be read by other actors. The actor outputs the name of the converted file. The ClimateChangeFileProcessor actor is very similar to the ClimateFileProcessor actor, except that it is designed to work with predicted climate change datasets, which differ from historical (1961-1990) IPCC datasets in formatting. For more information about IPCC, see http://www.ipcc.ch/. The ClimateChangeFileProcessor actor receives the name of an IPCC climate change data file via its input port and saves the reformatted file in the same directory. Specify another location using the baseOutputFileName parameter. Customize the type of climate change data to output (minimum, maximum, or average climate change values) with the outputType parameter. Specify the output period for the data (the season or year) with the outputPeriod parameter.

Parameters

nodatavalueParameter	The value used to indicate missing data.
outputPeriod	The time period to output: annual, fall, winter, summer, or spring.
colsParameter	The number of columns in the data file. The number of columns is specified in the IPCC data source metadata. To view the metadata, right-click the IPCC dataset and select Get Metadata.
rowsParameter	The number of rows in the data file. The number of rows is specified in the IPCC data source metadata. To view the metadata, right-click the IPCC dataset and select Get Metadata.
baseOutputFileName	The base file path for the raster file, which the actor creates and saves. If no path is specified, the file is placed in the same directory as the source IPCC file. In either case, the actor appends text indicating the type and time period of the data to the file name.
outputType	The type of climate change data to output: minimum, maximum, or average climate change values. The selected type will be output for the selected time period, which is specified with the outputPeriod parameter.

Ports

output	An output port that broadcasts the file name of the raster file created by the actor. The actor creates this file from the IPCC source data, based on the time and type values specified by the user.
input	An input port that accepts the file name of the source IPCC climate change dataset. IPCC data can be located via the EcoGrid.

ClimateFileProcessor (util.ClimateFileProcessor)

Author: dan higgins
Version: Unknown

The ClimateFileProcessor actor converts historical climate data from the Intergovernmental Panel on Climate Change (IPCC) into a more generalized raster format that can be read by other actors. The actor outputs the name of the converted file. The ClimateFileProcessor actor is very similar to the ClimateChangeFileProcessor actor, except that it is designed to work with historical climate datasets rather than climate change datasets, which have different formats. For more information about IPCC, see http://www.ipcc.ch/. The ClimateFileProcessor actor receives the name of an IPCC climate data file via its input port and saves the reformatted file in the same directory. Specify another location using the baseOutputFileName parameter. Customize the type of climate data to output (minimum, maximum, or average climate values) with the outputType parameter. Specify the output period for the data (the season or year) with the outputPeriod parameter.

Parameters

nodatavalueParameter	The value used to indicate missing data.
outputPeriod	The time period to output. Select annual, fall, winter, summer, or spring.
colsParameter	The number of columns in the data file. The number of columns is specified in the IPCC data source metadata. To view the metadata, right-click the IPCC dataset and select Get Metadata.
rowsParameter	The number of rows in the data file. The number of rows is specified in the IPCC data source metadata. To view the metadata, right-click the IPCC dataset and select Get Metadata.
baseOutputFileName	The base file path for the generated raster file, which the actor creates and saves. If no path is specified, the file is placed in the same directory as the source IPCC file. In either case, the actor appends text indicating the type and time period of the data to the file name.
outputType	The type of climate data to output: minimum, maximum, or average climate values. The selected type will be output for the selected time period, which is specified with the outputPeriod parameter.

Ports

output	An output port that broadcasts the file name of the raster file created by the actor. The actor creates this file from the IPCC source data, based on the time and type values specified by the user.
input	An input port that accepts the file name of the source IPCC climate dataset. IPCC data can be located via the EcoGrid.

Close Database Connection (org.geon.CloseDBConnection)

Author: efrat jaeger **Version:** Unknown	The CloseDatabaseConnection actor disconnects a database connection established by the OpenDatabaseConnection actor. A reference to the database connection is passed to the CloseDatabaseConnection actor via the dbcon input port.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
dbcon	An input port that accepts a reference to an established database connection. The OpenDatabaseConnection actor can be used to generate this reference.

ColorParameter (ptolemy.actor.gui.ColorAttribute)

Author: Edward A. Lee **Version:** Unknown	The ColorParameter specifies a color, which can be selected from the Choose menu that is available when the parameter is double-clicked. Each color is represented by an array of the form {red, green, blue, alpha}. Each value in the array is a double that falls within the range of 0.0 to 1.0 (a blue color might be represented as {0.0,0.0,1.0,1.0), for example). The "alpha" value represents opacity, where 1.0 is opaque, and 0.0 is fully transparent (invisible).

arguments	An input port that accepts an argument or arguments (e.g., -r or –al), which will be passed to the command. If the infileHandle parameter is used, this port can be empty.
inputStream	An input port that accepts strings to pass to the command (e.g., a file name)
infileHandle	An optional input port that accepts the path to an input file. Used an input file if the command accepts one instead of a list of arguments.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
outfileHandle	An output port that broadcasts the path to the output file, if specified. An output file replaces the standard output.
exitCode	An output port that indicates whether or not the command executed successfully. The exit code will be 1 if the command is success.
output	An output port that broadcasts the result stream of the command. The port broadcasts only if no outputFile is specified.

Comparator (ptolemy.actor.lib.logic.Comparator)

Author: Edward A. Lee
Version: Unknown

The Comparator reads two values and compares them. The actor outputs a Boolean value (true or false) that indicates whether the comparison criteria were met or not. Specify the type of comparison to perform with the comparison parameter. The following comparisons may be performed: ComparisonDefinition >left is greater than right; the default comparison. >=left is greater than or equal to right less thanleft is less than right less than or equalleft is less than or equal to right ==left equals right The input ports are named "left" and "right" to indicate which side of the comparison operator their value appears on. The tolerance parameter, which defaults to zero, defines an error tolerance. If the tolerance is greater than zero, the actor will output true when the specified comparison is not exactly satisfied, but rather is almost satisfied within the specified tolerance.

Parameters

tolerance	The tolerance for the comparison. If the tolerance is greater than zero, the actor will output true when the comparison is satisfied within the specified tolerance (i.e., the comparison need not be exactly satisfied). The value is a double that defaults to 0.0.
comparison	The comparison to perform. The default is greater than.

Ports

output	An output port that broadcasts a Boolean value (true or false) that indicates whether the comparison criteria were met or not.
right	An input port that accepts a double(or any data type that can be losslessly converted to a double token) that will be used on the right side of the comparison operator.
left	An input port that accepts a double (or any data type that can be losslessly converted to a double token) that will be used on the left side of the comparison operator.

Complex To Cartesian (ptolemy.actor.lib.conversions. ComplexToCartesian)

Author: Michael Leung, Edward Lee, Paul Whitaker
Version: Unknown

The ComplexToCartesian actor converts a single complex token (e.g., 2 + 3i) into a Cartesian pair (x,y), which it outputs on its two output ports (x and y).

Ports

x	An output port that broadcasts the x-coordinate (i.e., the real part of the complex input), which has type double.
input	An input port that accepts a complex token.
y	An output port that broadcasts the y-coordinate (i.e., the imaginary coordinate), which has type double.

Complex To Polar (ptolemy.actor.lib.conversions. ComplexToPolar)

Author: Michael Leung
Version: Unknown

The ComplexToPolar actor converts a single complex token (e.g., 2 + 3i) into polar coordinates (magnitude and angle). The angle is returned in radians.

Ports

input	An input port that accepts a complex token.
magnitude	An output port that broadcasts the magnitude component, which has type double.
angle	An output port that broadcasts the angle component, which has type double. The angle is returned in radians.

CompositeActor (ptolemy.actor.TypedCompositeActor)

Author: Yuhong Xiong
Version: Unknown

A Composite actor is an aggregation of actors. It may have a local director that is responsible for executing the contained actors. A Composite actor with a local director is called an opaque actor. Composite actors do not require a local director. Composite actors with no local director "inherit" the director from the containing workflow and are called non-opaque. To create a composite actor, drag and drop the Composite actor onto the Workflow canvas. Right-click the actor and select Open Actor from the drop-down menu. A new Kepler application window will open for designing the composite.

Concatenate Arrays (util.ConcatenateArrays)

Author: Dan Higgins
Version: 1.0 Sept 21, 2007

The concatenateArrays actor reads individual arrays via its input port and outputs an array consisting of a concatenation of those arrays. The actor accepts input elements of arrays of ints, doubles, strings, etc.. Each time the actor fires, it reads one token from each channel of the input port, concatenates the contents of these arrays, and outputs a resulting array.

Ports

output	An output port that broadcasts a concatenation of the input arrays.
input	A multiport that accepts arrays

Constant (ptolemy.actor.lib.Const)

Author: null **Version:** Unknown	The Constant actor outputs a constant, which is specified by the value parameter. By default, the actor outputs the integer 1. The actor can be used to output other types of values, e.g., a string (such as "Hello") or a double (such as 1.2). The actor' s output type matches the type of the specified value. NOTE: If using a PN Director, the 'firingCountLimit' parameter is often set to a finite integer (e.g. '1') so that the workflow will terminate.

Parameters

firingCountLimit	The limit on the number of times the actor will fire. The default value is 'NONE', meaning there is no limit on the number of time the constant will be provided to the output port. Any integer can be provided as a value for this parameter.
value	The value produced by the Constant actor. By default, the value is the integer token 1. The value can be set to another type, e.g., a string (such as "Hello") or a double (such as 1.2). The output type matches the type of the value specified here.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
output	An output port that broadcasts the specified constant. By default, the output is 1.

Convert Image To String (ptolemy.actor.lib.image. ImageToString)

Author: Christopher Hylands **Version:** Unknown	The ConvertImageToString actor reads an image token and outputs a string containing information about the image (e.g., the image dimensions). Use the ImageReader or the ConvertURLToImage actor to read the URL or path of an image and convert it to an image token that the ConvertImageToString actor can use.

Ports

output	An output port that broadcasts a string containing information about the image.
input	An input port that accepts an image token.

Convert URL To Image (ptolemy.actor.lib.image.URLToImage)

Author: Christopher Hylands **Version:** Unknown	The ConvertURLToImage actor reads the URL or path of an image and outputs the image in a form that can be used by other actors (an image token). The actor accepts a string representing the URL or file path of an image via the input port. Specify local files by using the prefix "file://" instead of "http://". To specify a file relative to the current directory, use "../" or "./". For example, if the current directory contains a file called "test.jpg", then the input should be set to "file:./test.jpg". If the parent directory contains a file called "test.jpg", then the input should be set to "file:../test.jpg". To reference the file test.jpg, located at "/tmp/test.jpg", the input should be set to file:///tmp/test.jpg. The ConvertURLToImage actor is similar in function to the ImageReader, except that the ConvertURLToImage actor receives and converts a URL via an input port, while the ImageReader receives a URL via a parameter.

Ports

output	An output port that broadcasts an image token.
input	An input port that accepts a string representing the URL or file path of an image.

Convex Hull (org.ecoinformatics.seek.gis.java_gis. GISHullActorJ)

Author: Dan higgins **Version:** Unknown	The ConvexHull actor reads a set of (x,y) points, calculates the points that define the convex hull (i.e., the smallest polygon that contains the given points), and saves the hull points to a file. The actor outputs the number of points in the convex hull, as well as the name of the file that contains the calculated hull points. The pointFileName port accepts a set of (x,y) points that are input as a tab-delimited text file. Use the hullFileName input port to specify a name for the generated output file. The actor uses the value of the scaleFactor parameter to linearly scale the output hull shape from its center. To scale by area (2x the area, 3x the area, etc), set the scaleFactor to the square root of the desired scaling factor. To make a shape with twice the area, set the scale factor to SQRT(2), for example. Note: if the scaleFactor parameter is empty or not a number, no scaling will be done. The actor outputs the number of (x,y) pairs contained in the generated hull file via the numHullPoint point, and the name of the convex hull file via the hullFileResult port. Note that the value of the hullFileName input port and the hullFileResult port are the same (the output is used as a trigger).

Parameters

scaleFactorParameter	The actor uses the value of the scaleFactor parameter to linearly scale the output hull shape from its center. To scale by area (2x the area, 3x the area, etc), set the scaleFactor to the square root of the desired scaling factor. For example, to make a shape with twice the area, set the scale factor to SQRT(2).

Ports

hullFileResult	An output port that broadcasts the hull file name (the output is used as a trigger). The value is the same as that of the hullFileName input.
hullFileName	An input port that accepts a name to be given to the output convex hull point list file.
pointFileName	An input port that accepts a tab-delimited text file containing (x,y) input points.
numHullPoint	An output port that broadcasts the number of (x,y) pairs in the hull file.

Correlation (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin **Version:** Unknown	The Correlation actor uses R to perform parametric and non-parametric tests of association between two input variables (e.g., two arrays of equal length). The actor outputs the level of association (r, rho, or tau, depending on the analysis) between the two variables, an estimate of the p-value (if possible), and n. The actor uses R to perform a variance analysis and outputs a graphical representation of its calculations. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. By default, the actor performs a Pearson's correlation analysis; to specify another analysis type, connect a Constant actor to the actor's method port and enter another type of analysis (e.g., "spearmen" or "kendall"). The actor performs the analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
displayGraphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

association	An output port that broadcasts the calculated level of variable association.
graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.

output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Variable2	An input port that accepts the second variable. The variable is specified as an array of values. The array length must match that of the array specified for Variable1.
Method	The method used to perform the correlation analysis. The default method is Pearson. To change the method, connect a Constant actor and enter another type of analysis (e.g., "spearman" or "kendall").
p_value	An output that broadcasts the p-value.
n	An output port that broadcasts the number of data pairs in the input.
Variable1	An input port that accepts the first variable. The variable is specified as an array of values. The array length must match that of the array specified for Variable2.

Counter (ptolemy.actor.lib.Counter)

Author: Steve Neuendorffer
Version: Unknown

The Counter actor increments or decrements an internal counter, depending on its inputs. The actor outputs the counter value as an integer. Each time the actor fires, it adjusts its internal counter (incrementing or decrementing the count) based on the input. Whenever a token is received from the increment input, the internal counter is incremented. Whenever a token is received from the decrement port, the internal counter is decremented. Whenever a token is received from either input port, a token is created on the output port with the integer value of the current count. At most one token will be consumed from each input during each firing. If a token is present on both input ports, then the increment and the decrement will cancel out (i.e., the count will remain unchanged), and only one output token will be produced.

Ports

increment	An input port used to increment the counter. The port accepts tokens of any type.
output	An output port that broadcasts the counter value.
decrement	An input port used to decrement the counter. The port accepts tokens of any type.

Current Time (ptolemy.actor.lib.CurrentTime)

Author: jie liu and edward a lee **Version:** Unknown	The CurrentTime actor outputs the current time in the workflow. The actor outputs a double token that represents "model time," which may differ from "real time." For example, "model time" may advance 30 seconds while only a second of "real time" passes. Note that time is kept by the workflow director, and that not every director has a notion of time. For example, with an SDF Director, the current time will likely be 0.0; by default, time does not advance in SDF workflows.

Parameters

stopTime	The time at which the actor will stop producing output. By default, stopTime is set to infinity, meaning that the actor will never stop (or at least, it will not stop because stopTime has been exceeded).

Ports

output	An output port that broadcasts a double token representing the current model time.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

DDF Boolean Select (ptolemy.domains.ddf.lib.DDFBooleanSelect)

Author: Gang Zhou
Version: Unknown

A type polymorphic select with boolean valued control for use in the DDF domain. In the first iteration, an input token at the control port is read and its value is noted. In the second iteration, if the control input read from the previous iteration is true, then an input token at the trueInput port is read and sent to the output. Likewise with a false control input and the falseInput port. It alternates between these two kinds of iterations until stopped. The control port must receive Boolean Tokens. The trueInput and falseInput ports may receive Tokens of any type. Because tokens are immutable, the same Token is sent to the output, rather than a copy. Note this actor sends an output token every two iterations. Contrast this with BooleanSelect which sends an output token every iteration.

Ports

trueInput	Input for tokens on the true path. The port type can be any type.
falseInput	Input for tokens on the false path. The port type can be any type.
output	The output port. The type is at least the type of trueInput and falseInput.
control	Input that selects one of the other input ports. The type is boolean.

DarwinCoreDataSource (org.ecoinformatics.seek.datasource.darwincore.DarwinCoreDataSource)

Author: Spears, Higgins **Version:** 1	The DarwinCoreDataSource actor uses the DiGIR (Distributed Generic Information Retrieval) protocol to search for and access occurrence data for species contained in museum collections. The DarwinCoreDataSource actor provides query access to DiGIR data and handles the mechanical issues involved in searching the collection, downloading the results, and emitting the data to downstream actors. By default, the ports created by the DarwinCoreDataSource actor provide a complete data table as a string with additional ports specifying the number of attributes and the attribute delimiter used in the data table. When As Field is specified for the outputType, the actor automatically reconfigures its exposed ports to provide one port for each attribute contained in the result table (after receiving initial search results from the DiGIR service). NOTE: Retrieving data from DiGIR collections may take several minutes per species the first time the workflow is run. Subsequent workflow executions are considerably faster because the data is cached locally during the first execution.

Parameters

outputType	Specify which ports are created for the actor and what data is emitted on those ports during each fire cycle. For example, this field can be configured to produce one port for each column in the data table, or one port that emits the entire data table at once. Specifically, the output format choices are: As Field The actor creates one output port for each field (i.e., column/attribute/variable) that is contained in the DiGIR results. The type of each port (e.g., string, int, double, etc.) matches the base type of the field. Note: a search term should be specified in the searchData in order for the field ports to be created. As Table (the default) The selected data will be output as a string that contains the entire search result data. The actor creates three output ports: DataTable - the data itself, Delimiter - delimiter to separate fields, and numColumns - the number of fields in the table. As Row One tuple of selected data is formatted as an array and output. The actor creates one output port (DataRow), and the data type is a record containing each of the individual data fields.
searchData	Search DiGIR using the entered search term (if not using the speciesName input port)

endPoint	The DiGIR query service endpoint

Ports

speciesName	Allows search term[s] to be dynamically input as part of the workflow rather than specificed as a static parameter for this actor
trigger	Allows other actors to signal that the the search be performed. This should be used in cases when the speciesName input port is not being used.
output	Output ports are automatically configured depending on the selected outputType. See the documentation on outputType for more information

Data Group **(org.resurgence.moml.DataGroup)**

Author: unknown **Version:** Unknown	The DataGroup actor is a composite actor used inside the GamessInputGenerator actor. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

GroupName	Open the actor to configure the parameter.
CalculationTitle	Open the actor to configure the parameter.
SymmetryGroup	Open the actor to configure the parameter.
HighestAxisOrder	Open the actor to configure the parameter.
SymmetryGenerators	Open the actor to configure the parameter.

Ports

inputListIn	An input port that accepts a list of inputs
gamessAtomData	An input port that accepts GAMESS atom data.
inputListOut	An output port that broadcasts the generated results.

Database Query (org.geon.DatabaseQuery)

Author: efrat jaeger **Version:** Unknown	The DatabaseQuery actor performs database queries against an open database and outputs the query results in a specified format. Use the OpenDatabaseConnection actor to establish a database connection and generate a reference to that connection. The reference is passed to the DatabaseQuery actor via the dbcon port. A query is passed to the actor via the query port or parameter. Specify whether to output all results at once, or one row at a time using the outputEachRowSeparately parameter. The outputType parameter specifies the format in which to return results: XML, record, array, string, no metadata, or result set.

Parameters

outputEachRowSeparately	Specify whether to display the complete result at once or each row separately.
schemaDef	The schema definition contains the field names of data types.
query	An input query string. Queries can be specified via the query port or query parameter.
outputType	The output format: XML, record, array, string, no metadata, or result set.

Ports

result	An output port that broadcasts the query result. Results will be output in the format specified with the outputType parameter: XML, record, array, string, no metadata (i.e., a relational string with no metadata), or result set.
dbcon	An input port that accepts a reference to an established database connection. The OpenDatabaseConnection actor can be used to generate this reference.

Database Writer (org.sdm.spa.DatabaseWriter)

Author: Yang Zhao, Daniel Crawl
Version: Unknown

The DatabaseWriter actor performs database updates against an open database and outputs the number of rows inserted. Use the OpenDatabaseConnection actor to establish a database connection and generate a reference to that connection. The reference is passed to the DatabaseWriter actor via the dbcon port. An update is passed to the actor via the update port or parameter.

Parameters

query	An input update string. Updates can be specified via the query port or query parameter.

Ports

result	An output port that broadcasts how many rows were sucessfully updated.
dbcon	An input port that accepts a reference to an established database connection. The OpenDatabaseConnection actor can be used to generate this reference.

Differential Equation (ptolemy.actor.lib.Differential)

Author: Edward Lee
Version: Unknown

The DifferentialEquation actor reads differential equations, subtracts the current equation from the previously received one, and outputs the difference, or the current equation if no previous input has been received.

Ports

output	An output port that broadcasts the difference between the current input and a previous input, or the current input if the actor has received no previous input.
input	An input port that accepts differential equations.

Directory Listing (ptolemy.actor.lib.io.DirectoryListing)

Author: christopher hylands, edward lee
Version: Unknown

The DirectoryListing actor reads a local or remote directory name, and outputs an array of file and/or folder names contained by that directory. The returned file names are absolute. Returned file and folder names may be "filtered" using the optional pattern parameter. Specify a pattern that file and folder names must match in order to be included in the output. The pattern is a regular expression. For example, *.htm matches all .html files, but not .jpg or .pdf files. For a reference on regular expression syntax, see http://java.sun.com/docs/books/tutorial/extra/regex/index.html. Specify the name of the directory with the directoryOrURL parameter. If directoryOrURL is a local directory, then the listOnlyDirectories and the listOnlyFiles parameters can be used to refine the returned listings. Select listOnlyDirectories to output only directory names. Select listOnlyFiles to output only file names. If both parameters are selected, then the actor generates an error. If directoryOrURL is a remote directory, then the actor will list the full contents of the directory. Note that the contents of the directory are returned in a different order depending on whether one is using Sun JVM or the IBM JVM. Connect the output to an ArraySort actor to ensure consistency.

Parameters

directoryOrURL	The name of a local or remote directory. The actor will return the names of the files and/or folders contained by this directory.
listOnlyFiles	Select to output only file names. This parameter is only relevant if the directoryOrURL parameter refers to a local directory. If directoryOrURL is a remote URL, then this parameter is ignored.
listOnlyDirectories	Select to output only directory names. This parameter is only relevant if the directoryOrURL parameter refers to a local directory. If directoryOrURL is a remote URL, then this parameter is ignored.
firingCountLimit	The number of iterations that transpire before the actor indicates that it is finished. If firingCountLimit is set to zero, the actor has no limit imposed.
pattern	A regular expression that must match file and directory names. The default value is the empty String "", which indicates that everything matches. For a reference on regular expression syntax, see http://java.sun.com/docs/books/tutorial/extra/regex/index.html.

Ports

directoryOrURL	An input port that accepts the directory name or URL from which to read.
output	An output port that broadcasts an array of file and/or folder names.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Directory Maker (org.resurgence.actor.DirectoryMaker)

Author: Wibke Sudholt **Version:** Unknown	The DirectoryMaker actor creates a new local directory and outputs its path. The name of the new directory is specified with the directoryName parameter.

Parameters

Directory name	The name and path of the new directory. See FileParameter for more information about specifying paths.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
path	An output port that broadcasts the new directory path.

Discrete Random Number Generator (ptolemy.actor.lib.DiscreteRandomSource)

Author: Jeff Tsay, Yuhong Xiong **Version:** Unknown	The DiscreteRandomNumberGenerator actor reads an array of values and selects a value at random based on specified probabilities. The actor outputs the selected value. The actor will select and output one of the values specified by the values parameter. This parameter accepts an array of elements of any type. The default is an array of integers {0,1}. The actor selects random output values based on the probability mass function (pmf) specified with the pmf parameter. This parameter accepts an array of values between 0.0 and 1 that sum to 1. The default is {0.5, 0.5}. The length of the pmf array must be the same as the length of the values array. Each element of the pmf array specifies the probability of the corresponding element in the values array.

Parameters

seed	The seed that controls the random number generation. A seed of zero (the default) means that the seed is derived from the current system time and a Java hash code (i.e., System.currentTimeMillis() + hashCode()). With extremely high probability, the default seed will ensure that two distinct actors will have distinct seeds. However, current time may not have enough resolution to ensure that two subsequent executions of the same model have distinct seeds. The parameter contains a long token, initially with value 0.
values	An array of values that can be sent to the output. The actor will select and output one value based on the value of the pmf parameter. The initial value is an integer array,{0, 1}.
pmf	The probability mass function (pmf). This parameter accepts an array of values between 0.0 and 1 that sum to 1. The default is {0.5, 0.5}. The length of the pmf array must be the same as the length of the values array. Each element of the pmf array specifies the probability of the corresponding element in the values array.
resetOnEachRun	Select to reset the random number generator each time the workflow is run. By default, the generator does not reset.

Ports

output	An output port that broadcasts the selected random value.

trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Display (ptolemy.actor.lib.gui.Display)

Author: Yuhong Xiong, Edward A. Lee **Version:** Unknown	The Display actor reads tokens of any type via its input multiport, and displays each token on a separate line in a text display window. Specify the size of the text display window with the rowsDisplayed and columnsDisplayed parameters. Simply resizing the window onscreen does not persistently change the size when the workflow is saved, closed, and then re-opened. If the input is a string token, then the actor strips the surrounding quotation marks before displaying the value. Select the suppressBlankLines parameter to specify that the actor not add blank lines to the display. By default, the actor will add blank lines. Note: this actor can consume large amounts of memory. It is not advisable to use it to display large output streams.

Parameters

suppressBlankLines	Specify whether the actor should display blank lines (the default) or suppress them.
rowsDisplayed	The vertical size of the display, in rows. The value is an integer that defaults to 10.
columnsDisplayed	The horizontal size of the display, in columns. The value is an integer that defaults to 40.
title	The title of the text display window. If specified, the value will appear in the title bar of the text display window.

Ports

input	A multiport that accepts tokens of any type.

Distributor (ptolemy.actor.lib.Distributor)

Author: Mudit Goel, Edward A. Lee
Version: null

A polymorphic distributor, which splits an input stream into a set of output streams. The distributor has an input port and an output port, the latter of which is a multiport. The types of the ports are undeclared and will be resolved by the type resolution mechanism, with the constraint that the output type must be greater than or equal to the input type. On each call to the fire method, the actor reads at most N tokens from the input, where N is the width of the output port times the blockSize parameter, and writes blockSize tokens to each output channel, in the order of the channels. If there are fewer than N tokens at the input, then the all available input tokens are sent to the output channels, and the fire () method returns. In the next iteration of this actor, it will begin producing outputs on the first channel that did not have enough tokens in the previous iteration. For the benefit of domains like SDF, which need to know the token consumption or production rate for all ports before they can construct a firing schedule, this actor sets the tokenConsumptionRate parameter for the input port to equal the number of output channels times the blockSize parameter, and the output production rate is set to the blockSize parameter. The consumption rate parameter is set each time that a link is established with the input port, or when a link is removed. The director is notified that the schedule is invalid, so that if the link is modified at run time, the schedule will be recalculated if necessary.

Parameters

blockSize	The number of tokens produced on each output channel on each firing. This is an integer with default value 1.

Ports

input
output

DocViewer (ptolemy.kernel.util.SingletonAttribute)

Author: Steve Neuendorffer and Edward A. Lee
Version: Unknown

The DocViewer attribute renders a clickable documentation icon on the Workflow canvas. Clicking the icon opens the workflow documentation window.

Documentation (ptolemy.vergil.kernel.attributes. DocumentationAttribute)

Dot Product (ptolemy.domains.sdf.lib.DotProduct)

Author: Jeff Tsay, Paul Whitaker, Adam Cataldo
Version: Unknown

The DotProduct actor reads either two arrays or two matrices of equal length and computes and outputs their dot product. The elements of the input arrays or matrices can be of any scalar type (integer, double, complex, etc.)

Ports

input2	An input port that accepts an array or matrix containing elements of any scalar type. The length of the specified array or matrix must equal the length of the one specified via the input1 port.
input1	An input port that accepts an array or matrix containing elements of any scalar type. The length of the specified array or matrix must equal the length of the one specified via the input2 port.
output	An output port that broadcasts the dot product of the inputs.

EML 2 Dataset (org.ecoinformatics.seek.datasource.eml.eml2.Eml200DataSource)

Author: matt jones, jing tao, chad
Version: Unknown

The EML2Dataset actor provides access to a wide variety of data packages that have been described using EML (Ecological Metadata Language). The actor accesses an EML dataset and handles the mechanical issues involved in parsing metadata, downloading the dataset (if applicable), and emitting data to downstream actors. Each data package contains an EML metadata description and one or more data entities (e.g., data tables, spatial raster images, spatial vector images). The EML metadata allows the described data to be easily ingested into Kepler and exposed for use in downstream components. The data packages can be accessed from the local file system or through any EcoGrid server that provides access to its collection of data objects. The supported data transfer protocols include http, ftp, file, ecogrid, and srb. After parsing the EML metadata, the actor automatically reconfigures its exposed ports to provide one port for each attribute described by the first entity of the EML description. For example, if the first entity is a data table with four columns, the ports might be "Site", "Date", "Plot", and "Rainfall." These details are obtained from the EML document. To preview the data, right-click the actor icon and select Preview from the drop-down menu. By default, the ports created by the EML2Dataset actor represent data fields, and one tuple of data (e.g., one database row) is emitted on these ports each time the actor fires. Alternatively, the actor can be configured so that the ports represent an array of field values ("AsColumnVector"), or so that the ports represent an entire table of data ("AsTable"). Data tables are formatted in comma-separated-value (CSV) format. If more than one data entity is described in the EML metadata, then the output of the actor defaults to the first entity listed in the EML. To select the other entities, specify an entity with the Selected parameter, or use the Query Builder to describe the filter and join that should be used to produce the data output. To use the Query Builder, right-click the actor and select "Open Actor." Specify the fields to be output and any filtering constraints to be applied.

Parameters

fileExtensionFilter	A file extension used to limit the array of file names returned by the actor when the selected output type is "As UnCompressed File Name". This parameter is ignored for other output types.
dataOutputFormat	Specify which ports are created for the actor and what data is emitted on those ports during each fire cycle. For example, this field can be configured to produce one port for each column in a data table, or one port that emits the entire data table at once in CSV format. Specifically, the output format choices are: As Field (the default) The actor creates one output port for each field (i.e., column/attribute/variable) that is described in the EML metadata for the data package. The type of each port (e.g., string, int, double, etc.) matches the base type of the field. If a query statement has been used to subset the data, then only those fields selected in the query statement will be configured as ports. As Table The selected data will be output as a string that contains the entire entity data. The actor creates three output ports: DataTable - the data itself, Delimiter - delimiter to separate fields, and NumColumns - the number of fields in the table. As Row One tuple of selected data is formatted as an array and output. The actor creates one output port (DataRow), and the data type is a record containing each of the individual data fields. As Byte Array Selected data will be output as an array of bytes. The actor creates two output ports: BinaryData - contains the raw data itself, and EndOfStream - indicates whether the end of data stream has been reached. As UnCompressed File Name This format is only used when the data package is a compressed file (zip, tar, et al). The compressed archive file is uncompressed after it is downloaded. The actor creates one output port, which contains an array of the filenames of all of the uncompressed archive files. If a FileExtensionFilter is specified, then the array will only contain files that match the specified extension. As Cache File Name Kepler stores data files downloaded from remote sites into its cache system. This output format will send the local cache file path for the data package so that workflow designers can directly access the cache files. The actor creates two output ports: CacheLocalFileName (the local file path) and CacheResourceName (the EML data link. e.g., ecogrid://knb/tao.2.1). As Column Vector This output format is similar to "As Field", except instead of sending out a single value on each port, the actor sends out an array of all of the data for each field. The type of each port is an array of the base type for the field. As ColumnBased Record This output format sends all data on one port using a record structure that encapsulates the entire data entity. The record will contain one array for each data field, and the type of each array will be determined by the

	type of the field it represents.
selectedEntity	If this EML data package has multiple entities, the selectedEntity parameter specifies which entity should be output. When this parameter is unset (the default), data from the first entity described in an EML package is output. This parameter is only used if no query statement is specified, or if a query statement is used and the output format is one of "As Table", "As Byte Array", "As Uncompressed File Name", and "As Cache File Name". To specify a query statement, right-click the actor and select Open Actor.
emlFilePath	The file path of a local EML metadata file used to describe and access an EML data set.
dataFilePath	The path to a local data file described by EML (must be used in conjunction with a local EML file). The actor will retrieve the data and automatically configure its ports to output it.
isLenient	If this parameter is selected, "extra" columns of data (e.g., comments that people have entered on a line or something of that nature) that are not described in the metadata are ignored, allowing the workflow to execute. If the option is unchecked (the default), the workflow execution will halt until the discrepancy between the data and metadata is corrected.
checkVersion	Select this parameter to check the EarthGrid for updates to the data. If the actor finds a version of the data that is more recent than the cached data on your local system, the actor will prompt the user to either download the latest data and metadata or ignore the newer version. Note that different versions of the data can have vastly different structures (new columns, or even new tables of data might be included or removed). If this parameter is selected, users should be prepared to handle changes that might arise from differences in the data structure.

Ports

output	Output ports are automatically configured to provide one port for each attribute in the first entity described in the EML description. For example, if the first entity is a data table with four columns, the ports might be "Site", "Date", "Plot", and "Rainfall." These details are obtained from the EML document.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

ENM PCP (org.kepler.gis.display.ENMPCPVisualizer)

Author: Jianting Zhang **Version:** Unknown	The ENMPCP actor invokes a Parallel Coordinate Plot (PCP) window to display GARP presampling results. Known problem: c.f. ENMPCPFrame.java GARP is a genetic algorithm that creates an ecological niche model that represents the environmental conditions where a species would be able to maintain populations. GARP is an acronym for Genetic Algorithm for Rule Set Production, a program originally created by David Stockwell. For more information about GARP, see http://www.lifemapper.org/desktopgarp/.

Ports

SampleStringPort	A string that consists of GARP presampling results, which can be passed to the actor by the GARPPresampleLayers actor.

ESRI Shape File Displayer (org.kepler.gis.display. JumpSHPDisplayer)

Author: Jianting Zhang **Version:** Unknown	The ESRIShapeFileDisplayer reads the name of a local ESRI shape file and displays the file on the screen. ESRI shape files contain a set of vector coordinates that represent non-topological geographic data. For more information about ESRI shape files, see http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

Ports

SHPFileNamePort	The file path of the local ESRI shape file to read. See FileParameter for more information about specifying file names.

Ecogrid Writer (org.ecoinformatics.seek.ecogrid.EcogridWriter)

Author: tao
Version: Unknown

The EcogridWriter actor writes a data file and the EML metadata describing that data file to a remote EcoGrid repositiory. Identifiers for the data and metadata are created and these IDs are sent to the output ports. These IDs might be used for future access to the data and metadata files. Note also that the ID of the data file is inserted into the metadata file as a reference (i.e., a pointer from the metadata to the data). The EcoGrid is a distributed network providing scientists access to ecological, biodiversity, and environmental data and analytic resources. The EcoGrid can be used to store ecological data, or to model or analyze it via remote EcoGrid services. Ecological Metadata Language (EML) is a standard set of terms and definitions used to describe ecological data. For example, EML metadata might contain infomation about a data set's units of measurement, date of collection, location, etc.

Parameters

authenticationURL	The URL of the EcoGrid service for authenticating the user (e.g., http://ecogrid.ecoinformatics.org/knb/services/AuthenticationService)
passWord	Authentication password.
metadataDestination	The URL of the EcoGrid service for receiving metadata and data (e.g., http://ecogrid.ecoinformatics.org/knb/services/PutService)
userName	Authentication username.

Ports

metadataDocid	An output port that broadcasts the metadata doc ID, which is generated by the actor for future reference.
metadata	An input port that accepts a string of metadata describing the data file.
dataDocid	An output port that broadcasts the data file ID, which is generated by the actor for future reference.
dataFileNamePort	An input port that accepts the file name and path of the local data file to write to the EcoGrid service.

ElectronicUnitBase (ptolemy.data.unit.UnitSystem)

Author: Xiaojun Liu
Version: Unknown

ElectronicUnitBase defines a unit system that consists of a set of base and derived units. To view or edit the defined units, right-click the ElectronicUnitBase icon and select Configure Attribute from the drop-down menu.

Elements To Array (ptolemy.actor.lib.ElementsToArray)

Author: null **Version:** Unknown	The ElementsToArray actor reads individual elements via its input port and outputs an array consisting of those elements. The actor accepts input elements of any one type (int, double, etc.). The type must be consistent. Each time the actor fires, it reads one token from each channel of the input port and outputs a corresponding array.

Ports

output	An output port that broadcasts an assembled array. The type of the array elements matches that of the input.
input	A multiport that accepts tokens of any one type. The type must be consistent.

Ellipse (ptolemy.vergil.kernel.attributes.EllipseAttribute)

Author: Edward A. Lee
Version: Unknown

The Ellipse attribute renders an ellipse that is centered on its origin (the default). Single-click the ellipse and drag the resize handles to adjust its size, or double-click the ellipse to customize the height, width, line width, and color.

Parameters

height	The height of the ellipse. The value is a double that defaults to 100.0.
lineWidth	The line width. The value is a double that defaults to 1.0.
centered	Indicate whether the shape should be centered on its origin. By default, the ellipse is centered.
dashArray	Specify a dash-pattern for dashed or dotted lines. The value consists of an array of doubles that specify the length of the alternating solid and transparent segments. An empty value indicates that the line should not be dashed (the default).
width	The width of the ellipse. The value is a double that defaults to 100.0.
lineColor	The line color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. The default is an opaque black, {0.0, 0.0, 0.0, 1.0}
fillColor	The fill color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. By default, the value is "none."

Email Sender (org.sdm.spa.Email)

Author: Ilkay Altintas **Version:** Unknown	The EmailSender actor sends email notifications from a workflow to a specified address. Email notifications are especially handy for managing remotely executed long-running workflows. Specify a "to" and "from" address and an SMPT host via the actor's parameters. The actor will send a message with the subject and body provided by its input port.

Parameters

subject	The message subject (e.g., Notification email from Kepler). The subject can be specified via the subject port or parameter.
host	The SMTP host of the message sender (e.g., smtp.yourisp.com).
fromAddress	Email address of the message sender.
toAddress	Email address of the message recipient.

Ports

subject	A multiport that accepts the message subject (e.g., Notification email from Kepler). The subject can be specified via the subject port or parameter.
messageBody	A multiport that accepts the message that Kepler will send to the specified email address.

End GAMESS Input (org.resurgence.moml.EndGamessInput)

Author: unknown **Version:** Unknown	The EndGamessInput actor is a composite actor used inside the GamessInputGenerator actor, which is used for computational chemistry workflows. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Ports

inputList	An input port.
inputFileHandle	An input port.
gamessInputHandle	An output port that broadcasts parameterized GAMESS input files.

Equals (ptolemy.actor.lib.logic.Equals)

Author: John Li and Edward A. Lee
Version: Unknown

The Equals actor reads one token of any type from each channel of its input multiport and compares the values to see if they are equal. If all input tokens are equal, the actor outputs the Boolean value true; if the tokens are not all equal, the actor outputs false. The actor automatically resolves the input type; however, if all input channels cannot resolve to the same type (which would be the case if one input were a matrix and another an array, for example), the actor will generate an error. If the actor does not receive an input token on at least one input channel, it will produce no output.

Ports

output	An output port that broadcasts a Boolean token: true if all input tokens are equal, or false if they are not.
input	A multiport that receives tokens of any type. The actor automatically resolves the input type; however, if all input channels cannot resolve to the same type (which would be the case if one input were a matrix and another an array, for example), the actor will generate an error.

Experiment Monitor (org.resurgence.moml.ExperimentMonitor)

Author: null **Version:** Unknown	The ExperimentMonitor actor is a composite actor used inside the GAMESSNimrodRun composite actor, which is used in computational chemistry workflows. To see inside the actor, right-click it and select "Open Actor" from the menu. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed. Nimrod is a tool for scheduling and managing parametric experiments on the Grid, allowing scientists to easily and efficiently run a computational model and vary key object parameters (e.g., length, speed, etc). For more information about Nimrod, see http://www.csse.monash.edu.au/~davida/nimrod/ GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ExperimentName	The name of the experiment (e.g., gamess)
NimrodBinary	The location of the Nimrod application (e.g., /opt/nimrodg-3.0.0/bin/nimrod)

Ports

exitCode	An output port that broadcasts the exit status of the operation (e.g., "success" or a generated error).
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Experiment Preparator (org.resurgence.moml. ExperimentPreparator)

Author: null **Version:** Unknown	The ExperimentPreparator actor is a composite actor used inside the GAMESSNimrodRun composite actor, which is used in computational chemistry workflows. To see inside the actor, right-click it and select "Open Actor" from the menu. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed. Nimrod is a tool for scheduling and managing parametric experiments on the Grid, allowing scientists to easily and efficiently run a computational model and vary key object parameters (e.g., length, speed, etc). For more information about Nimrod, see http://www.csse.monash.edu.au/~davida/nimrod/ GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ExperimentName	The name of the experiment (e.g., gamess)
ExperimentType	The type of the experiment (e.g., G, O, or A)
NimrodBinary	The location of the Nimrod application (e.g., /opt/nimrodg-3.0.0/bin/nimrod)

Ports

output	An output port that broadcasts the result of the actor's execution.
planFileHandle	An input port that receives a file handle, which can be generated and output by the GAMESSPlanFileGenerator actor.

Experiment Starter (org.resurgence.moml. ExperimentStarter)

Author: null **Version:** Unknown	The ExperimentStartor actor is a composite actor used inside the GAMESSNimrodRun composite actor, which is used in computational chemistry workflows. To see inside the actor, right-click it and select "Open Actor" from the menu. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed. Nimrod is a tool for scheduling and managing parametric experiments on the Grid, allowing scientists to easily and efficiently run a computational model and vary key object parameters (e.g., length, speed, etc). For more information about Nimrod, see http://www.csse.monash.edu.au/~davida/nimrod/ GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ExperimentName	The name of the experiment (e.g., gamess)
NimrodBinary	The location of the Nimrod application (e.g., /opt/nimrodg-3.0.0/bin/nimrod)

Ports

exitCode	An output port that broadcasts the exit status of the operation (e.g., "success" or a generated error).
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Expression (ptolemy.actor.lib.Expression)

Author: Xiaojun Liu, Edward A. Lee, Steve Neuendorffer
Version: Unknown

The Expression actor evaluates a specified expression (e.g., an addition or multiplication operation), which may reference the values of user-specified input ports, the current time, or the actor's iteration count. The actor outputs the value of the evaluated expression. Expressions are specified in the Ptolemy expression language via the expression parameter. For more information about the expression language, see http://ptolemy.eecs.berkeley.edu/papers/05/ptIIdesign1-intro/ptIIdesign1-intro.pdf. By default, the expression parameter is empty, and attempting to execute the actor without first specifying an expression generates an error. Expressions can refer to the values of inputs by the port name; to the current time by the identifier "time"; and to the current iteration count by the identifier "iteration." Input ports are created by the user and correspond to variables used in the specified expression. Currently, the Expression actor does not support input multiports. The actor requires all of its inputs to be present. If inputs are not all present, then the actor will generate an error. Note: the Expression actor can be used instead of many of the arithmetic actors, such as AddSubtract, MultiplyDivide, and TrigFunction. However, those actors will be usually be more efficient, and sometimes more convenient to use.

Parameters

expression	An expression to evaluate. Expressions are specified in the Ptolemy expression language. For more information about the expression language, see http://ptolemy.eecs.berkeley.edu/papers/05/ptIIdesign1-intro/ptIIdesign1-intro.pdf. By default, the parameter is empty, and attempting to execute the actor without first specifying an expression generates an error. Expressions can refer to the values of inputs by the port name; to the current time by the identifier "time"; and to the current iteration count by the identifier "iteration."

Ports

output	An output port that broadcasts the value of the evaluated expression. The actor automatically determines the type based on the type of the input.

Expression Reader (ptolemy.actor.lib.io.ExpressionReader)

Author: edward lee **Version:** Unknown	The ExpressionReader reads a file or URL, one line at a time, and evaluates each line as an expression. One evaluated result is output each time the actor iterates. The first line in the file determines the data type of the output. For example, if the first line is 20.1*4 (which evaluates to 80.4), the type of the output would be "double." All other lines must contain expressions that evaluate to the same type or a subtype, or an error will occur. Note: To output all evaluated values over the course of a single actor iteration, use the FileToArrayConverter actor.

Parameters

numberOfLinesToSkip	The number of lines to skip at the beginning of the file or URL. The value of this parameter must be non-negative and defaults to 0.
fileOrURL	The file name or URL of the file to be read. See FileParameter for more information about specifying file names.

Ports

output	An output port that broadcasts the evaluated result. One result is output each time the actor iterates. The type will match the type of the expression evaluated in the first line of the file.
endOfFile	An output port that indicates whether or not the end of the file has been reached. If the end of the file has been reached, the port will produce a true value. Otherwise, the value is false.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Expression To Token (ptolemy.actor.lib.conversions.ExpressionToToken)

Author: Xiaojun Liu, Edward A. Lee, Steve Neuendorffer
Version: Unknown

The ExpressionToToken actor reads and evaluates an expression (e.g., 2+4) and outputs the result (e.g., 6, in this simple example). The actor accepts an expression that is passed as a string. By default, the output type is "general," meaning that no specific type is assigned. Expressions are specified in the Ptolemy expression language. For more information about the expression language, see http://ptolemy.eecs.berkeley.edu/papers/05/ptIIdesign1-intro/ptIIdesign1-intro.pdf.

Ports

output	An output port that broadcasts the value of the evaluated result.
input	An input port that accepts an expression passed as a string. Expressions are specified in the Ptolemy expression language. For more information about the expression language, see http://ptolemy.eecs.berkeley.edu/papers/05/ptIIdesign1-intro/ptIIdesign1-intro.pdf.

External Execution (ptolemy.actor.lib.Exec)

Author: Xiaojun Liu, Edward A. Lee, Steve Neuendorffer **Version:** Unknown	The ExternalExecution actor executes a system command from a workflow. The actor accepts a command, a directory and environment in which to execute the command, and a command input string. Once the command has finished executing, the actor will output the execution results along with any errors. If no results are generated, the actor will output an empty string. The ExternalExecution actor depends on system-specific executables and is operating-system specific. To mimic the effect of executing a command in a shell interpreter, set the command parameter to "cmd" (Windows) or "sh" (Windows with Cygwin or Linux), and then provide commands, as a string, via the input port. Note that each passed command must be terminated with a newline. For example, to open a workflow in vergil and run it, set command to "sh" and use a StringConstant actor to pass the input port the string: "vergil -run model. xml\n exit\n"

Parameters

command	The command string to execute (e.g., ls or C:/ Program Files/Internet Explorer/IEXPLORE. EXE) and, optionally, one or more arguments. The command can also by input via the actor's command port.
directory	The directory in which to execute the command. The default value of this parameter $CWD, which represents the user's current working or home directory.
environment	An array of records that name an environment variable and a value: {{name = "NAME1", value = "value1"}...} Where NAME1 is the name of the environment variable, and value1 is the value. For example, {{name = "PTII", value = "c:/ptII"}} would set the value of the PTII to c:/ ptII. If the parameter is set to {{name="", value = ""}}, then the environment from the parent process is used. If environmental variables are set with the parameter, the parent values may not be passed to the process. To view the current environment, use the "env" command.

firingCountLimit	Specify a positive integer to limit the maximum number of times the actor is executed.
prependPlatformDependent ShellCommand	If this parameter is selected, the actor will preface the command with a platform-dependent shell command 'cmd.exe \c' (under Windows NT or XP) or 'command.com / C' (under Windows 95) or '/bin/sh –c' (all other platforms). By default, the parameter is not selected. This parameter must be selected if file redirection is used in the command. NOTE: Under Cygwin, if this parameter is selected, the path environment of the subprocess is not identical to the path of the calling process.
throwExceptionOnNon ZeroReturn	If selected, the actor will generate an error message if the invoked subprocess returns an error.
waitForProcess	Select to indicate that the command should finish executing before the actor outputs results. By default, the actor will stream command results as they are generated.

Ports

input	An input port that accepts strings to pass to the standard input of the subprocess. Note that a newline is not appended to the string. If you require a newline, add one using the AddSubtract actor. This port is an input port of type String.
trigger	If connected, a token must be available on this port before actor will execute.
output	An output port that broadcasts data generated by the executed command, output as a string after the command has finished executing. If the command generates no data on standard out, then the empty string (a string of length zero) is generated.
error	An output port that broadcasts any errors generated by the command execution. Errors are output as a string after the command has finished executing. If the execution generates no errors, an empty string is output.

FTP Client (org.geon.FTPClient)

Author: unknown **Version:** Unknown	The FTPClient actor uploads or downloads files from a remote FTP server. FTP (File-Transfer-Protocol) is used to copy files from one computer to another over a network. If the server requires a username and password, these values must be specified in the actor parameters. Upload or download a single file, multiple files, or a directory by passing the desired files as a string via the arguments port. Kepler contains several actors used for uploading and downloading files. Use the GridFTP, FileFetcher and UpdatedGridFTP actors to upload and download files from Globus servers, which can use an authorization certificate. See those actors for more information.

Parameters

remote path	Directory to get files from or put files into.
operation	The operation performed (Put or Get). Currently, the actor can either "put" a local file on a remote server, or "get" a remote file.
password	The authentication password
host	The host server name (e.g., myserver.com)
mode	The transfer mode: asc or bin. Use "asc" (i.e., ASCII) when transferring plain text files. Use "bin" (i.e., Binary) for everything else (MS Word files, images, etc).
username	The authentication username

Ports

url	An output port that broadcasts the URL of the uploaded/downloaded file.
arguments	An input port that accepts a string representing local files to upload, or remote files to download.

File Copier (org.resurgence.actor.FileCopier)

Author: Wibke Sudholt
Version: Unknown

The FileCopier actor copies a file to a specified directory. The actor outputs the file's new path. Specify an existing file to copy via the actor's input port, and choose a new file location with the destination port or parameter. If the specified destination directory does not exist, the actor will create it. If the destination file already exists, the actor can be instructed overwrite it. By default, the actor will not overwrite an existing file. To append the content of a specified file to an existing destination file, use the FileCopy actor. The FileCopier actor can also be used to move a file to the specified destination. To move (rather than copy) a file, select the move parameter.

Parameters

directory	The destination directory name. The directory name can be specified via either the directory port or parameter.
Overwrite existing	Specify whether the actor should overwrite an existing destination file. By default, the actor will not overwrite an existing file.
Move files	Specify whether the actor should move the specified file instead of copying it. By default, the actor will not move the specified file.

Ports

directory	An input port that accepts the destination directory name. The directory name can also be specified with the directory parameter.
newFile	An output port that broadcasts the new file path and name.
oldFile	An input port that accepts an existing file path and name. See FileParameter for more information about specifying paths.

File Copy (org.geon.FileCopy)

Author: Efrat Jaeger **Version:** Unknown	The FileCopy actor copies a specified file to a destination file and outputs the destination file name and path. The source and destination file paths can be accepted through either the sourceFile port or parameter. See FileParameter for more information about specifying paths. Choose to overwrite the destination file or append the copied file to the destination file using the actor's append parameter. By default, the actor will request confirmation before overwriting a file. Note: to move a file instead of copying it, use the FileCopier actor.

Parameters

append	Specify whether to append the source file to the specified destination file or to overwrite it. By default, the actor will overwrite any preexisting file (after asking for permission).
confirmOverwrite	Specify whether the actor should request confirmation before overwriting an existing file. By default, the actor will ask for confirmation.
sourceFileParam	The file name and path of the file to be copied. See FileParameter for more information about specifying paths. The file name and path can also be specified via the sourceFile port.
destFileParam	The name and path of the destination file. See FileParameter for more information about specifying paths. The destination file name can also be specified using the destinationFile parameter.

Ports

sourceFile	An input port that accepts the file name and path of a file to be copied. See FileParameter for more information about specifying paths. The file name and path can also be specified using the sourceFile parameter.
outputFile	An output port that broadcasts the name and path of the copied file.
destinationFile	An input port that accepts the name and path of the destination file. See FileParameter for more information about specifying paths. The destination file name can also be specified using the destinationFile parameter.

File Existence Monitor (org.resurgence.moml. FileExistenceMonitor)

Author: unknown
Version: Unknown

The FileExistenceMonitor actor is a composite actor designed to work with a computational chemistry workflow. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system.

Ports

monitoredFileName	An input port that accepts the input of the composite actor.
existingFileHandle	An output port that broadcasts the output of the composite actor.

File Fetcher (org.sdm.spa.FileFetcher)

Author: ilkay altintas
Version: Unknown

The FileFetcher actor reads a set of files on a remote Globus machine, copies the files into a specified local directory, and outputs the list of file paths as a string when it is done. In order to access the Globus machine, the FileFetcher actor uses a proxy certificate provided by the GlobusProxy actor. A certificate allows the actor to access the Grid without requiring that the user enter a password. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org.

Parameters

SourceHostname	The name of a Globus machine (e.g., griddle.sdsc.edu)
DestinationDirectoryPath	The path from which the specified files should be fetched.

Ports

certificate	An input port that accepts the Globus proxy certificate generated by the GlobusProxy actor.
filesToGet	An input port that accepts the list of files to fetch. File paths should be specified in string form, and each should be separated from the next by semicolons. File paths can be full paths, or can be given relative to the directory specified in the DestinationDirectoryPath parameter.
fetchedFiles	An output port that broadcasts the list of fetched files. File paths are specified in string form and are separated by semicolons.

File List Sequencer (org.resurgence.moml. FileListSequencer)

Author: unknown **Version:** Unknown	The FileListSequencer actor is a composite actor used to select chemical modeling data from a local file system. The actor reads a directory of files, and outputs a sequence of files with a specified extension (e.g., cml). Right-click the actor and select Open Actor from the menu to set the required parameters (the directory and desired file type). The output structures may be passed to an OpenBabel actor, which can convert the files to a new format. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed.

Parameters

DirectoryHandle	The directory to use as a source.
FileType	The type of file to return (e.g., cml).

Ports

trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
fileHandle	An output port that broadcasts the file names of the specified file type.

File Location Chooser (org.resurgence.moml.FileLocationChooser)

Author: unknown **Version:** Unknown	The FileLocationChooser actor is a composite actor that reads a sequence of molecule file names and outputs those files to a new user-specified directory. Users may also append information to the file name and specify a new file extension. Double-click the actor to specify the new file directory as well as a file name addition and file type. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed.

Parameters

NewFileDirectory	The directory in which to place the output files.
FileNameAddition	A string to append to each file name.
InputFileType	The type of the input files.
OutputFileType	The type of the output files.

Ports

originalFileHandle	An input port that accepts a sequence of source file names. This sequence can be generated by the MoleculeSelector actor
inputFileHandle	An output port that broadcasts the file handle of the original files.
outputFileName	An output port that broadcasts the new file names.

File Name Chooser (org.resurgence.moml. FileNameChooser)

Author: unknown **Version:** Unknown	The FileNameChooser actor is a composite actor used inside the GamessInputGenerator actor, which is used in computational chemistry workflows. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/ To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed.

Parameters

OutputFileDirectory	The directory in which to place the output files.
FileNameAddition	A string to append to each file name.
OutputFileType	The type of the output files.

Ports

inputFileHandle	The names of the source files.
inputFileName	An output port that broadcasts the file name of the original files.
outputFileName	An output port that broadcasts the new file names.

FileParameter (ptolemy.data.expr.FileParameter)

Author: yang zhao, edward lee
Version: none

The FileParameter specifies a file or URL. The value of the parameter is a string that may contain references to variables within scope (i.e., variables defined at the same level of the hierarchy or higher) using the $ID ${ID}, or $(ID) syntax. Change the name of the FileParameter to better identify the file it specifies (right-click the parameter and select "Customize Name" from the menu). Other actors may refer to the specified file using the $NAME syntax (e.g., $FileParameter or $SourceFile). File names can be specified in several ways: As a full path, which can be selected via the Browse option. As a path relative to a base, often the path of the parent workflow. Files specified relative to a base can be kept in the same directory as the parent workflow file, and should be moved together with the workflow. Note: workflows must be saved as MOML files before the workflow path can be used as a base. If the workflow has not been saved to a file, then the class path is used for identifying relative file names. A path that contains references to system-defined strings via the $NAME syntax (e.g., $CWD or $HOME). The following strings may be useful (to see the values of these strings, select "Check System Settings" from the Tools menu in the Toolbar): $CWDThe current working directory $HOMEThe user's home directory $TMPDIRThe temporary directory A special file name. The following file names are understood: System.in Standard input System.outStandard output

Parameters

fileParameter The specified file or URL.

File Reader (ptolemy.actor.lib.io.FileReader)

Author: yang zhao, edward lee **Version:** none	The FileReader actor reads a local file or URL and outputs the contents of the file as a single string. The actor is similar to the SimpleFileReader, except that the FileReader has an additional trigger port and can read a URL via either an input port or parameter. Kepler contains several actors used to read and output files in different ways. To output each line of the file as a separate string, use the LineReader actor. To output a specified section of the file, use the SegmentFileReader.

Parameters

newline	The end of line character(s). In general, this setting can be left at its default setting.
fileOrURL	The file name or URL of the file to be read. See FileParameter for more information about specifying file names.

Ports

output	An output port that broadcasts the contents of the file as a single string.
endOfFile	An output port that indicates whether or not the end of the file has been reached. If the end of the file has been reached, the port will produce a true value. Otherwise, the value is false.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
fileOrURL	An optional input port that accepts the name or URL of a file to be read. When the port is connected, the actor reads the file sent by the previous workflow step. The file name or URL can also be specified using the fileOrURL parameter.

File Stager (org.sdm.spa.FileStager)

Author: ilkay altintas
Version: Unknown

The FileStager actor copies a specified set of files to a remote Globus machine, and outputs the list of copied file paths as a string. In order to access the Globus machine, the FileStager actor uses a proxy certificate provided by the GlobusProxy actor. A certificate allows the actor to access the Grid without requiring the user to enter password. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org.

Parameters

DestinationHostname	The name of a Globus machine (e.g., griddle.sdsc.edu)
DestinationDirectoryPath	The location to which the specified files should be copied.

Ports

stagedFiles	The list of copied files. The list consists of a string of file paths separated by semicolons.
certificate	The Globus proxy certificate generated by the GlobusProxy actor.
filesToPut	The list of files to copy to the Globus machine. The list consists of a string of file paths separated by semicolons.

File To Array Converter (org.geon.FileToArray)

Author: efrat jaeger
Version: Unknown

The FileToArrayConverter actor reads a file or URL, evaluates each line, and outputs an array of the evaluated values. The actor is similar to the ExpressionReader actor, except that the FileToArrayConverter outputs all of the evaluated expressions as a single array instead of outputting each value separately. All of the elements in the output array must be of the same type (e.g., integers or doubles, etc.) If the expressions evaluate to different types, an error will occur. Select a local file to read by right-clicking the actor, selecting Configure Actor, and using the Browse option to select a file. Specify a URL by typing directly into the directoryOrURL field. Note that URLs must contain a trailing slash after the last directory name: http://www.myurl.com/directory/ Use the numberOfLinesToSkip parameter to optionally specify the number of lines to skip at the beginning of the file. 0 is specified by default.

Parameters

numberOfLinesToSkip	The number of lines to skip at the beginning of the file or URL. 0 is specified by default. The value of this parameter must be non-negative.
fileOrURL	The file name or URL of the file to be read. See FileParameter for more information about specifying file names.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
output	An output port that broadcasts an array of evaluated expressions read from an input file. The actor outputs an array of evaluated expression each time it iterates. The type of each array element will match the type of the evaluated expression in the first line of the file.

File Writer **(org.geon.FileWrite)**

Author: No author given **Version:** Unknown	The FileWriter actor reads a string and writes it to a file. The actor outputs the file path of the generated file. Specify a destination file path with the fileName parameter. If the specified file does not exist, then the actor will create it. If the file already exists, then the actor will ask for permission to overwrite it (unless the append parameter is selected, in which case the new content is appended to the existing content). The actor is similar to the LineWriter, except that the FileWriter outputs the generated file path, while the LineWriter actor does not. FileWriter is also similar to TextFileWriter, except that the FileWriter actor adds line breaks, while the TextFileWriter does not.

Parameters

append	Specify whether to append the input string to an existing, specified file. By default, the actor will overwrite any preexisting file.
confirmOverwrite	Specify whether the actor should confirm before overwriting an existing file. By default, the actor will not ask for confirmation.
fileName	The name of the file to which to write. See FileParameter for more information about specifying file names.

Ports

input	An input port that receives a string to write to a file.
url	An output port that broadcasts the file name of the generated file.

FilterUI (org.geon.FilterUI)

Author: unknown **Version:** Unknown	The FilterUI actor allows users to interact with a workflow. The actor accepts an array of strings, which it displays in a Web browser. Users can select one or more of the displayed strings via the browser interface. The actor outputs an array of the user-selected strings.

Ports

input	An input port that accepts an array of strings. The actor displays the input in a Web browser, allowing users to select one or more of the input strings.
output	An output port that broadcasts an array of user-selected strings.

Fork Resource Adder (org.resurgence.moml.ForkResourceAdder)

Author: null
Version: Unknown

The ForkResource actor is a composite actor designed for computational chemistry workflows. To see inside the actor, right-click it and select "Open Actor" from the menu. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed. Nimrod is a tool for scheduling and managing parametric experiments on the Grid, allowing scientists to easily and efficiently run a computational model and vary key object parameters (e.g., length, speed, etc). For more information about Nimrod, see http://www.csse.monash.edu.au/~davida/nimrod/ GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ExperimentName	The name of the experiment (e.g., gamess)
NimrodBinary	The location of the Nimrod application (e.g., /opt/nimrodg-3.0.0/bin/nimrod)

Ports

output	An output port that broadcasts the result of the actor's execution.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Formatted Group (org.resurgence.moml.FormattedGroup)

Author: unknown **Version:** Unknown	The FormattedGroup actor is a composite actor designed to work with computational chemistry workflows. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

GroupName	A parameter.

Ports

inputListIn	An input port.
formattedData	An input port.
inputListOut	An output port that broadcasts the results of the actor's execution.

GAMESS Atom Data Extractor (org.resurgence.moml.GamessAtomDataExtractor)

Author: unknown
Version: Unknown

The GamessAtomDataExtractor actor is a composite actor used inside the GamessInputGenerator actor, which is used in computational chemistry workflows. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Ports

gamessDataGroupHandle	An input port that accepts files that are formatted for use in GAMESS software (e.g., gamin files)
gamessAtomData	An output port that broadcasts the atom data.

GAMESS Input Generator (org.resurgence.moml.GamessInputGenerator)

Author: unknown **Version:** Unknown	The GamessInputGenerator actor is a composite actor that creates a GAMESS input file, which is used in computational chemistry workflows. The actor reads chemical modeling data in gamin format. To convert existing model data into the appropriate format, use the Babel or OpenBabel actors. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ScfType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
MpLevel	A parameter used to create the GAMESS input file.For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
RunType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
ExeType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
Charge	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
Multiplicity	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
TimeLimit	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.

ReplicatedMemory	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
DistributedMemory	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
GaussianBasisSet	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
NumberOfGaussians	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
HeavyAtomPolarizationFunctions	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
LightAtomPolarizationFunctions	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
HeavyAtomDiffuseShell	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
LightAtomDiffuseShell	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
DftType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
GamessInputDirectory	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
NameOfStep	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.

Ports

gamessDataGroupHandle	An input port that accepts molecular model files that are formatted for use in GAMESS software (e.g., gamin files)
gamessInputHandle	An output port that broadcasts the file names of parameterized GAMESS input files.

GAMESS Keywords (org.resurgence.moml. GamessKeywords)

Author: unknown **Version:** Unknown	The GamessKeywords actor is a composite actor used inside the GamessInputGenerator actor, which is used in computational chemistry workflows. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ScfType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
MpLevel	A parameter used to create the GAMESS input file.For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
RunType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
ExeType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
Charge	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
Multiplicity	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
TimeLimit	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
ReplicatedMemory	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.

DistributedMemory	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
GaussianBasisSet	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
NumberOfGaussians	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
HeavyAtomPolarizationFunctions	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
LightAtomPolarizationFunctions	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
HeavyAtomDiffuseShell	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
LightAtomDiffuseShell	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
DftType	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
GamessInputDirectory	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.
NameOfStep	A parameter used to create the GAMESS input file. For more information about GAMESS parameters, see http://www.msg.ameslab.gov/GAMESS/.

Ports

gamessInputName	An input port that accepts molecular model files that are formatted for use in GAMESS software (e.g., gamin files)
gamessAtomData	An input port that accepts atom data, which can be generated by the GamessAtomDataExtractor actor.
gamessInputHandle	An output port that broadcasts parameterized GAMESS input files.

GAMESS Local Run (org.resurgence.moml. GamessLocalRun)

Author: unknown **Version:** Unknown	The GAMESSLocalRun actor is a composite actor is that runs a local GAMESS application and outputs the result. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. The actor reads a GAMESS input file, which can be generated with the GamessInputGenerator actor. Set execution parameters by double-clicking the actor and entering the desired settings. GAMESS applications can also be run through Nimrod (with the GamessNimrodRun actor) or remotely using SSH and/or Globus technologies. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

GamessVersionNumber	The version number of the GAMESS application (e.g., Nov222004R1)
GamessRunScript	The path to the GAMESS script to run (e.g., /home/apps/gamess/rungms)
NumberOfCpus	The number of CPUs in the computer system used for calculations.

Ports

gamessInputHandle	An input port that accepts the file names of parameterized GAMESS input files. These input files can be generated with the GAMESSInputGenerator actor.
gamessOutputHandle	An output port that broadcasts the name of the results file.
gamessDataHandle	An output port that broadcasts the GAMESS data.

GAMESS Nimrod Run (org.resurgence.moml. GamessNimrodRun)

Author: unknown **Version:** Unknown	The GAMESSNimrodlRun actor is a composite actor that runs a GAMESS application through Nimrod (an application that allows computations to be run on the Grid). For more information about Nimrod, see http://www.csse.monash.edu.au/~davida/nimrod/ To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed as well. Set the actor's execution parameters by double-clicking the actor and entering the desired settings. The actor reads a GAMESS input file, which can be generated with the GamessInputGenerator actor. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

ExperimentName	The name of the experiment (i.e., collection of parallel jobs)e.g., gamess
GamessVersionNumber	The version number of the GAMESS application (e.g., Nov222004R1)
GamessRunScript	The path to the GAMESS script to run (e.g., /home/apps/gamess/ rungms)
NimrodBinary	The location of the Nimrod application (e.g., /opt/nimrodg-3.0.0/bin/ nimrod)
NumberOfCpus	The number of CPUs in the computer system used for calculations.
NumberOfMolecules	The number of molecules to run (e.g., 9)

Ports

gamessInputHandle	An input port that accepts the file names of parameterized GAMESS input files. These input files can be generated with the GAMESSInputGenerator actor.
gamessOutputHandle	An output port that broadcasts the name of the results file.
gamessDataHandle	An output port that broadcasts the GAMESS data.

GARP Algorithm (org.ecoinformatics.seek.garp.GarpAlgorithm)

Author: unknown
Version: Unknown

The GARPAlgorithm actor reads a set of spatial locations and associated environmental data, and uses a genetic algorithm to create a "rule set" that can be used to make predictions about the presence or absence of a species at various locations. The input data is passed to the actor by the GARPPresampleLayers actor, which generates the environmental data (or "layers") in the appropriate format. Output is usually passed to the GARPPrediction actor, which makes the environmental predictions based on the generated rule set. The actor requires libgarp.so (on linux systems) or garp.dll and libexpat.dll (on Windows systems). Currently, the actor does not work on MacOSX systems. GARP (Genetic Algorithm for Rule Set Production) is a genetic algorithm that creates an ecological niche model representing the environmental conditions where a species would be able to maintain populations. For more information about GARP, see http://www.lifemapper.org/desktopgarp/.

Parameters

cellSetFileNameParameter	The name of the input data file containing spatial and environmental information. This input is usually the output of the GarpPresampleLayers actor.
ruleSetFilenameParameter	The file name for the output rule set. The file name can also be specified via the cellSetFileName port.

Ports

ruleSetFilenameOutput	An output port that broadcasts the generated rule set file. The port is only fired after the rule set has been created, and is usually used as a trigger.
cellSetFileName	An input port that accepts the name of the data file containing the spatial and environmental information. This input is usually the output of the GarpPresampleLayers actor.
ruleSetFilename	An input port that accepts the file name for the output rule set. The file name can also be specified with the cellSetFileNameParameter parameter.

GARP Presample Layers (org.ecoinformatics.seek.garp.GarpPresampleLayers)

Author: Chad Berkeley, Dan Higgins, NCEAS, UC Santa Barbara **Version:** Unknown	The GARPPresampleLayers actor reads environmental and species location data, samples the data, and outputs the information in a format that other GARP actors can process. Environmental information is input in the form of "environmental layers", which contain environmental information (e.g., temperature or precipitation) and spatial data. Multiple layers are summarized in a single xml file (*.dxl), and passed to the actor. Location data consists of a file that contains a list of known points where a species has been found. This file is a text file, consisting of one location per line. Each line contains a point location (x,y) with the numeric values of x and y separated by a 'tab' character. Typically the (x, y) is (longitude, latitude). The output file is typically passed to the GARPAlgorithm actor, which can generate a predictive rule set from the information. The rule set can be used to predict other locations where the species might be found. The actor requires libgarp.so (on linux systems) or garp.dll and libexpat.dll (on Windows systems). Currently, the actor does not work on MacOSX systems. GARP (Genetic Algorithm for Rule Set Production) is a genetic algorithm that creates an ecological niche model representing the environmental conditions where a species would be able to maintain populations. For more information about GARP, see http://www.lifemapper.org/desktopgarp/.

Parameters

cellSetFileNameParameter	The file name for the generated output file. This file is usually passed to the GARPAlgorithm actor, which creates a rule set for predicting species occurrences. This information can also be specified via the cellSetFileName port.
dataPointFileNameParameter	The file name of the species location data. Location data consists of a list of known points where a species has been found. This file is a text file, consisting of one location per line. Each line contains a point location (x,y) with the numeric values of x and y separated by a 'tab' character. Typically the (x,y) is (longitude, latitude). This information can also be specified via the dataPointFileName port.

layersetFilenameParameter	The file name of the environmental data. Environmental data is input in the form of environmental layers containing environmental information (e.g., temperature or precipitation) and spatial data. Multiple layers are summarized in a single xml file (*.dxl), and passed to the actor. This information can also be specified via the layersetFilename port.

Ports

cellSetFileName	An input port that accepts a file name for the output file. The output file is usually passed to a GARPAlgorithm actor, which can generate a predictive rule set from the information. This information can also be specified with the cellSetFileNameParameter.
dataPointFileName	An input port that accepts the file name of the species location data. Location data consists of a file that contains a list of known points where a species has been found. This file is a text file, consisting of one location per line. Each line contains a point location (x,y) with the numeric values of x and y separated by a 'tab' character. Typically the (x,y) is (longitude, latitude). This information can also be specified with the dataPointFileNameParameter.
layersetFilename	An input port that accepts the file name of the environmental data. Environmental data is input in the form of environmental layers containing environmental information (e.g., temperature or precipitation) and spatial data. Multiple layers are summarized in a single xml file (*.dxl), and passed to the actor. This information can also be specified with the layersetFilenameParameter.
cellSetFileNameOutput	An output port that broadcasts the file name of the output file. The port is only fired after the output data has been calculated, and is thus used as a trigger for the next step in the GARP calculation.

GARPSummary (org.ecoinformatics.seek.gis.java_gis. GARPSummary)

Author: Dan Higgins
Version: Unknown

The GARPSummary actor examines a predicted species distribution and calculates error statistics (omission and commission). Because GARP predictions of the spatial distribution of a species are probabilistic, they will differ every time a calculation is executed. The GARPSummary actor generates a measure of how well a given run does in predicting distributions, based on a "test set" of known species locations. The actor reads a predicted species distribution (an ASCII raster grid file that can be output by a GARPPrediction actor) as well as a list of test locations consisting of (long, lat) points. In addition, the actor reads a rule set file, which is required to reproduce the predicted distribution. The actor outputs the omission and commission values, as well as the rule set file name after the calculations have been performed. Omission is the fraction of test set points that are not predicted by the calculation. Commission is the proportion of area predicted present with regard to the total area of interest, not counting masked pixels. GARP (Genetic Algorithm for Rule Set Production) is a genetic algorithm that creates an ecological niche model representing the environmental conditions where a species would be able to maintain populations. For more information about GARP, see http://www.lifemapper.org/desktopgarp/.

Ports

omissionValue	An output port that broadcasts the omission value.
output	An output port that broadcasts the file name of the ASCII raster grid file.
ruleSetFileName	An input port that accepts the file name of the rule set used to create the predicted species distribution.
input	An input port that accepts the file name of an ASCII raster grid file (*.asc) containing predicted species distribution data.
outputRuleSetFileName	An output port that broadcasts the file name of the rule set.
commissionValue	An output port that broadcasts the commission value.
pointFileName	An input port that accepts the file name of a file containing a list of locations (long, lat) used to evaluate the prediction. This file is a text file, consisting of one location per line. Each line contains a point location (x,y) with the numeric values of x and y separated by a 'tab' character.

GDAL Format Translator (org.ecoinformatics.seek.gis.gdal.GDALTranslateActor)

Author: Chad Berkley **Version:** Unknown	The GDALFormatTranslator actor reads a geospatial raster file and translates it to a specified format (e.g., JPEG, AAIGrid, etc). Before performing the translation, the actor checks the Kepler file cache to see if the translated file already exists. The actor outputs the name of the translated file (or the name of an existing cached version). The actor uses the GDAL (Geospatial Data Abstraction Library) translation library to perform the translation. For more information about GDAL, see http://www.gdal.org/index.html. Translated files are often stored in the Kepler cache. Files are cached by their file names (without the file extension) and their output format. Use the actor's cacheOption parameter to specify whether the output should be copied to the cache ("Copy files to cache"), copied to the cache as well as the directory where the input raster is stored ("Cache files but preserve location"), or not cached ("No caching"). If "No caching" is selected, the actor will not cache the translated file and will ignore all previously stored cache items. Select this option to force the actor to perform a translation even if the input file was previously translated and cached.

Parameters

output type	The type of output (e.g., "byte" or "int").
Cache options	Specify whether the output should be copied to the cache ("Copy files to cache"), copied to the cache as well as the directory where the input raster is stored ("Cache files but preserve location"), or not cached ("No caching"). If "No caching" is selected, the actor will not cache the translated file and will ignore all previously stored cache items. Select this option to force the actor to perform a translation even if the input file was previously translated and cached.
output format	The format to which the input raster file should be translated (e.g., JPEG, AAIGrid, etc).

Ports

inputFilename	An input port that accepts the file name of the input geospatial raster file.

trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
outputCacheType	An output port that broadcasts the specified output format (e.g., "GTiff" or "AAIGrid")
outputFilename	An output port that broadcasts the path to the translated file.
outputCachename	An output port that broadcasts the cache location of the translated output file. If "No cache" is selected as the cacheOption, the port will broadcast nothing.

GDAL Warp and Projection (org.ecoinformatics.seek.gis.gdal.GDALWarpActor)

Author: Chad Berkley **Version:** Unknown	The GDALWarpAndProjection actor "streches" or "warps" a geospatial raster file (e.g., a digital elevation model) from one cartographic projection to another. Before creating the projection, the actor checks the Kepler file cache to see if the projected file already exists. The actor outputs the name of the projected file (or the name of an existing cached version). The actor uses the GDAL (Geospatial Data Abstraction Library) library to create the projection. For more information about GDAL, see http://www.gdal.org/index.html. Projected files are often stored in the Kepler cache. Files are cached by their file names (without the file extension) and their output format. Use the actor's cacheOption parameter to specify whether the output should be copied to the cache ("Copy files to cache"), copied to the cache as well as the directory where the input data is stored ("Cache files but preserve location"), or not cached ("No caching"). If "No caching" is selected, the actor will not cache the translated file and will ignore all previously stored cache items. Select this option to force the actor to perform a translation even if the input file was previously translated and cached. The inputParams and outputParams parameters specify the format for the coordinate system. The parameter values must be of a form used by the GDAL Warp utility. See the -s_srs and -t_srs parameters of the GDAL Warp utility for more information about accepted forms: http://www.remotesensing.org/gdal/gdalwarp.html.

Parameters

output params	The format for the coordinate system of the output projection. The parameter values must be of a form used by the GDAL Warp utility. For more information about supported formats, see www.remotesensing.org/geotiff/proj_list/. For example, a simple latitude/longitude projection is specified as +proj=latlong.
input params	The format for the coordinate system of the input data. The parameter values must be of a form used by the GDAL Warp utility. For more information about supported formats, see www.remotesensing.org/geotiff/proj_list/. For example, a Lambert Azimuthal Eagual Area Projection could be specified as +proj=laea+lat_0=45+long_0=-100 +x_0=0+y_0=0.

Cache options	Specify whether the output should be copied to the cache ("Copy files to cache"), copied to the cache as well as the directory where the input raster is stored ("Cache files but preserve location"), or not cached ("No caching"). If "No caching" is selected, the actor will not cache the projection file and will ignore all previously stored cache items. Select this option to force the actor to create a projection even if a projection file was previously created and cached.
output format	The format of the output projection (e.g., GTiff).

Ports

inputFilename	An input port that accepts the file name of the input geospatial raster file. The actor can process any of the dozens of GIS raster formats supported by GDAL.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
outputCacheType	An output port that broadcasts the specified output format (e.g., "GTiff" or "AAIGrid")
outputFilename	An output port that broadcasts the path to the translated file.
outputCachename	An output port that broadcasts the cache location of the translated output (a file path). If "No cache" is selected as the cacheOption, the port will broadcast nothing.

GML Displayer (org.kepler.gis.display.JumpGMLDisplayer)

Author: Jianting Zhang **Version:** Unknown	The GMLDisplayer accepts a string of GML (Geography Markup Language) data and displays it on the screen. GML is an XML-based encoding for geographic information. For more information about GML, see http://www.w3.org/Mobile/posdep/GMLIntroduction.html. Known problem: the actor currently omits any attribute data in GML. Allowing attribute data by setting proper schema format is planned.

Ports

GMLStringPort	An input port that accepts a string of GML data. For more information about GML, see http://www.w3.org/Mobile/posdep/GMLIntroduction.html.

GUIRunCIPRes (org.cipres.kepler.GUIRunCIPRes)

Author: Zhijie Guan **Version:** Unknown	The GUIRunCIPRES actor helps facilitate phylogenic analysis by invoking an external CIPRES application and outputting its response. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. PAUP software is required in some instances as well. The actor generates a graphical interface that allows users to input information (e.g., a command or working directory) and pass the data to the CIPRES application. Once the application has been contacted and has processed the input, the actor returns the generated results, or an error if the application did not execute correctly. If the actor is used to actor is used to contact a CIPRES CORBA service, it must be used with the Initializer actor, which connects to a CIPRES registry. The actor can also be used to monitor error words in the generated results. Whenever the actor identifies any of the user-specified error words, it will notify the user and ask for further instructions. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Parameters

monitoredErrorWords	Words to monitor. Whenever the actor identifies any of the specified error words, it will notify the user and ask for further instructions.
parameterToOutput	The parameter to return. Usually the parameter is "outfile".
command	Commands are automatically generated and sent to the CIPRES application via the actor's graphical user interface.
uiXMLFile	The actor uses the specified file to generate its user interface
outputFile	The standard output file path (e.g., C:\Documents and Settings \Myself\results.txt). Results are written in Nexus format.
workingDirectory	The working directory of the external CIPRES program.
parameterForOutput	The parameter to return. Usually the parameter is "outfile".
errorFile	The standard error file path (e.g., C:\Documents and Settings\Myself \errors.txt).

Ports

standardError	An output port that broadcasts the standard error stream of the execution.

inputParameterValue	An input port that accepts the name of an input file. Input files should contain the data to be analyzed, and should be in the format required by the CIPRES application.
inputParameterName	An input port that accepts the name of the input data (usually "infile").
exitCode	An output port that broadcasts the exit code of the execution.
outputParameterValue	An output port that broadcasts the name of the file that contains the execution results.
standardOutput	An output port that broadcasts results of the execution.

Garp Prediction (org.ecoinformatics.seek.garp.GarpPrediction)

Author: Chad Berkeley, Dan Higgins, NCEAS, UC Santa Barbara **Version:** Unknown	The GARPPrediction actor predicts the distribution of a species in given locations based on a predefined rule set. The actor outputs a bitmapped image (a map) representing species presence or absence with different pixel values (e.g., different colors). The input rule set is usually passed to the actor by the GARPAlgorithm actor, which creates the rule set and outputs it in the proper format. Input locations with environmental data, or "environmental layers", are passed to the GARPPrediction actor in a summary xml file (*.dxl). The actor outputs the predicted species distribution in two formats: bitmap and ASCII raster grid. Both files contain the same information. The bitmap file is often more easily displayed; the ASCII raster grid can be used by the GARPSummary actor to test the accuracy of the predicted distribution. The actor requires libgarp.so (on linux systems) or garp.dll and libexpat.dll (on Windows systems). Currently, the actor does not work on MacOSX systems. GARP (Genetic Algorithm for Rule Set Production) is a genetic algorithm that creates an ecological niche model representing the environmental conditions where a species would be able to maintain populations. For more information about GARP, see http://www.lifemapper.org/desktopgarp/.

Parameters

ruleSetFilenameParameter	The file name of the file containing the rule set data (usually the output of a GarpAlgorithm actor).
outputASCIIParameter	The file name to be used for the output ASCII grid file (*.asc).
outputBMPParameter	The file name to be used for the output bitmap file (*.bmp)
layersetFilenameParameter	The file name of the location and environmental data. The file is a *.dxl file, which summarizes a set of spatial data files that contain environmental data.

Ports

ruleSetFilename	An input port that accepts the file name of the file containing the rule set data (usually the output of a GarpAlgorithm actor).
layersetFilename	An input port that accepts the file name of the location and environmental information (the environmental layers). The file is a *.dxl file, which summarizes a set of spatial data files.

outputBMPFileName	An output port that broadcasts the file name of the generated bitmap file after the file has been created. The bitmap file contains the same information as the ASCII grid file, but in a more easily displayed raster format.
outputBMP	The file name for the output bitmapped file (*.bmp).
outputASCIIFileName	An output port that broadcasts the file name of the ASCII grid file containing the predicted species distribution. The port fires after the file has been created.
outputASCII	The file name for the output ASCII grid file (*.asc).

Gaussian Distribution Random Number Generator (ptolemy.actor.lib.Gaussian)

Author: Edward A. Lee
Version: Unknown

The GaussianDistributionRandomNumberGenerator actor outputs a new random number each time the actor iterates. The output has a Gaussian distribution. Output values are independent and identically distributed with the mean and the standard deviation given by parameters. In addition, a random number seed can be specified as a parameter to control the generated sequence.

Parameters

mean	The mean of the random numbers. The value is a double that defaults to 0.
seed	The seed that controls the random number generation. A seed of zero (the default) means that the seed is derived from the current system time and a Java hash code (i.e., System.currentTimeMillis() + hashCode()). With extremely high probability, the default seed will ensure that two distinct actors will have distinct seeds. However, current time may not have enough resolution to ensure that two subsequent executions of the same model have distinct seeds. The parameter contains a long token, initially with value 0.
standardDeviation	The standard deviation of the random numbers. The value is a double that defaults to 1.
resetOnEachRun	Select to reset the random number generator each time the workflow is run. By default, the generator does not reset.

Ports

output	An output port that broadcasts the generated random numbers.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Get 2D Mineral Composition Point (org.geon. Get2DPoint)

Author: Efrat Jaeger
Version: Unknown

The Get2DPoint actor is a project-specific actor used within the GEON mineral classifier workflow for calculating a classification point given mineral composition and coordinate names The Get2DPoint actor reads information about mineral composition (e.g., data from the Virginia Igneous Rocks database) as well as an expression representing (x,y) coordinates. The actor evaluates the expression and returns an array of two doubles representing the specified (x,y) classification point. The actor accepts mineral composition data in XML-string format via its rowInfo port. An expression representing coordinates is passed via the coordinates port as an array (e.g., {"svg", "(clinozoisite + plagioclase) * 2","chlorite * epidote"}). GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/.

Ports

point	An output port that broadcasts the calculated classification point.
rowInfo	An input port that accepts mineral composition information in XML-string format.
coordinates	An input port that accepts an expression representing the coordinates to be evaluated.

Get Mineral Composition Point (org.geon.GetPoint)

Author: Efrat Jaeger **Version:** Unknown	The GetPoint actor is a project-specific actor used within the GEON mineral classifier workflow for calculating a classification point given mineral composition and coordinate names. The GetPoint actor reads information about mineral composition (e.g., data from the Virginia Igneous Rocks database) as well as an expression representing (x,y) coordinates. The actor evaluates the expression and returns an array of two doubles representing the specified (x,y) classification point. The actor accepts mineral composition data in XML-string format via its rowInfo port. An expression representing coordinates is passed via the coordinates port as an array (e.g., {"svg", "(clinozoisite + plagioclase) * 2","chlorite * epidote"}). GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/.

Ports

point	An output port that broadcasts the calculated classification point.
rowInfo	An input port that accepts mineral composition information in XML-string format.
coordinateNames	An input port that accepts an expression representing the coordinates to be evaluated.

Globus Job (org.nmiworkflow.GlobusJob)

Author: no author given
Version: Unknown

The GlobusJob actor submits an executable or command (a "job") to a Globus host, allowing users to take advantage of remote computational resources. The job output is gathered and sent to the output port as a string. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org. A proxy certificate, generated by a GlobusProxy actor, must be passed to the actor via the input port. Specify the name of a Globus server with the Globus Host parameter, and the job instructions with the RSL String parameter. RSL ("Resource Specification Language") strings are used to define Globus jobs. A full RSL string containing an execuatable and argument must be specified. For more information about RSL strings, see http://www.globus.org/.

Parameters

Globus Host	The name of a Globus host (e.g., griddle.sdsc.edu).
RSL String	The RSL ("Resource Specification Language") string specifing the Globus job to run. Currently, a full RSL string (e.g., (executable is /bin/cat)(arguments is /tmp/pas.local)) must be specified. For more information about RSL strings, see http://www.globus.org/.

Ports

output	An output port that broadcasts the job results as a string.
input	An input port that accepts a proxy certificate produced by the GlobusProxy actor.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Globus Proxy (org.nmiworkflow.GlobusProxy)

Author: stephen mock **Version:** Unknown	The GlobusProxy actor creates a proxy certificate that can be used by other actors to access a Globus server. A certificate allows actors to access the Grid without requiring that the user type in a password. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org.

Parameters

key file	The location of the Globus key file (userkey.pem), which is an encrypted file stored on the local computer. The key is issued by a trusted Grid authority, called the Certificate Authority (CA).
cert file	The location of the user's certificate file (usercert.pem), which contains information that helps identify and authenticate the user. The certificate file is stored on the local computer, and is issued by a trusted Grid authority, called the Certificate Authority (CA).
passphrase	The passphrase used to decrypt the key file.

Ports

output	An output port that broadcasts a Globus proxy certificate, which can be used by other actors to access the Grid.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Grass Buffer (org.ecoinformatics.seek.gis.grass. GISBufferActor)

Author: Jianting Zhang
Version: Unknown

The GRASSBuffer actor expands a "mask file" generated by the GRASSRaster actor by creating a "buffer" around the outer boundary of the "in-scope" raster cells (cells with a value of 1). The actor saves the new, expanded mask raster and outputs the path. The actor is similar to the CVHullToRaster actor, only it is based on GRASS (Geographic Resources Analysis Support System), an open source software toolkit used to manage and analyze geospatial data and produce graphics and maps. For more information about GRASS, see http://grass.itc.it/.

Ports

rasterFileName	The name of the input mask file. This file is generated by the GRASSRaster actor.
numRasterRows	The number of rows in the input raster. This is an integer.
numRasterCols	The number of columns in the input raster. This is an integer.
bufferFileResult	The name of the "buffered" file the actor creates.
numDistances	The number of buffered buffer cells to add around non-NULL cells. See http://grass.itc.it/grass62/manuals/html62_user/r.buffer.html for more information.
valDistances	The distance values to add. See http://grass.itc.it/grass62/manuals/html62_user/r.buffer.html for more information.
bufferFileName	The name to give to the buffered output file.

Grass Hull (org.ecoinformatics.seek.gis.grass.GISHullActor)

Author: Jianting Zhang **Version:** Unknown	The GRASSHull actor reads a set of (x,y) points, calculates the points that define the convex hull (i.e., the smallest polygon that contains the given points), and saves the hull points to a file. The actor outputs the number of points in the convex hull, as well as the name of the file that contains the calculated hull points. The actor is similar to the ConvexHull actor, only it is based on GRASS (Geographic Resources Analysis Support System), which is an open source software toolkit used to manage and analyze geospatial data and produce graphics and maps. For more information about GRASS, see http://grass.itc.it/. The pointFileName port actor accepts a set of (x,y) points as a tab-delimited text file. Use the hullFileName input port to specify a name for the generated output file. The actor outputs the number of (x,y) pairs contained in the generated hull file via the numHullPoint point, and the name of the convex hull file via the hullFileResult port. Note that the value of the hullFileName input port and the hullFileResult port are the same (the output is used as a trigger).

Ports

hullFileResult	An output port that broadcasts the hull file name (the output is used as a trigger). The value is the same as that of the hullFileName input.
hullFileName	An input port that accepts a name to be given to the output convex hull point list file
pointFileName	An input port that accepts a tab-delimited text file containing (x,y) input points
numHullPoint	An output port that broadcasts the number of (x,y) pairs in the hull file.

Grass Raster (org.ecoinformatics.seek.gis.grass. GISRasterActor)

Author: Jianting Zhang **Version:** Unknown	The GRASSRaster actor reads a list of (x,y) points representing a convex hull polygon (i.e., the smallest polygon that contains a set of given points). The actor creates a "mask" based on the input: points within the convex hull are set to a value of 1 and points outside the hull have a value of "NO_DATA." The actor saves the mask as a raster file and outputs the path of the file. The actor is based on GRASS (Geographic Resources Analysis Support System), which is an open source software toolkit used to manage and analyze geospatial data and produce graphics and maps. For more information about GRASS, see http://grass.itc.it/. The hullFileName port accepts a space-delimited text file containing one pair of (x,y) convex hull points per line. The GRASSHull actor can be used to generate this input. Use the rasterFileName input port to specify a name for the output raster file. The size and resolution of the generated raster file can be specified with the numRasterRows, numRasterCols, xMin, xMax, yMin and yMax parameters. Note: Use the GrassBuffer actor to expand the masked area.

Ports

rasterFileResult	An output port that broadcasts the path of the generated mask file, which the actor creates and saves in *.asc format.
numHullPoint	An input port that accepts the number of hull points in the input convex hull file.
numRasterRows	An input port that accepts the number of rows for the output mask file. This is an integer.
xmin	An input port that accepts the minimum x-value for the output mask file. Set xMin to -1 to set the parameter to the minimum x-value in the input convex hull file.
ymax	An input port that accepts the maximum y-value for the output mask file. Set yMax to -1 to set the parameter to the maximum y-value in the input convex hull file.
xmax	An input port that accepts the maximum x-value for the output mask file. Set xMax to -1 to set the parameter to the maximum x-value in the input convex hull file.
rasterFileName	An input port that accepts a name to be given to the output mask file.

hullFileName	An input port accepting a space-delimited text file containing (x,y) convex hull data points (one pair of points per line).
ymin	An input port that accepts the minimum y-value for the output mask file. Set yMin to -1 to set the parameter to the minimum x-value in the input convex hull file.
numRasterCols	An input port that accepts the number of columns for the output mask file. This is an integer.

GridFTP (org.nmiworkflow.GridFTP)

Author: null **Version:** Unknown	The GridFTP actor copies files from any remote Globus server to a specified host machine. Copied files are placed in the location specified by the destinationHostname and fullPathToDestinationFile parameters. The output file paths are gathered and sent to the output port as a string. The actor must be used with a GlobusProxy actor, which passes a proxy certificate used to connect to the remote host. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org. GridFTP is a secure data transfer protocol optimized for wide-area networks.

Parameters

Source hostname	The name of a Globus machine source (e.g., "griddle.sdsc.edu")
Full path to source file	The full path to the files to fetch (e.g., "/archive/2000")
Destination hostname	The name of the destination host (e.g., "localhost")
Full path to destination file	The full path in which to place the fetched files.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
input	An input port that accepts a proxy certificate produced by the GlobusProxy actor.
output	An output port that broadcasts the new file paths as a semicolon-separated string.

Grid Overlay (org.geon.GridOverlay)

Author: Efrat Jaeger
Version: Unknown

The GridOverlay actor is a GEON specific actor used in a gravity modeling design workflow. The actor has not yet been implemented. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/.

Ports

output	Not yet implemented
input	Not yet implemented

Grid Rescaler (org.ecoinformatics.seek.gis.java_gis. GridRescaler)

Author: null **Version:** Unknown	The GridRescaler actor "rescales" a raster grid file so that it has a new extent and cellspacing. The actor outputs the name of the rescaled file. The actor reads the file name of one or more ASCII grid files via the input port. The input port is a multiport. If multiple files are input, the actor will output a sequence of output file name tokens. The new file is generated based on the values specified in the actor's parameters. The actor uses an algorithm to convert cell values in the output grid from the values in the input grid. The actor currently supports two algorithms: 'Nearest Neighbor' or 'Inverse Distance Weighted'. Algorithms are disk based, allowing very large rasters to be converted.

Parameters

xllcorner	A double token representing the x-value (usually a longitude) of the lower-left corner of the raster. If the parameter is empty, the x-value is set to the minimum x in the convex hull.
yllcorner	A double token representing the y-value (usually a latitude) of the lower-left corner of the raster. If the parameter is empty, the y-value is set to the minimum y in the convex hull.
cellsize	A double token representing the cell size for the raster (assumed square). If the parameter is empty, the actor automatically generates a raster with 50 x-direction cells and y-direction cells to match the y-extent of the convex hull.
numrows	An integer representing the number of rows for the raster.
numcols	An integer representing the number of columns for the raster.
use disk storage (for large grids)	Specify whether or not to use disk for storing grid data rather than putting all data in RAM arrays during processing. Selecting this option can result in a much slower execution, but allows for very large raster files.

algorithm	The algorithm used to calculate cell values in the output grid from the values in the input grid. The actor currently supports two algorithms: 'Nearest Neighbor' or 'Inverse Distance Weighted'. Algorithms are disk based, allowing very large rasters to be converted.
use Existing File	Select to instruct th e actor to check for currently existing output files. If the specified output file exists, the actor will output the name of the existing file, without repeating the calculation.
outputFileName	The name of the output file. If the parameter is left empty, the output file will be given the name of the input file plus a suffix (".out"+i). If the parameter is a directory name, output file(s) will be placed in the specified directory. Note that the input port is a multiport so multiple input files can be converted in a single firing.

Ports

input	The name of the the ASCII grid file or files to be converted.
output	The name or names of the converted ASCII grid files.

GridReset (org.ecoinformatics.seek.gis.java_gis.GridReset)

Author: Dan Higgins NCEAS UC Santa Barbara **Version:** Unknown	The GridReset actor transforms a specified range of values in a grid to some other value without changing the cell size or extent of the grid. The actor reads an input grid file, and replaces a range of values with a new value specified in the actor's parameters. The actor outputs a new grid file. If no new value is specified in the actor parameters, the actor will transform the existing value based on the transformation parameters: the actor will multiply the existing value by the value of the multiplicationFactor parameter and add the value of the additionParameter.

Parameters

useDisk	Boolean setting to determine whether or not to use disk for storing grid data rather than putting all data in RAM arrays during processing. This option in much slower but allows for very large raster files.
minvalParameter	The minimum value in the replacement range. Grid values that fall above this value (and below the maximum value) will be replaced.
additionParameter	The value to add to the existing value if the newValue parameter is left empty.
multiplicationFactor	The factor to multiply the existing value by if the newValue parameter is left empty.
maxvalParameter	The maximum value in the replacement range. Grid values that fall below this value (and above the minimum value) will be replaced.
outputFileName	The file name of the new grid file.
newvalParameter	The new value used to replace the value of cells that fall in the specified replacement range (i.e., between the minimum and maximum values). If no value is specified, the actor will transform the existing value based on the multiplicationFactor and the additionParameter.

Ports

output	An output port that broadcasts the new grid file.
input	An input port that accepts a grid file.

Griddles Exec **(org.monash.griddles.GriddlesExec)**

Author: Jagan Kommineni
Version: Unknown

The GriddlesExec actor securely connects to a Web service (a computer program that runs on a remote host and communicates using a standardized protocol) and executes a command at a specified remote location. The actor finishes executing the command and then outputs the result as a string, along with any error messages. GriddLeS is a tool used to create Grid workflows that use legacy software, which has not been designed for distributed use. GriddLeS provides the input/output mechanism that allows components to communicate. For more information about GriddLes, see http://www.csse.monash.edu.au/~davida/griddles/index.htm. The Grid consists of geographically distributed resources (computers or scientific instruments, for example) that can be easily accessed, allowing users to share computing power, databases, and other tools.

Parameters

command	The command to execute. Can be passed via either the command input port or parameter.
directory	The directory in which to execute the command. The default value of this parameter $HOME.
environment	An array of records that name an environment variable and a value: {{name = "NAME1", value = "value1"}...} Where NAME1 is the name of the environment variable, and value1 is the value. For example {{name = "PTII", value = "c:/ptII"}} would set the value of the PTII to c:/ptII. If the parameter is set to {{name="", value = ""}}, then the environment from the parent process is used. If environmental variables are set with the parameter, the parent values are not passed to the process. To view the current environment, use the "env" command.
cmdArgs	Arguments to pass to the command.
executeHostName	The name of the remote host where the command will be executed (e.g., brecca-2.vpac.org)
provider	????
hostPort	The port number to which to connect (e.g., 33755)
isSSHinteractive	???
gsiFtpThirdPartyTransfer	The protocol to use when copying files. By default, the actor uses the gsiFTP protocol.
userName	The user name for the remote system.

sshPrivateKey	The file path to the user's SSH identity file. This file is located on the local machine and used to access the remote one.
modelReference	??? The name of a remote model (e.g., the global climate model, C-CAM or a regional weather model, DARLAM)
hasTrigger	Select to activate a trigger port.

Ports

input	An input port that accepts a string, which is passed to the command (e.g., a file name)
configInfoPORT	???
error	An output port that broadcasts any errors generated by the command execution. Errors are output as a string after the command has finished executing. If the execution generates no errors, an empty string is output.
output	An output port that broadcasts data generated by the executed command. Results are output as a string after the command has finished executing.

Griddles Input File (org.monash.griddles.GriddlesInputFile)

Author: null **Version:** Unknown	The GriddlesInputFile actor points to a remote Grid file. This actor must be used with a GriddlesExec actor, which establishes a connection to a remote Griddles host. GriddLeS is a tool used to create Grid workflows that use legacy software, which has not been designed for distributed use. GriddLeS provides the input/output mechanism that allows components to communicate. For more information about GriddLes, see http://www.csse.monash.edu.au/~davida/griddles/index.htm. The Grid consists of geographically distributed resources (computers or scientific instruments, for example) that can be easily accessed, allowing users to share computing power, databases, and other tools.

Parameters

fileURL	The file path of a remote file (e.g., gsiftp://dione.csse.monash.edu.au/home/user/test)

Ports

GriddlesInputFile	The file path of a remote file. (e.g., gsiftp://dione.csse.monash.edu.au/home/user/test)

Griddles Output File (org.monash.griddles. GriddlesOutputFile)

Author: null **Version:** Unknown	The GriddlesOutputFile actor points to a remote Grid file. This actor must be used with a GriddlesExec actor, which establishes a connection to a remote Griddles host. The actor will write data to the remote file, save, and close the file when the workflow is done. GriddLeS is a tool used to create Grid workflows that use legacy software, which has not been designed for distributed use. GriddLeS provides the input/output mechanism that allows components to communicate. For more information about GriddLes, see http://www.csse.monash.edu.au/~davida/griddles/index.htm. The Grid consists of geographically distributed resources (computers or scientific instruments, for example) that can be easily accessed, allowing users to share computing power, databases, and other tools.

Parameters

fileURL	The file path of a remote file. (e.g., gsiftp://dione.csse.monash.edu.au/home/user/test)

Ports

GriddlesOutputFile	The file path of a remote file. (e.g., gsiftp://dione.csse.monash.edu.au/home/user/test)

GriddlesParameter (org.monash.griddles. GriddlesParameter)

Author: null **Version:** Unknown	The GriddlesParameter is used to select a transfer protocol for use with the GriddlesExec actor. Select either scp or gsiftp, or specify another protocol by typing it into the parameter by hand. GriddLeS is a tool used to create Grid workflows that use legacy software, which has not been designed for distributed use. GriddLeS provides the input/output mechanism that allows components to communicate. For more information about GriddLes, see http://www.csse.monash.edu.au/~davida/griddles/index.htm. The Grid consists of geographically distributed resources (computers or scientific instruments, for example) that can be easily accessed, allowing users to share computing power, databases, and other tools.

IJMacro (util.IJMacro)

Author: Dan Higgins **Version:** Unknown	The IJMacro actor runs ImageJ macros, which are used to display, edit, analyze, process, save, and print a wide variety of images. For more information about ImageJ, see http://rsb.info.nih.gov/ij/. The actor accepts a string representing an image file name via the input port or the fileOrURL parameter. The actor will perform the macro specified in the macroString parameter. By default, the actor will open the specified image. The ImageJ menu toolbar, which appears when the actor opens a specified image, can be used to create additional macros. Select "Plugins > Macros > Record" from the ImageJ menu to create a new macro that can be used by the actor.

Parameters

fileOrURL	The file name or URL of an image to process. The file name may also be input through the input port.
macroString	The ImageJ macro to execute. The macro may include the expression "_FILE_", which will be replaced by the path of the specified image file.

Ports

input	An input port that accepts a string representing the path of an image

Image (ptolemy.vergil.kernel.attributes.ImageAttribute)

Author: Edward A. Lee and Steve Neuendorffer **Version:** Unknown	The Image attribute renders an image on the Workflow canvas. Specify a source image (a GIF, JPEG, etc) with the source parameter.

Parameters

source	The source image file. Specify a file name or URL. The default is "$CLASSPATH/ptolemy/vergil/kernel/attributes/ptIIplanetIcon.gif".

Image Contrast (ptolemy.domains.sdf.lib.vq.ImageContrast)

Author: Michael Leung, Steve Neuendorffer
Version: Unknown

The ImageContrast actor changes the contrast of an image. If the input image contains many pixels with the same or similar color, the actor uses gray scale equalization to redistribute the value of each pixel between 0 and 255.

Ports

output	An output port that broadcasts the processed image.
input	An input port that receives an image.

Image Converter **(util.ImageConverter)**

Author: No author given **Version:** Unknown	The ImageConverter actor converts an image to the specified format (e.g., TIFF, PNG, JPG, etc). The actor reads the url of an image, converts the image, and saves the converted file in the same directory as the original source image. The actor outputs the path of the converted file.

Parameters

convertTo	The name of the format to which the source image should be transformed.

Ports

inputimageFilename	An input port that accepts the path to an image file.
outputimageFilename	An output port that broadcasts the path of the converted file.

Image Display (ptolemy.actor.lib.image.ImageDisplay)

Author: James Yeh, Edward A. Lee
Version: Unknown

The Image Display actor reads an image token and displays the image on the screen. If the actor receives a sequence of images that are all the same size, it will continually update the display with the new data. If the size of the input image changes, the actor generates a new picture display. Image tokens can be generated from image URLs with the ImageReader or the ConvertURLToImage actors.

Ports

input	A multiport that accepts image tokens to display. Use the ImageReader or ConvertURLToImage actors to create an image token from a specified image path.

ImageJ (util.ImageJActor)

Author: Dan higgins **Version:** Unknown	The ImageJ actor reads an image file name and opens and displays the image along with a toolbar of image-processing options, which can be used to process the image. The actor uses the ImageJ application to open and work with images. ImageJ can be used to display and process a wide variety of images (tiffs, gifs, jpegs, etc.) For more information about ImageJ, see http://rsb.info.nih.gov/ij/.

Parameters

fileOrURL	The file name or URL of an image to process (a tiff, gif, jpeg, or any other format supported by ImageJ). The file name may also be input through the input port.

Ports

input	A multiport that accepts the path of an image file (a tiff, gif, jpeg, or any other format supported by ImageJ). The file name can also be input through the fileOrURL parameter.

Image Reader (ptolemy.actor.lib.image.ImageReader)

Author: Christopher Hylands
Version: Unknown

The ImageReader actor reads an image path (e.g., C:\pictures\signature.jpg), and outputs it as an image token, which can be displayed and/or manipulated by other Kepler actors, such as ImageRotate or ConvertImageToString. Currently, the actor only reads image paths via a parameter. To read and convertn an image path to an image token via an input port, use the ConvertURLToImage actor.

Parameters

fileOrURL	The file name or URL of the image to be read. See FileParameter for more information about specifying file names.

Ports

output	An output port that broadcasts an image token.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Image Rotate (ptolemy.actor.lib.image.ImageRotate)

Author: Christopher Hylands
Version: Unknown

The ImageRotate actor reads an image token, rotates the image by the specified number of degrees, and outputs the transformed image as an image token. Use the ImageReader or the ConvertURLToImage actor to read an image URL or path and convert it to an image token that the ImageRotate actor can use. Use the ImageDisplay actor to view the rotated image.

Parameters

rotationInDegrees	An integer representing the number of degrees to rotate the image. The default is 90.

Ports

output	An output port that broadcasts the rotated image as an image token.
input	An input port that accepts an image token.

Initializer (org.cipres.kepler.Initializer)

Author: Zhijie Guan **Version:** Unknown	The Initializer actor initializes the registry of CORBA services required by CIPRES actors that invoke CORBA services (e.g., RecIDCM3 or PAUPInfer). CORBA services, much like Web services, are computer programs that run on a remote host and communicate using a standardized protocol that allows them to interoperate. CIPRES services, such as Tree Decomposer, are specifically designed to help analyze phylogenetic data sets. Once the Inititalizer actor has successfully connected to the registry, it broadcasts a trigger signal that can be used by other actors that require access to the registry. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

outputTrigger	An output port that broadcasts a trigger signal after the registry has been initialized successfully. The signal may be used by other actors that require access to the registry.

IntRangeParameter (ptolemy.actor.parameters.IntRangeParameter)

Integrator (ptolemy.domains.ct.lib.Integrator)

Author: Jie Liu
Version: Unknown

The Integrator actor is used with a CT Director to help solve ordinary differential equations (ODEs). The actor helps control the accuracy of the ODE solution by adjusting integration step sizes. The actor has memory, which is its state. The actor has one input port and one output port. Conceptually, the input is the derivative of the output with respect to time. So an ordinary differential equation dx/dt = f(x, t) can be built by: +---------------+ dx/dt | | x +--------->| Integrator |---------+-----> | | | | | +---------------+ | | | | |---------------| | +-------------| f(x, t) |------+ | Expression | |---------------| The above schematic can be built with two actors: the Integrator and the Expression actor. The Expression actor specifies the integrand of a differential equation. The output of the Expression actor is connected to the input of an Integrator actor. The output of the Integrator is then connected back to the input of the Expression actor. This loop is then iterated a number of times by the CT Director, numerically integrating the differential equation. The Integrator has one parameter: the initialState, which specifies the initial state of the integrator. Changes to the initialState parameter are ignored after workflow execution starts, unless the initialize() method is called again. The default value of the parameter is 0.0.

Parameters

initialState	The initial state of the integrator. The default value is 0.0.

Ports

output	An output port that broadcasts the "next step" for the integration.
input	An input port that receives the derivative of the actor's output with respect to time.

Interactive Shell (ptolemy.actor.lib.gui.InteractiveShell)

Author: Edward Lee
Version: Unknown

The InteractiveShell actor creates a Unix command shell on the screen, sending commands that are typed by the user to its output port, and reporting strings received at its input by displaying them. Each time the actor fires, it reads the input, displays it, then displays a command prompt, and waits for a command to be typed by the user. The command is terminated by an enter or return character, which then results in the command being produced on the output. The actor is typically preceded by a SampleDelay actor, which provides an initial welcome message or instructions. The InteractiveShell actor's output is routed to some subsystem for processing, and the result is fed back to the input.

Parameters

prompt	An input port that accepts a string that will be used as the prompt in the command shell. To use no prompt (aka, the empty string), create a parameter that has the value "" (for example, foo) and then set the value of the prompt parameter to $foo.

Ports

input	An input port that accepts strings. Each time the actor iterates, it displays any received strings before displaying a command prompt.
output	An output port that broadcasts the command.
prompt	An input port that accepts a string that will be used as the prompt in the command shell. To use no prompt (aka, the empty string), create a parameter that has the value "" (for example, foo) and then set the value of the prompt parameter to $foo.

Interpolator (ptolemy.actor.lib.Interpolator)

Author: Sarah Packman, Yuhong Xiong
Version: Unknown

The Interpolator actor produces an interpolation based on its parameters. The actor receives an array of values, and approximates the values between the specified ones based on an index array that "anchors" each value to the actor's internal iteration count. The values parameter specifies a sequence of values to produce at the output. The indexes parameter specifies when those values should be produced. The values and indexes parameters must both contain arrays of equal lengths; if the lengths are not equal, the actor will generate an error. In addition, the indexes array must be increasing and non-negative. This actor counts iterations. Whenever the iteration count matches an entry in the indexes array, the corresponding entry (at the same position) in the values array is produced at the output. Whenever the iteration count does not match a value in the indexes array, an interpolation of the values is produced at the output. The values are periodic if the period parameter contains a positive value. In this case, the period must be greater than the largest index, and values within the index range 0 to (period-1) are repeated indefinitely. If the period is zero, the values are not periodic, and the values outside the range of the indexes are considered to be 0.0. The order parameter specifies which order of interpolation to apply whenever the iteration count does not match an index in the indexes array. The actor currently supports zero, first, and third order interpolations.

Parameters

order	The order of interpolation for non-index iterations. The actor currently supports zero, first, and third order interpolations.
values	The values that will be produced at the specified indexes. This parameter is an array, and must have the same length as the array specified by the indexes parameter. The default value is {1.0, 0.0}.
indexes	The indexes at which the specified values will be produced. This parameter is an array of integers, and must have the same length as the array specified by the values parameter. The indexes array must be increasing and non-negative. The default value is {0, 1}.
firingCountLimit	The number of iterations that transpire before the actor indicates that it is finished. If firingCountLimit is set to zero, the actor has no limit imposed.
period	The period of the reference values. This parameter must contain an integer token.

Ports

output	An output port that broadcasts a specified value or an interpolation.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Is Present (ptolemy.actor.lib.logic.IsPresent)

Author: Edward A. Lee
Version: Unknown

The IsPresent actor outputs "true" or "false" depending on whether it has received a data token or not. The actor reads tokens of any type via the channels of its input multiport. If a token is present on a channel, the actor outputs the Boolean value true on the corresponding channel of the output port; if a token is not present on a channel, the actor outputs false on the corresponding channel of the output port. The width of the input and output port (i.e., the number of channels on each port) must be the same. Note that this actor is most useful when used with synchronous directors like SDF. Under a PN Director, input is always present (by definition). Under a DE Director, the actor is only triggered if one of the input channels has data.

Ports

output	A multiport that broadcasts Boolean values: true if the corresponding input channel receives a token; false if the corresponding input channel does not receive a token.
input	A multiport that accepts values of any type. The width of the port must match the width of the output ports.

JCOGGridFTP (org.monash.griddles.JCOGGridFTP)

Author: Jagan Kommineni
Version: Unknown

Documentation coming soon.

JCOGPROXYExec (org.monash.griddles. JCOGPROXYExec)

Author: Jagan Kommineni
Version: Unknown

Documentation coming soon.

JCOGWorkflowExec (org.monash.griddles. JCOGWorkflowExec)

Author: Jagan Kommineni
Version: Unknown

Documentation coming soon.

JGridlet Creator (org.monash.griddles.JGridletCreator)

Author: Jagan Kommineni
Version: Unknown

The JGridletCreator actor creates a customizable "gridlet", an actor that can securely connect to a Web service and execute a command at a specified remote location. A Web service is a computer program that runs on a remote host and communicates using a standardized protocol. The JGridletCreator actor saves the customized gridlet to the local Kepler library. Users can customize created gridlets, adding ports and/or parameters, for use with a specific application via the Kepler interface.

Parameters

gridletActorName	The name to give the new gridlet actor.
numberOfInPorts	The number of input ports to give the new gridlet actor.
numberOfOutPorts	The number of output ports to give the new gridlet actor.

JobCreator (org.kepler.actor.job.JobCreator)

Documentation coming soon!

JobGetDirectory (org.kepler.actor.job.JobGetDirectory)

Documentation coming soon!

JobGetRealJobID (org.kepler.actor.job.JobGetRealJobID)

Documentation coming soon!

JobManager (org.kepler.actor.job.JobManager)

Documentation coming soon!

JobRemover (org.kepler.actor.job.JobRemover)

Documentation coming soon!

JobStatus (org.kepler.actor.job.JobStatus)

Documentation coming soon!

Line (ptolemy.vergil.kernel.attributes.LineAttribute)

Author: Edward A. Lee
Version: Unknown

The Line attribute renders a line on the Workflow canvas. Double-click the line to customize its length, width, and color.

Parameters

x	The horizontal extent. The default is 100.0.
lineWidth	The line width. The value is a double that defaults to 1.0.
dashArray	Specify a dash-pattern for dashed or dotted lines. The value consists of an array of doubles that specify the length of the alternating solid and transparent segments. An empty value indicates that the line should not be dashed (the default).
lineColor	The line color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. The default is an opaque black, {0.0, 0.0, 0.0, 1.0}
y	The vertical extent. The default is 0.0.

JobSubmitter (org.kepler.actor.job.JobSubmitter)

Documentation coming soon!

Keyword Group (org.resurgence.moml.KeywordGroup)

Author: unknown **Version:** Unknown	The KeywordGroup actor is a composite actor used inside the GamessInputGenerator actor, which is used for computational chemistry workflows that prepare, run, and display quantum chemical calculations on different compounds. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Parameters

GroupName	
Keywords	The important GAMESS keywords

Ports

inputListIn	An input port that accepts molecular model files that are formatted for use in GAMESS software (e.g., gamin files)
inputListOut	A sequence of file handles belonging to the individual molecules.

Limiter (ptolemy.actor.lib.Limiter)

Author: Edward A. Lee
Version: Unknown

The Limiter actor reads a scalar value and compares it to the top and bottom value of a specified range. The actor outputs a value that falls within the range: The value of the top parameter (if the input is greater than the top value) The input value (if the input falls between the top and bottom value) The bottom value (if the input falls below the bottom value)

Parameters

bottom	The bottom of the limiting range. The value is scalar and default to 0.0.
top	The top of the limiting range. The value is scalar and default to 1.0.

Ports

output	An output port that broadcasts a value that falls within the specified range.
input	An input port that accepts a scalar value.

Line Reader (ptolemy.actor.lib.io.LineReader)

Author: null
Version: Unknown

The LineReader actor reads a file or URL, one line at a time, and outputs each line as a string. Kepler contains several actors used to read and output files in different ways. To read and output a file as a single string, use the FileReader or SimpleFileReader actor. To read and output a specified section of the file, use the SegmentFileReader actor. Use the numberOfLinesToSkip parameter to optionally specify the number of lines to skip at the beginning of the file. When the end of the file is reached, the actor will cease firing. With some directors (such as SDF), the model will stop executing when the actor does. With other directors (such as DE), the actor will cease firing, but the model will continue to execute. To reset the LineReader to start again at the beginning of the file, use a modal model with a reset-enabled transition. For more information about modal models, see the User Manual.

Parameters

fileOrURL	The file name or URL of the file to be read. See FileParameter for more information about specifying file names.
numberOfLinesToSkip	The number of lines to skip at the beginning of the file or URL. 0 is specified by default. The value of this parameter must be non-negative.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
output	An output port that broadcasts a string for each line of the input file.
endOfFile	An output port that indicates whether or not the end of the file has been reached. If the end of the file has been reached, the port will produce a true value and the actor will cease firing. Otherwise, the value is false.

Line Writer (ptolemy.actor.lib.io.LineWriter)

Author: Edward A. Lee
Version: Unknown

The LineWriter actor reads a string token and writes it, one line at a time and without enclosing quotation marks, to a file. The actor has no output. The actor is similar to the FileWriter and TextFileWriter actors, except that the LineWriter does not output the generated file path, while the other two actors do. To write to a file one line at a time with enclosing quotation marks, use the ExpressionWriter actor. Specify whether the input tokens are appended to the specified file (if it exists), or if an existing file is overwritten--with or without confirmation--via the append and confirmOverwrite parameters.

Parameters

append	Specify whether to append the generated file to the existing, specified file. By default, the actor will overwrite any preexisting file.
confirmOverwrite	Specify whether the actor should confirm before overwriting an existing file. By default, the actor will not ask for confirmation.
fileName	The file name to which to write. See FileParameter for more information about specifying file names. The default value is "System.out".

Ports

input	An input port that receives a string token to write--one line at a time--to a file.

LinearModel (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin
Version: Unknown

The LinearModel actor runs a variance or linear regression analysis on its inputs and outputs the result. The actor accepts an independent and a dependent variable. If the independent variable is categorical, the actor uses R to run a variance analysis (or a t-test if the variable has only 2 categories). If the independent variable is continuous, a linear regression is run. The actor outputs both a graphical and textual representation of the analysis. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor performs the linear analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port and/or a Display actor to the results port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory, by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to allow downstream R-actors to retrieve the workspace later in a workflow.

Ports

graphicsFileName	An output port that broadcasts the file name of the generated graphic.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).

results	An output port that broadcasts the textual results of the analysis as an array of integers.
Dependent	An input port that accepts a dependent variable (continuous).
Independent	An input port that accepts an independent variable (continuous or categorical).

Logger (org.kepler.actor.Logger)

Documentation coming soon!

Logic Function (ptolemy.actor.lib.logic.LogicFunction)

Author: Paul Whitaker **Version:** Unknown	The Logic Function actor reads Boolean tokens, performs a specified logical operation (e.g., "and" or "xnor"), and outputs the evaluated result as a Boolean token. Specify a logic function with the function parameter. The following logic functions may be performed: Logic Function Definition and The logical "and" operator (the default) or The logical "or" operator. xorThe logical "xor" operator. nandThe logical "nand" operator. Equivalent to the negation of "and". norThe logical "nor" operator. Equivalent to the negation of "or". xnorThe logical "xnor" operator. Equivalent to the negation of "xor". For more information about these functions, see http://whatis.techtarget.com/definition/0,, sid9_gci213512,00.html. The actor consumes at most one token on each input channel. It does not require that each input channel have a token upon firing. As long as one channel contains a token, the actor will produce output. If no input tokens are available, then no output is produced.

Parameters

function	The logical operation to perform: and, or, xor, nand, nor, xnor. For more information about these functions, see http://whatis.techtarget.com/definition/0,, sid9_gci213512,00.html.

Ports

output	An output port that broadcasts Boolean tokens representing the evaluated input.
input	A multiport that accepts Boolean tokens.

Long To Double (ptolemy.actor.lib.conversions.LongToDouble)

Author: Christopher Hylands
Version: Unknown

The LongToDouble actor converts a long token to a double token. Integers followed by the letter L are long integers (e.g., 1234L), represented by long tokens. A double is a high-precision floating point number (a series of digits that includes a decimal point, such as 3.1416). Note that some loss of precision may be introduced by the type conversion (the least significant digits may be lost).

Ports

input	An input port that accepts data with type long.
output	An output port that broadcasts the converted input as type double.

Lookup Table (ptolemy.actor.lib.LookupTable)

Author: Paul Whitaker and Christopher Hylands
Version: Unknown

The LookupTable actor reads an array of elements and "looks up" an element specified by a given index number. The specified element is output. The first element in the array has an index of 0 (the second 1, and so on). The index must be an integer. If the index is out of range, the actor will output nothing. The LookupTable actor is similar to the ArrayElement actor in function; however, the two actors specify arrays and index numbers in different ways. The LookupTable actor specifies an array with its table parameter and receives an index number via its input port. The ArrayElement actor receives an array via its input port and an index number via a parameter.

Parameters

table	The array from which a value is "looked up" and output. The default value is {0, 1}.

Ports

input	An input port that accepts an array index number. The index must be an integer. If the index is out of range, the actor will output nothing. The first element in the array has an index of 0 (the second 1, and so on).
output	An output port that broadcasts the array element identified by the index number.

MappedLogger (org.kepler.actor.MappedLogger)

Documentation coming soon!

MatlabExpression (org.dart.matlab.MatlabExpression)

Author: Tristan King, Nandita Mangal
Version: Unknown

Kepler's MATLABExpression actor runs a MATLAB function or script and outputs the result of the evaluated script. MATLAB ("MATrix LABoratory") is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. The application is available through Mathworks, http://www.mathworks.com. Please refer to the Mathworks site for information about obtaining and installing MATLAB. The actor will not run unless MATLAB is installed on the local system.

Specify the desired MATLAB expression and configure the appropriate input and output ports. The expression may include references to the input port names, current time (time), and a count of the firing (iteration). To refer to parameters in scope, use $name or ${name} within the expression.

NOTE: You must set an environment variable to the MATLAB libraries directory before running Kepler. The following examples are for MATLAB R2007b installed in a common location:

Mac:
In a terminal window:
```
export DYLD_LIBRARY_PATH=/Applications/MATLAB_R2007b/bin/maci
kepler
```

Windows:
Start->Run
cmd
```
set PATH=%PATH%;c:\Program Files\MATLAB\R2007b\bin\win32
kepler.bat
```

Linux:
In a terminal window:
```
export LD_LIBRARY_PATH=/usr/local/matlab/bin/glnx86
kepler
```

Once your system is properly configured, you can begin to build and run workflows using the MatlabExpression actor.

The actor's get1x1asScalars and getIntegerMatrices parameters control data conversion. get1x1asScalars specifies that all 1x1 matrix results be converted to scalar tokens (the default). Select the getIngegerMatrices parameter to check all double-valued matrix results and return an IntMatrixToken if all elements represent integers. This setting is off by default for performance reasons. To augment the search path used by the MATLAB engine to locate files, set a user-defined parameter named packageDirectories containing a comma-separated list of paths to be prepended to the MATLAB engine search path. Paths can be relative to the directory in which Kepler was started, or any directory listed in the current classpath (in that order, first match wins). After evaluation, the previous search path is restored. Note: to add a new actor parameter, double-click the MatlabExpression actor and click the Add button. Add a _debugging parameter to send debug statements to stdout. An integer value of 1 will return statements from the MATLAB Engine, a value of 2 returns debug statements from both the MATLAB Engine and the Kepler JNI, and a value of 0, or the absence of the parameter, restores the debug behavior to the default setting (off).

Parameters

expression	The MATLAB expression to evaluate. The expression may also be specified via the expression port.
get1x1asScalars	Specify that all 1x1 matrix results be converted to scalar tokens (the default).
getIntegerMatrices	Select the getIngegerMatrices parameter to check all double-valued matrix results and return an IntMatrixToken if all elements represent integers. This setting is off by default for performance reasons.

Ports

output	An output port for the MATLAB results.

Matrix Viewer (ptolemy.actor.lib.gui.MatrixViewer)

Author: Bart Kienhuis and Edward A. Lee
Version: Unknown

The MatrixViewer accepts matrix tokens and displays the received values in a scrollable table format. The actor opens a viewing window to display the matrix.

Parameters

height	The height of the display table in pixels. The value must be an integer. If the table is larger than the specified width, then scrollbars will appear. The default value is 300.
width	The width of the display table in pixels. The value must be an integer. If the table is larger than the specified width, then scrollbars will appear, or the column width will be adjusted. The default value is 500.

Ports

input	An input port that accepts matrix tokens (e.g., [1, 2; 3, 4], which is a 2x2 matrix)

Maximum (ptolemy.actor.lib.Maximum)

Author: Edward A. Lee
Version: Unknown

The Maximum actor reads multiple scalar values and outputs the maximum value. Each time the actor fires, it reads one token from each input channel and outputs the greatest value. The actor also outputs the number of the channel from which it received the maximum value. For complex values, the actor outputs the token with the maximum magnitude.

Ports

input	A multiport that accepts any type of scalar value (e.g., integers, rational, real, or complex numbers).
maximumValue	An output port that broadcasts the value of the maximum scalar token received. The port may be left unconnected.
channelNumber	An output port that broadcasts the number of the channel from which the maximum value was received (e.g., 0 for the first channel or 1 for the second). The port may be left unconnected.

Merge Grids (org.ecoinformatics.seek.gis.java_gis. MergeGrids)

Author: Dan Higgins
Version: Unknown

The MergeGrids actor reads the names of two geospatial image files, merges the files according to a specified merge-operation, and outputs the name of the merged file. The actor can be used to combine several regions into a large region--combining a grid covering North America with one for South America to create a raster grid for the western hemisphere, for example. The grid1Filename and grid2Filename ports each accept a string representing the name of a geospatial raster file. The order of the input files may be significant (e.g., for "subtraction" merge operations). The geographic extent of the output file will always include the combined extent of the inputs. The cell size will match that of the first input file. Select a merge operation with the mergeOperation parameter. Choices include: averageAverage the values of each cell. addAdd the values of each cell. subtractSubtract the values of each cell. Note: the order in which the input files are specified is significant when using the subtract operation. maskMissing values in grid2 will mask the corresponding points in grid1. not_maskMissing values in grid2 will not mask corresponding points in grid1. The MergeGrids actor is similar to the AddGrid actor, except that the MergeGrids can use a variety of merge operations and merges only two grids. The AddGrids actor can be used to merge multiple grids, but uses only addition to merge them.

Parameters

useDisk	Select this parameter to use disk memory for storing grid data. This option in much slower than using only RAM memory (the default) but allows for very large rasters.
mergeOperation	The type of merge to execute. Choices include: averageAverage the values of each cell. addAdd the values of each cell. subtractSubtract the values of each cell. Note: the order in which the input files are specified is significant when using the subtract operation. maskMissing values in grid2 will mask the corresponding points in grid1. not_maskMissing values in grid2 will not mask corresponding points in grid1.

Ports

mergedGridFileResult	An output port that broadcasts the file name of the resulting merged file.
grid2FileName	The second grid file to be merged (in ESRI ASCII Grid format).
grid1FileName	The first raster file to be merged (in ESRI ASCII Grid format).
mergedGridFileName	The file name to give the output file.

Message Digest Test (util.MessageDigestTest)

Author: Dan Higgins **Version:** Unknown	The MessageDigestTest actor compares a file to a previous version to see whether or not the contents have changed. The actor outputs a Boolean value: true if the file is unchanged; false if it differs from the comparison string. The actor uses an MD5 algorithm (a function that is often used to test the integrity of files) to perform the test. Specify a previous file version with the MD5_MessageDigest parameter. The parameter accepts a hex string that the actor will test the input file against. To create a MD5 string, place the actor in "learning mode" by selecting the learningMode parameter. When the actor is in learning mode, it will read the input file and create a comparison string that the actor will store in its MD5_MessageDigest parameter. When the actor is in learning mode, it will output false.

Parameters

learningMode	Select to put the actor in learning mode. When the actor is in learning mode, it will read the input file and create a hex comparison string that the actor will store in its MD5_MessageDigest parameter. When the actor is in learning mode, it will output false.
MD5_MessageDigest	A MD5 hex string that the actor will use to compare against the input file to see if its content has changed. (e.g., 22D62410E268D65B7D59FCFBE3AEFB8C).

Ports

testResult	An output port that broadcasts the result of the test: true if the file is unchanged, or false if it differs from the comparison string.
testFileName	An input port that accepts the name of the file to be tested.

Metadata Source (util.MetadataSource)

Author: Dan Higgins
Version: 1.0

This actor acts as a simple 'Metadata' source for a datafile. It purpose is to feed metadata to an ecogrid writer for submission to the Ecogrid. Metadata (e.g. EML) is entered as a string along with the name of a datafile that the metadata characterizes. There are 2 optional input ports that can be used to input strings that replace the substings '_PARAM1_' and '_PARAM2' in the metadata. This allows things like package title or id to be dynamically changed in a workflow.

Ports

dataFilenameIn	An input port that accepts the file name of the datafile described by the metadata.
parameter1In	First parameter passed to input. The string '_PARAM1_' in the metadata will be replaced by this value if there is a token on this port.
parameter2In	Second parameter passed to input. The string '_PARAM2_' in the metadata will be replaced by this value if there is a token on this port.
dataFilenameOut	An output port that supplies the file name of the datafile described by the metadata.
metadataOut	Supplies string metadata to output port

Minimum (ptolemy.actor.lib.Minimum)

Author: Edward A. Lee **Version:** Unknown	The Minimum actor reads one token from each input channel and outputs the lowest value. Values must be scalar (e.g., integers, rational, real, or complex numbers). The actor also outputs the number of the channel from which it received the minimum value. For complex values, the actor outputs the token with the lowest magnitude.

Ports

minimumValue	An output port that broadcasts the value of the lowest scalar token received. The port may be left unconnected.
input	A multiport that accepts any type of scalar value (e.g., integers, rational, real, or complex numbers).
channelNumber	An output port that broadcasts the number of the channel from which the minimum value was received (e.g., 0 for the first channel or 1 for the second). The port may be left unconnected.

Molecule Array Producer (org.resurgence.moml. MoleculeArrayProducer)

Author: unknown **Version:** Unknown	The MoleculeArrayProducer actor reads a sequence of molecule file names and outputs an array of molecule file names. Double-click the actor to specify the number of molecules to include in the array. The input sequence can be produced by and received from the MoleculeSelector actor. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system.

Parameters

NumberOfMolecules	The number of molecules in the output array.

Ports

moleculeFileHandle	An input port that accepts a sequence of molecule file names.
moleculeNameArray	An output port that broadcasts an array of molecule file names.

Molecule Selector (org.resurgence.moml.MoleculeSelector)

Author: unknown
Version: Unknown

The MoleculeSelector actor is a composite actor used to select chemical modeling data from a local file system. The actor reads a directory of files, and outputs a sequence of files with a specified extension (e.g., cml). Right-click the actor and select Open Actor from the menu to set the required parameters (MoleculeDirectory and MoleculeFileType). The output structures can be passed to an OpenBabel actor, which can convert the files to a new format.

Parameters

MoleculeFileType	The format of the data (e.g., cml)
MoleculeDirectory	The local directory in which the chemical modeling data is stored.

Ports

moleculeFileHandle	An output port that broadcasts a sequence of molecule file handles with the specified extension.

Monitor Image (ptolemy.vergil.actor.lib.MonitorImage)

Author: Edward A. Lee
Version: Unknown

The MonitorImage actor accepts image tokens and displays them on the Workflow canvas. Use the ConvertURLToImage actor to convert an image path or URL into an image token that the MonitorImage actor can read.

Ports

input A multiport that accepts image tokens.

Monitor Value (ptolemy.actor.lib.MonitorValue)

Author: Edward A. Lee
Version: Unknown

The MonitorValue actor displays a received value on the Workflow canvas. The actor accepts tokens of any type and stores the most recently received token in its value parameter, which is displayed on the Workflow canvas.

Parameters

value	The most recent input value. The value is displayed on the Workflow canvas, and has the same type as the input.

Ports

input	A multiport that accepts tokens of any type.

Multiply or Divide (ptolemy.actor.lib.MultiplyDivide)

Author: Edward lee
Version: Unknown

The MultiplyOrDivide actor reads values via its two input ports (multiply and divide), performs the multiply and/or divide operation, and outputs the result. The actor's input ports are multiports, meaning that they can accept multiple inputs. Any values received via the multiply port will be multiplied; any values received via the divide port will be divided. Either port can be left unconnected. If no input tokens are available on the multiply input, then a numerator of one is assumed for division operations. Alternatively, the divide port can be left unconnected to create a simple multiplier. Both of the actor's input ports are polymorphic, accepting data of multiple types (integers, floats, etc). The actor will automatically resolve the input type to the least upper bound of the presented values. The actor may permit the multiply and divide inputs to resolve to types that cannot in fact be multiplied or divided. If the actor resolves an input type into a type that cannot be processed, it will generate an error. The actor outputs the result of the calculation and derives an output type from the input values.

Ports

output	An output port that broadcasts the result of the calculation. The actor derives the output type based on the type of the inputs.
multiply	A muliport that accepts values to be multiplied. The actor automatically infers the input type based on the type of the input values.
divide	A muliport that accepts values to be divided. The actor automatically infers the input type based on the type of the input values.

Next Diagram (org.geon.NextDiagram)

Author: efrat jaeger
Version: Unknown

The NextDiagram actor is a project-specific actor designed to work with the GEON Mineral Classification workflow. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/. The actor receives SVG (Scalable Vector Graphics) rock-naming diagram and transition information as well as mineral composition information, which is retrieved from the Virginia igneous rocks data base. In addition, the actor receives a string representing a specific region in the SVG diagram. The actor uses these inputs to generate a reference to the "next" (more specific) SVG diagram to process. The actor will output a reference to the generated SVG diagram, or the mineral name, if it has been identified.

Ports

rockName	An output port that broadcasts the identified rock name.
transitionTable	An input port that accepts a SVG rock-naming diagram and transition information, as a string. SVG (Scalable Vector Graphics) diagrams are high resolution graphical images that can be scaled without loss. SVG diagrams are stored in the DiagramsTransition actor.
nextDiagram	An output port that broadcasts a reference to the next, more specific, rock-naming SVG diagram. This reference is used by the PolygonDiagramsDataset actor to generate the diagram and output its coordinates.
rowInfo	An input port that accepts the mineral composition information retrieved from the Virginia igneous rocks data base.
region	An input port that accepts a specific region of the SVG diagram, identified by earlier workflow actors.

NexusFileReader (org.cipres.kepler.NexusFileReader)

Author: Zhijie Guan
Version: Unknown

The NexusFileReader actor reads a Nexus file from the local file system and outputs the file content as a string.

Parameters

fileNamePar	The name of the Nexus file to open and read. Click Browse to select a file from the local system.

Ports

inputTrigger	An input port that can receive tokens of any type. The actor will not fire until the port receives an input token. Often, the trigger port receives input from an Initializer actor, which connects to the CIPRES registry.
outputFileContent	An output port that broadcasts the Nexus file content.

Nondeterministic Merge (ptolemy.domains.pn.kernel.NondeterministicMerge)

Author: Edward A. Lee, Haibo Zeng
Version: Unknown

This actor takes any number of input streams and merges them nondeterministically. This actor is intended for use in the PN domain. It is a composite actor that creates its own contents. It contains an instance of PNDirector and one actor for each input channel (it creates these actors automatically when a connection is created to the input multiport). The contained actors are special actors (implemented as an instance of an inner class) that read from the port of this actor and write to the port of this actor. They have no ports of their own. The lifecycle of the contained actors (when they are started or stopped) is handled by the PNDirector in the usual way.

Ports

input	An input multi-port that accepts tokens of any type.
output	An output port for the merged input tokens.
channel	Output port used to indicate which input channel the current output came from. This has type int.

Nonstrict Test (ptolemy.actor.lib.NonStrictTest)

Author: null
Version: Unknown

The NonstrictTest actor compares its input to a known value, and only fires successfully if the two values match. The actor reads an input value of any type and compares it to a corresponding value in a specified array of values. If the two values match, the actor fires successfully. If the two values do not match, the actor generates an error. The comparison value is specified with the correctValues parameter. The parameter accepts an array of values, the type of which must match the type of the input (the default array is {true}). The actor cycles through the array values, comparing each consecutive input to the next token in the correctValues array. After each of the values in the correctValues parameter has been matched, any subsequent iteration always succeeds. This behavior allows the actor to be used as a "power-up" test for a workflow; the actor will check the first few iterations against some known results. If the input is a double or complex token, then the comparison "passes" if the value is close to what it should be (i.e., within the specified tolerance). Tolerance is specified with the tolerance parameter, and defaults to 10-9. Select the trainingMode parameter to collect the input values and place them in the correctValues parameter. The trainingMode parameter is a shared parameter, meaning that changing any one instance of the actor in the workflow will change all instances. To use this actor, place it in a workflow, select trainingMode to collect the reference data, and then unselect trainingMode. Any subsequent run of the actor will generate an error if the input data does not match the training data. Unlike the Test actor, the Nonstrict Test actor does not support a multiport input; only single port inputs may be used. The actor also differs from the Test actor in that it ignores absent inputs.

Parameters

tolerance	A double token specifying how closely the input must match the value from the correctValues array. The default is 10-9.
correctValues	An array specifying what the input should be. The type of the elements must match the type of the input. The default is an array containing a single Boolean value, {true}.
trainingMode	Select the trainingMode parameter to collect the input values and place them in the correctValues array. The trainingMode parameter is a shared parameter, meaning that changing any one instance of the actor in the model will change all instances.

Ports

input	An input port that accepts values of any type. Values will be compared against the values specified by the correctValues parameter. If the input is a double or complex token, then the comparison "passes" if the value is close to what it should be (i.e., within the specified tolerance).

Object To Record Converter (org.ROADnet.ObjectToRecord)

Author: null
Version: Unknown

The ObjectToRecordConverter actor converts an object token to a record. An object token is a data container for an arbitrary Java object (most complex 'things' in Java are objects). These tokens can be used to pass complex Java objects around a Kepler workflow. Object tokens are primarily used for custom workflows with custom actors. A record is a composite data type consisting of one or more elements. Each element is named and can have a distinct type. For example, {number=1, name="dog"} is a record containing two elements. The first element, named "number", contains an integer value. The second element, named "name", contains a string value. The ObjectToRecord actor maps the object fields to record elements. The actor ignores object methods.

Ports

input	An input port that accepts data tokens of type object.
output	An output port that broadcasts the converted data tokens of type record.

Open Babel (org.resurgence.moml.OpenBabel)

Author: unknown **Version:** Unknown	The OpenBabel actor is a composite actor that uses Open Babel software to convert chemical modeling data from one file format to another. The actor reads a chemical structure via its input port and outputs the structure in a specified new format (e.g., SMILES, or MDL MOL). To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. In addition, the Nimrod/G part of the Nimrod toolkit for distributed parametric modeling must be installed as well. Double-click the OpenBabel actor to customize the internal parameters (e.g., the output file type and the output file directory). Input can be passed to the actor by the MoleculeSelector actor, which is used to select molecular structures from a local file system. Open Babel is an application designed to convert file formats used in molecular modeling and computational chemistry. For more information about Open Babel, see http://openbabel.sourceforge.net/api/.

Parameters

OpenBabelBinary	The directory in which the OpenBabel application resides (e.g., /usr/local/bin/babel)
OutputFileDirectory	The local directory into which to place the output files (e.g., /tmp/openbabel)
AdditionalOptions	Additional options to perform (e.g., -t to denote that all input files describe a single molecule). For a list of options, see http://openbabel.sourceforge.net/wiki/Tutorial:Basic_Usage
OutputFileType	The desired output format (e.g., gamin or box). Select a format from the drop-down menu.

Ports

inputFormatHandle	An input port that accepts a sequence of molecule file names.
outputFormatHandle	An output port that broadcasts the file names of the converted files.

Open Database Connection (org.geon.OpenDBConnection)

Author: efrat jaeger **Version:** Unknown	The OpenDatabaseConnection actor opens a database connection using the specified database format and URL, username, and password. Once a database connection has been established, the actor outputs a reference to the connection. Actors downstream in the workflow can use this reference to access the database.

Parameters

password	Database password.
databaseURL	The location of the database (e.g. jdbc:db2://compute1.sdsc.geongrid.org:50000/testpgap)
databaseFormat	The type of database to access (e.g. Oracle, MySQL, etc)
username	Database user name.

Ports

dbcon	An output port that broadcasts a reference to the established database connection. Other actors can use this reference to interact with the database.

Orb Image Source (org.ROADnet.OrbImageSource)

Author: tobin fricke
Version: Unknown

The ORBImageSource actor connects to an Antelope ORB ("Object Ring Buffer") and collects ORB image packets from the specified host and source. The actor "unstuffs" the packets and outputs image tokens. Antelope is a system, originally developed by Boulder Real-Time Technologies (http://www.brtt.com/), for archiving and distributing environmental monitoring information, such as data from a remote camera. Antelope ORBs act as sources (and sinks) for real-time data, such as waveforms and events.

Parameters

srcname	The name of the requested ORB source (e.g., "PF_GVS/MGENC")
orbname	The ORB hostname. The orbname should be specified in the format "hostname: port". Note that "orbnames.pf-style" names are not supported. The value must use a valid IP address or resolvable DNS name and a numeric port number (e.g., "mercali.ucsd.edu:6770").

Ports

output	An output port that broadcasts samples from incoming ORB image packets.

Orb Packet Object Source (org.ROADnet.OrbPacketObjectSource)

Author: tobin fricke
Version: Unknown

The OrbPacketObjectSource actor connects to an Antelope ORB ("Object Ring Buffer") and produces a stream of ORBPacket object tokens that can be used in Kepler. Antelope is a system, originally developed by Boulder Real-Time Technologies (http://www.brtt.com/), for archiving and distributing environmental monitoring information, such as data from a remote camera. Antelope ORBs act as sources (and sinks) for real-time data, such as waveforms and events. The OrbPacketObjectSource actor is similar to the OrbWaveformSource actor, except that the output format is an object token representing the ORBPacket rather than integer sample values.

Parameters

srcname	The name of the requested ORB source (e.g., "PF_GVS/MGENC")
orbname	The ORB hostname. The orbname should be specified in the format "hostname: port". Note that "orbnames.pf-style" names are not supported. The value must use a valid IP address or resolvable DNS name and a numeric port number (e.g., "mercali.ucsd.edu:6770").

Ports

output	An output port that broadcasts ORB Packet-object tokens for use in Kepler workflows.
input	An input port that receives Antelope ORB Packet objects from the specified host and source.

Orb Waveform Sink (org.ROADnet.OrbWaveformSink)

Author: tobin fricke
Version: Unknown

The OrbWaveformSink actor connects to an Antelope ORB ("Object Ring Buffer") host and writes waveform data to it. Antelope is a system, originally developed by Boulder Real-Time Technologies (http://www.brtt.com/), for archiving and distributing environmental monitoring information, such as data from a remote camera. Antelope ORBs act as sources (and sinks) for real-time data, such as waveforms and events. The actor connects to the ORB hostname specified by the orbName parameter and writes to the source specified by the srcName parameter. The actor collects samples until it receives enough samples to form a packet. Then the actor packs the samples into an ORB packet and sends the packet to the ORB host. The number of samples per packet as well as the sample rate are specified with the nSamp and sampRate parameters.

Parameters

samprate	The rate at which samples are updated and entered into the packet.
srcname	The name of the requested ORB source (e.g., "PF_GVS/MGENC")
nsamp	The number of samples per packet.
orbname	The ORB hostname. The orbname should be specified in the format "hostname: port". Note that "orbnames.pf-style" names are not supported. Values must use a valid IP address or resolvable DNS name and a numeric port number (e.g., "mercali.ucsd.edu:6770").

Ports

input	An input port that accepts data samples to be written to the Antelope ORB.
times	The timestamp for the data samples written to the ORB packet.

Orb Waveform Source (org.ROADnet.OrbWaveformSource)

Author: tobin fricke
Version: Unknown

The OrbWaveformSource actor connects to an Antelope ORB ("Object Ring Buffer") and subscribes to an information feed. Antelope is a system, originally developed by Boulder Real-Time Technologies (http://www.brtt.com/), for archiving and distributing environmental monitoring information, such as data from a remote camera. Antelope ORBs act as sources (and sinks) for real-time data, such as waveforms and events. The actor connects to the ORB hostname specified by the orbName parameter and selects the source specified by the srcName parameter. The actor requests ORB packets from the selected source. Samples from the waveform packets are "unstuffed" into waveform data and sent to the actor's output port as integers representing the actual sample values. Samples may be encapsulated as GEN, GENC, or MGENC waveform packets. GEN and GENC packets will always go to channel 0 of the output port. MGENC packets may contain multiple channels of information, and the samples are sent to channels 0, 1, 2, etc of the output port. Use the Select actor to access specific channels.

Parameters

srcname	The name of the requested ORB source (e.g., "PF_GVS/MGENC")
orbname	The ORB hostname. The orbname should be specified in the format "hostname:port". Note that "orbnames.pf-style" names are not supported. Values must use a valid IP address or resolvable DNS name and a numeric port number (e.g., "mercali.ucsd.edu:6770").

Ports

output	An output port that broadcasts samples from incoming waveform packets. Waveforms may be encapsulated as GEN, GENC, or MGENC waveform packets. Multiplexed packets (e.g, MGENC packets), contain multiple channels of information that are mapped to the individual channels of this output port. Use the Select actor to access specific channels.
times	An output port that broadcasts the timestamps for the individual waveform samples. Sample times are computed from the start time contained in each packet and the sample rate. Currently, times is a single port, and the times it outputs correspond to the samples on channel zero of the port.

PAUPInfer (org.cipres.kepler.PAUPInfer)

Author: Zhijie Guan, Alex Borchers **Version:** Unknown	The PaupInfer actor is a CIPRES actor that invokes an external PAUP ("Phylogenic Analysis Using Parsimony") service and returns a generated phylogenetic tree, or an error if the service does not execute correctly. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The PAUPInfer actor also requires PAUP software, and must be used with the Initializer actor. The actor generates a graphical interface that allows users to input information (e. g., a command or working directory) and pass the data to the PAUP application. Once the application has been contacted and has processed the input, the actor returns the generated results, or an error if the application does not execute correctly. If the PAUP service returns correctly, the result of the execution will be output and written to a file in Nexus format. The actor can also be used to monitor error words in the results. Whenever the actor identifies any of the user-specified error words, it will notify the user and ask for further instructions. PAUP (Phylogenic Analysis Using Parsimony) is a tool used to infer phylogenetic relationships. For more information about PAUP, see http:// paup.csit.fsu.edu/ The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Parameters

command	The path to the command file. Alternatively, a command file can be generated and passed to the actor via the user interface generated by the GUIGen XML File. This is an advanced parameter used mainly by programmers.
uiXMLFile	The GUIGen XML file path and name. GUIGen XML files generate a flexible user interface that can be used to input and pass commands to the CIPRES application.This is an advanced parameter used mainly by programmers.
outputFile	The standard output file path (e.g., C:\Documents and Settings \Myself\results.log).
errorFile	The standard error file path (e.g., C:\Documents and Settings\Myself \error.log).
workingDirectory	The working directory of the external CIPRES program.

parameterForOutput	The parameter to return. Usually the parameter is "outfile"
paupCmdFile	The name of the PAUP command file.
iterations	The number of times the PAUP program should iterate.
monitoredErrorWords	Words to monitor. Whenever the actor identifies any of the specified error words, it will notify the user and ask for further instructions.

Ports

standardError	An output port that broadcasts the standard error stream of the execution.
inputParameterValue	An input port that accepts the name of an input file. Input files should contain the data to be analyzed, and may be in NEXUS or tab-delimited format. For a complete list of supported file types, see http://paup.csit.fsu.edu/paupfaq/faq.html.
outputParameterValue	An output port that broadcasts the name of the output results file. Output files are written in Nexus format.
inputParameterName	An input port that accepts the name of the input (usually "infile").
exitCode	An output port that broadcasts the exit code of the execution.
standardOutput	An output port that broadcasts the results of the execution.

Parameter (ptolemy.data.expr.Parameter)

ParameterSet (ptolemy.actor.parameters.ParameterSet)

Author: Christopher Brooks, contributor: Edward A. Lee	An attribute that reads multiple values from a file and sets corresponding parameters in the container. The values are in the form: attributeName = value where variableName is the name of the attribute in a format suitable for {@link ptolemy.kernel.util.NamedObj#setName(String)} (i.e., does not contain periods) and value is the expression in the Ptolemy expression language. Comments are lines that begin with the # character. Each line in the file is interpreted as a separate assignment. The attributes that are created will have the same visibility as parameters of the container of the attribute. They are shadowed, however, by parameters of the container. That is, if the container has a parameter with the same name as one in the parameter set, the one in the container provides the value to any observer. If the file is modified during execution of a model, by default this will not be noticed until the next run. If you set the checkForFileUpdates parameter to true, then on each prefiring of the enclosing opaque composite actor, this parameter will check for updates of the file. Otherwise, it will only check between runs of the model or when the file name or URL gets changed. Note that the order the parameters are created is arbitrary, this is because we read the file in using java.util.Properties.load(), which uses a HashMap to store the properties. We use a Properties.load() because it provides a nice parser for the files and can read and write values in both text and XML.

Parameterized Globus Job (org.sdm.spa.ParameterizedGlobusJob)

Author: no author given
Version: Unknown

The ParameterizedGlobusJob actor runs an executable or command (a "job", specified with an executable path and arguments) on a Globus host, allowing users to take advantage of remote computational resources. The job output is gathered and sent to the output port as a string. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org. A proxy certificate, generated by a GlobusProxy actor, must be passed to the actor via the certificate port. Specify the name of a Globus server with the globusHost parameter, and the path to the remote executable with the executablePath parameter. Arguments can be added to the execution via the argument port.

Parameters

GlobusHost	The name of a Globus host (e.g., griddle.sdsc.edu).
Executable path	The path to the remote executable (e.g., /bin/date or /bin/echo).

Ports

output	An output port that broadcasts the job results as a string.
certificate	An input port that accepts a proxy certificate produced by the GlobusProxy actor.
arguments	An input port that accepts arguments for the job to execute. If the executable is "/bin/echo", the argument could be "Hello World", for example.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Pause (org.sdm.spa.Pause)

Author: Ilkay Altintas **Version:** Unknown	The Pause actor allows users to start and stop a workflow from the Workflow canvas. The actor receives Boolean tokens via its input multiport. When the actor receives a true token, it pauses the workflow and launches a control window that permits a user to restart the workflow when ready. The actor pauses the workflow after the current iteration is complete. The type of director used determines when an iteration is complete. For example, under an SDF Director, if the actor receives a "true" input, then the current iteration is concluded and the execution is stopped. Under a DE Director, if the actor receives a "true" input, concluding the current iteration means processing all events in the event queue with the same time stamp as that event. Thus, actors may be invoked after the Pause actor is invoked with a true input. Under a PN Director, where each actor has its own thread, there is no well-defined notion of an iteration; therefore, the Pause actor may not be the best way to stop a PN workflow. To pause a PN model, design the model so that all actors are starved of data when the model should pause. The director will detect this starvation, and halt the model. Note: if nondeterminism is acceptable, the Pause actor can be used.

Ports

output	An output port that broadcasts a string.
input	A multiport that accepts Boolean tokens. When the port receives a "true" token, the workflow is paused. If the port is not connected, then the actor requests a pause whenever it fires.

PhyloDataReader (org.cipres.kepler.PhyloDataReader)

Author: Zhijie Guan **Version:** Unknown	The PhyloDataReader actor reads phylogenetic data in Nexus format, parses the data, and outputs the data in a form that can be readily used by the CIPRES actors. Data can be read from a Nexus input file, or retrieved from a phylogenetic information database, such as TreeBase. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

outputTree	An output port that broadcasts the tree information contained in the input data. The port type is general.
outputDataMatrix	An output port that broadcasts the data matrix information contained in the input data. The port type is general.
outputTaxaInfo	An output port that broadcasts the taxa information contained in the input data. The port type is general.
inputNexusData	An input port that accepts phylogenetic data in Nexus format. The data can contain information about the phylogenetic tree, data matrix, and/or taxa. Generally this data is read from a Nexus input file, or retrieved from a phylogenetic information database, such as TreeBase. The port type is general.

Point In Polygon (org.geon.PointinPolygon)

Author: Efrat Jaeger **Version:** Unknown	The PointInPolygon actor is a project-specific actor used within the GEON mineral classifier workflow for calculating a classification point given mineral composition and coordinate names. The actor receives a classification point (x,y) and a set of polygons that are defined by an array of points and assigned a region name. The actor determines whether the classification point is contained in one of the input regions and outputs the name of the containing region along with a URL for display purposes. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/.

Ports

toBrowser	An output port that broadcasts a URL that displays the point in the regions.
point	An input port that accepts an array of two doubles representing the specified (x,y) classification point.
region	An output port that broadcasts the name of the polygon region that contains the classification point.
polygonRegions	An input port that accepts a set of polygons and their region names.

Point In Polygon XY (org.geon.PointinPolygonXY)

Author: Efrat Jaeger **Version:** Unknown	The PointInPolygonXY actor is a project-specific actor used within the GEON mineral classifier workflow for calculating a classification point given mineral composition and coordinate names. The actor receives a classification point (x,y) and a set of polygons that are defined by an array of points and assigned a region name. The actor determines whether the classification point is contained in one or more of the input regions and outputs an array of the names of the regions that contain the point. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/.

Ports

point	An input port that accepts an array of two doubles representing the specified (x,y) classification point.
region	An output port that broadcasts an array of the names of the polygon regions that contains the classification point.
polygonRegions	An input port that accepts a set of polygons and their region names.

Polar To Cartesian (ptolemy.actor.lib.conversions. PolarToCartesian)

Author: Michael Leung
Version: Unknown

The PolarToCartesian actor reads two double tokens representing polar coordinates (magnitude and angle) and outputs two new double tokens representing Cartesian coordinates (x and y). Note that the angle input is assumed to be in radians. If either input is NaN or infinity, then the outputs are NaN or infinity. The outputs are a Cartesian representation of the input, where x = magnitude * cos(angle) and y = magnitude * sin (angle).

Ports

x	An output port that broadcasts a double token representing the x-coordinate of the Cartesian pair.
magnitude	An input port that accepts a double token representing the magnitude component of the polar coordinate.
angle	An input port that accepts a double token representing the angle component of the polar coordinate in radians.
y	An output port that broadcasts a double token representing the y-coordinate of the Cartesian pair.

Polar To Complex (ptolemy.actor.lib.conversions.PolarToComplex)

Author: Michael Leung
Version: Unknown

The PolarToComplex actor reads two double tokens representing polar coordinates (magnitude and angle) and outputs a single complex token representing the input. Note that the angle input is assumed to be in radians. If either input is NaN or infinity, then the output is NaN or infinity. The complex token has two parts: the first part corresponds to the real part, which is magnitude * cos(angle), and the second part is the imaginary part, which is magnitude * sin(angle).

Ports

output	An output port that broadcasts a complex token representing the input values.
magnitude	An input port that accepts a double token representing the magnitude component of the polar coordinate.
angle	An input port that accepts a double token representing the angle component of the polar coordinate in radians.

Polygon (ptolemy.vergil.kernel.attributes. ResizablePolygonAttribute)

Author: Edward A. Lee
Version: Unknown

The Polygon attribute renders a polygon on the Workflow canvas. Single-click the polygon to drag the resize handles and adjust the shape's size, or double-click the polygon to customize its height, width, line width, color, and fill color. The vertices parameter is an array of doubles that specify the vertices of the polygon in the form {x1, y1, x2, y2, ... }. The width and height parameters specify the overall width and height. The polygon will be scaled to fit the specified width and height.

Parameters

height	The vertical extent. The value is a double that defaults to 100.0.
lineWidth	The line width. The value is a double that defaults to 1.0.
centered	Indicate whether the shape should be centered on its origin. By default, the location is the upper-left corner.
dashArray	Specify a dash-pattern for dashed or dotted lines. The value consists of an array of doubles that specify the length of the alternating solid and transparent segments. An empty value indicates that the line should not be dashed (the default).
vertices	An array of doubles that specify the vertices of the polygon in the form {x1, y1, x2, y2, ... } The default values specify a rhombus.
width	The horizontal extent. The value is a double that defaults to 100.0.
lineColor	The line color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. The default is an opaque black, {0.0, 0.0, 0.0, 1.0}
fillColor	The fill color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. By default, the value is "none."

Polygon Diagrams Dataset (org.geon.Diagrams)

Author: efrat jaeger
Version: Unknown

The PolygonDiagramsDataset actor is a project-specific actor designed to work with the GEON Mineral Classification workflow. GEON (Geosciences Network) is a distributed infrastructure for geoscience research and education. For more information about GEON, see http://www.geongrid.org/. The actor receives SVG (Scalable Vector Graphics) rock-naming diagram and transition information from the DiagramsTransitions actor, as well as a reference to the "next" diagram, output by a NextDiagram actor. The actor extracts the transitions table and the referenced "next" diagram and its coordinates.

Ports

diagram	An output port that broadcasts a SVG rock-classification diagram.
transitionTable	An output port that broadcasts the transitions between SVG rock-classification diagrams.
coordinateNames	An output port that broadcasts the coordinates of the referenced diagram.
nextDiagram	An input port that accepts a reference to the "next" diagram to be processed. The reference is output by the NextDiagram actor.
diagramTransitions	An input port that accepts SVG rock-naming diagram and transition information from the DiagramsTransitions actor.

Polygon Diagrams Transition (org.geon.DiagramTransitions)

Author: efrat jaeger **Version:** Unknown	The PolygonDiagramTransition actor is a project-specific actor designed to work with the GEON Mineral Classification workflow. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/. The PolygonDiagramTransition actor contains SVG (Scalable Vector Graphics) rock-naming diagrams and the transitions between them. The actor outputs the diagram and transition information to a subworkflow that iterates to use the information to classify a given mineral on increasingly fine descriptive levels.

Ports

index	An output port that broadcasts a reference to the initial diagram to be used by the workflow.
diagramsAndTransitions	An output port that broadcasts SVG rock-naming diagrams and transitions information.

PortParameter (ptolemy.actor.parameters.PortParameter)

Author: null
Version: Unknown

The PortParameter component specifies a persistent value, but can also accept an updated/current value via an associated port. The current value and the persistent value may differ. There are a few situations where PortParameter might not behave as expected: If PortParameter is used in a transparent composite actor, then a token provided to a PortParameter will never be read. A transparent composite actor is one without a director. Workaround: Put a director in the composite actor. Certain actors (such as the Integrator in CT) read parameter values only during initialization. During initialization, a PortParameter can only have a value set via the parameter (it can't have yet received a token). So if the initial value of the Integrator is set to the value of the PortParameter, then it will see only the parameter value, never the value provided via the port. Workaround: Use a RunCompositeActor to contain the model with the Integrator.

Parameters

portParameter The specified persistent value.

PtMatlabExpression (ptolemy.matlab.Expression)

Author: Zoltan Kemenczy and Sean Simmons
Version: null

On each firing send an expression for evaluation to matlab. The expression is any valid matlab expression, e.g.: [out1, out2, ...] = SomeMatlabFunctionOrExpression(in1, in2, ...);... The expression may include references to the input port names, current time (time), and a count of the firing (iteration). This is similar to the Expression actor. To refer to parameters in scope, use $$name or $${name} within the expression. The matlab engine is opened (started) during prefire() by the first matlab Expression actor. Subsequent open()s simply increment a use count. At the start of fire(), clear variables;clear globals commands are sent to matlab to clear its workspace. This helps detect errors where the matlab expression refers to a matlab variable not initialized from the input ports of this actor instance. After the evaluation of the matlab expression is complete, the fire() method iterates through names of output ports and converts matlab variables with corresponding names to Tokens that are sent to the corresponding output ports. Incorrect expressions are usually first detected at this point by not finding the expected variables. If an output port variable is not found in the matlab engine, an exception is thrown. The exception description string contains the last stdout of the matlab engine that usually describes the error. The parameters get1x1asScalars and getIntegerMatrices control data conversion (see below). A Parameter named packageDirectories may be added to this actor to augment the search path of the matlab engine during the firing of this actor. The value of this parameter should evaluate to a StringToken, e.g.: "path1, path2, ..." containing a comma-separated list of paths to be prepended to the matlab engine search path before expression is evaluated. The list may contain paths relative to the directory in which ptolemy was started, or any directory listed in the current classpath (in that order, first match wins). See ptolemy.data. expr.UtilityFunctions#findFile(String). After evaluation, the previous search path is restored. A Parameter named _debugging may be used to turn on debug print statements to stdout from the matlab Engine and the ptmatlab JNI. An IntToken with a value of 1 turns on Engine debug statements, a value of 2 adds ptmatlab debug statements as well. A value of 0 or the absence of the _debugging parameter yields normal operation.

Parameters

expression	The parameter that is evaluated to produce the output. Typically, this parameter evaluates an expression involving the inputs. To refer to parameters in scope within the expression, use $name or ${name}, where "name" is the name of the parameter.
get1x1asScalars	If true, 1x1 matrix results are converted to ScalarTokens instead of a 1x1 MatrixToken, default is true.
getIntegerMatrices	If true, all double-valued matrix results are checked to see if all elements represent integers, and if so, an IntMatrixToken is returned, default is false for performance reasons.

Ports

output

QMView Display (org.resurgence.moml.QmViewDisplay)

Author: unknown **Version:** Unknown	The QMView actor reads a molecular structure via its input port and displays the structure visually. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. QMView is a graphical chemistry program with features including: Viewing of molecular configurations, animation of molecular vibrations, building and comparing structures, viewing of molecular electron density data. For more information about QMView, see http://nbcr.net/software/QMView/

Parameters

QmViewBinary	The path to the QMView application (e.g., /opt/QMView/QMView.new)
QmViewOptions	Additional viewing options. For more information on available options, see http://nbcr.net/software/QMView/

Ports

moleculeFileHandle	An input port that accepts the file name of a molecular structure. Files can be in any format supported by QMView (see http://www.nbcr.net/software/QMView/file.html).

Quantizer (ptolemy.actor.lib.Quantizer)

Author: Jie Liu
Version: Unknown

The Quantizer actor reads a double token and quantizes it according to the specified quantization levels. The actor outputs the quantized version of the input. The quantization levels are specified with the levels parameter, which contains an array of doubles. The elements must be in an increasing order. The default value is {-1.0, 1.0}. Suppose u is the input, and levels = {a, b, c}, where a is less than b, and b is less than c. The output of the actor will be: a, if u is less than or equal to (b+a)/2 b, if (b+a)/2 is less than u, and u is less than or equal to (c+b)/2 c, if u is greater than (c+b)/2 For the default levels {-1.0, 1.0}, the output is (almost) the signum function of the input, or +1.0 if the input is positive, and -1.0 otherwise. If the input is 0, the actor outputs -1.0. The actor does not require that the quantization intervals be equal, i.e., we allow that (c-b) not equal (b-a).

Parameters

levels	The quantization levels. The parameter contains an array of doubles. The elements must be in an increasing order. The default value is {-1.0, 1.0}.

Ports

output	An output port that broadcasts the quantized version of the input (a double token).
input	An input port that accepts a double token.

RExpression (org.ecoinformatics.seek.R.RExpression)

Author: Dan Higgins **Version:** Unknown	The RExpression actor runs an R script or function. Input and output port are created by the user and correspond to R variables used in the specified R script. The actor outputs the result of the evaluated script. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. Kepler contains a number of preconfigured R-actors: Barplot, Boxplot, Scatterplot, Summary, RandomNormal, RandomUniform, Correlation, LinearModel, Regression, RMean, RMedian, RQuantile, and SummaryStatistics.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).

RMean (org.ecoinformatics.seek.R.RExpression)

Author: Dan higgins **Version:** Unknown	The RMean actor accepts an array of values and uses R to calculate their mean. The actor outputs both a graphical and textual representation of the analysis. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor performs the mean-analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port and/or a Display actor to the mean port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

Mean	An output port that broadcasts the calculated mean.
graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Values	An input port that accepts an array of values to evaluate.

RMedian (org.ecoinformatics.seek.R.RExpression)

Author: Dan Higgins
Version: Unknown

The RMedian actor accepts an array of values and uses R to calculate their median. The actor outputs both a graphical and textual representation of the analysis. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor performs the median-analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port and/or a Display actor to the median port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Values	An input port that accepts an array of values to evaluate.
Median	An output port that broadcasts the calculated median.

RQuantile (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin
Version: Unknown

The RQuantile actor accepts an array of values and uses R to produce sample quantiles. The actor outputs both a graphical and textual representation of the analysis. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor accepts an array of values as well as a p-value. The p-value must fall between 0 and 1. The actor performs the quantile-analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port and/or a Display actor to the quantile port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the graph once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

Quantile	An output port that broadcasts the calculated quantile as an array of doubles.
graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Values	An input port that accepts an array of values to evaluate.

P	An input port that accepts a p-value. Specified values must fall between 0 and 1.

Ramp (ptolemy.actor.lib.Ramp)

Author: Yuhong Xiong, Edward A. Lee
Version: Unknown

The Ramp actor is the equivalent of the "for loop" in many traditional computer languages. Its parameters include an intial value, the amount the value is incremented each time the actor fires (the 'step'), and the upper limit of the value (the firingCountLimit). The actor outputs an integer each time it is fired. The actor's output can be used as a counter (increasing or decreasing), or as a parameter used in an incremented workflow. For example, the output of a Ramp actor can be input to an Expression actor to create file names that are unique for each iteration (e.g., 'file1', 'file2', etc.) The Ramp actor is also particularly useful with PN directed workflows where there is no way to set the number of iterations as a Director parameter. The first time the actor fires, it outputs the value of its init parameter. The increment value, specified with either the step port or parameter, will only be added on the following iteration. To add an increment to the first iteration, use the Accumulator actor. The actor accepts values of any type that support addition.

Parameters

firingCountLimit	The number of iterations that transpire before the actor indicates that it is finished. If firingCountLimit is set to zero, the actor has no limit imposed.
step	The amount by which the output is incremented on each iteration. The port accepts a value of any type that supports addition. The default is the integer 1.
init	The value produced by the Ramp on its first iteration. The default value is the integer 0.

Ports

output	An output port that broadcasts the incremented value.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
step	An input port that accepts the amount by which the output is incremented on each iteration. The port accepts a value of any type that supports addition. The value is an integer and defaults to 1.

RandomNormal (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin
Version: Unknown

The RandomNormal actor uses an R-script to generate and output a set of normally (Gaussian) distributed numbers with a mean of 0 and a standard deviation of 1. The actor outputs an array of the generated integers as well as the file path to a graphical representation of the distribution. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R and the R-functions used by this actor (norm and hist), see http://www.r-project.org/. The actor generates the number of random numbers specified via the number port. If the "Graphics Output" parameter is selected (the default), the actor will output a histogram of the generated distribution.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

graphicsFileName	An output port that broadcasts the file name of the histogram representing the generated distribution.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Number	An input port that accepts the number of random numbers to generate.
Dist	An output port that broadcasts the array of random numbers.

RandomUniform (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin **Version:** Unknown	The RandomUniform actor uses an R-script to generate and output a set of uniformly distributed numbers. The actor outputs an array of the generated integers as well as the path to a graphical representation of the distribution. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor generates the number of random numbers specified via the number port. If the "Graphics Output" parameter is selected (the default), the actor will output a histogram of the generated distribution.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
Dist	An output port that broadcasts the array of random numbers.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Number	An input port that accepts the number of random numbers to generate.

ReadTable (org.ecoinformatics.seek.R.RExpression)

Author: Dan Higgins
Version: Unknown

The ReadTable actor reads a text-based data file on the local file system and outputs the data in a format that can be used by other R actors. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. Input data should be in a 'spreadsheet-like' tabular format. Each line of the data file should contain one row of values, separated by a 'separator' delimiter. Saving an Excel spreadsheet as a text file creates such a data file (with a tab separator). The input data is converted into a format that can be manipulated in many ways in R. By default, the first row of the file will be assumed to contain column names (e.g., "Date", "Occurence", etc). The default separator is any white space (e.g., spaces or tabs). Often, all input ports other than the file name can be left unconnected. The R documentation for 'read.table' provides further documentation for various parameters.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that reads the data file from the local system and produces the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

fileName	An input port that accepts the file name of a local data file. Data files must be in a 'spreadsheet-like' tabular format. Each line of the data file should contain one row of values, separated by a 'separator' delimiter. Saving an Excel spreadsheet as a text file creates such a data file (with a tab separator).
separator	An input port that accepts a character used to separate data values. "\t" is a tab; "" is any white space; "," is a comma. Often, this port can be left unconnected.
header	An input port that determines whether or not the output data will contain a header line with column names. Specify true (the default) to use a header line, or false to not. Often, this port can be left unconnected.
nrows	An input port that accepts the number of rows in the data table (if left unconnected, the actor reads to the end of the file). Often, this port can be left unconnected.
fill	An input port that determines whether or not the actor should "fill" missing coluns at the end of a line with empty strings. Specify true to fill empty columns with strings. Often, this port can be left unconnected.
dataframe	An output port that broadcasts the file handle of the transformed data that can be read by other R actors.
graphicsFileName	An output port that broadcasts the file name of the generated graphic.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).

RecIDCM3 (org.cipres.kepler.RecIDCM3)

Author: Zhijie Guan **Version:** Unknown	The RecIDCM3 actor implements a fast algorithmic technique for reconstructing phylogenetic trees. The actor, which invokes a CIPRES CORBA service to generate its results, accepts a data matrix and a phylogenetic tree, and outputs an inferred tree. CORBA services, much like Web services, are computer programs that run on a remote host and communicate using a standardized protocol that allows them to interoperate. CIPRES CORBA services are specifically designed to help analyze phylogenetic data sets. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The RecIDCM3 actor must also be used with the Initializer actor, which connects to the registry of CORBA services used by the CIPRES actors. The actor's input values--the data matrix and tree--can be generated by the PhyloDataReader actor, which reads a Nexus file and outputs the data in a form that can be readily used by the RecIDCM3 actor. The RecIDCM3 service implements the Recursive, Iterative, DCM3 (Disk Coverage Method) algorithm. For more information about RecIDCM3, see http://www.cs.njit.edu/~usman/RecIDCM3.html The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

inputTree	An input port that accepts a CIPRES tree data structure consisting of the tree name, tree score, leaf set, and/or Newick. The tree can be generated by the PhyloDataReader actor, which reads Nexus files and outputs data in a format that CIPRES actors can readily use. The specified tree is used as the initial tree for the RecIDCM3 algorithm.
inputDataMatrix	An input port that accepts a matrix containing character information about the analyzed taxa. The data matrix can be generated by the PhyloDataReader actor, which reads Nexus files and outputs data in a format that CIPRES actors can readily use.
outputTree	An output port that broadcasts the inferred tree in Nexus format.

Record Assembler (ptolemy.actor.lib.RecordAssembler)

Author: Yuhong Xiong
Version: Unknown

The RecordAssembler actor receives tokens of various types on user-defined input ports and assembles those tokens into a record, which it outputs. The actor reads and converts one token from each port every time it fires. A record is a composite data type consisting of one or more elements. Each element is named and can have a distinct type. For example, {number=1, name="dog"} is a record containing two elements. The first element, named "number", contains an integer value. The second element, named "name", contains a string value. The name of each record element generated by the actor is the name of the input port from which the value was received. An actor customized to use three input ports named "day", "week", "month" could output the following record {day="Monday", week=5, month=12}.

Ports

output An output port that broadcasts the generated records.

Record Disassembler (ptolemy.actor.lib. RecordDisassembler)

Author: Yuhong Xiong
Version: Unknown

The RecordDisassembler actor receives a record, which it disassembles and outputs via user-specified output ports. The actor reads and disassembles one record each time it fires. A record is a composite data type consisting of one or more elements. Each element is named and can have a distinct type. For example, {number=1, name="dog"} is a record containing two elements. The first element, named "number", contains an integer value. The second element, named "name", contains a string value. The name of each record element must match the name of a user-specified output port. For example, if the input record is {day="Monday", week=5, month=12}, the output ports must be named "Monday", "week" and "month". If the record contains more fields than the actor has output ports, the extra fields are ignored.

Ports

input An input port that accepts a record, which the actor will disassemble.

RecordUpdater (ptolemy.actor.lib.RecordUpdater)

Author: Yuhong Xiong
Version: Unknown

The RecordUpdater actor updates a record token. The actor receives a record token as well as tokens of various types on user-defined input ports. The actor outputs an updated record that contains the union of the received record and additional tokens. The actor reads and converts one token from each input port every time it fires. A record is a composite data type consisting of one or more elements. Each element is named and can have a distinct type. For example, {number=1, name="dog"} is a record containing two elements. The first element, named "number", contains an integer value. The second element, named "name", contains a string value. The name of each generated record element is the name of the input port from which the value was received. Input record elements can be added to the existing record, or can replace existing record elements. For example, if the input record is {item="oak", height=5}, and the actor contains two additional input ports named "height" (with a value of 10.6) and "age" (with a value of 2), the output record would be {item="oak", height=10.6, age=2}. Note that when an input element is used to update the original record, the output type may be affected. In our example, the type value for "height" is an integer in the original record, but becomes a double in the updated record.

Ports

input	An input port that accepts a record, which the actor will update and output.
output	An output port that broadcasts the updated record

Recorder (ptolemy.actor.lib.Recorder)

Author: Edward A. Lee
Version: Unknown

The Recorder actor receives tokens of any type via its input multiport. By default, the actor stores each token, along with the current time. The Recorder is often used to test configurations of actors, or by programs that invoke Kepler workflows and then query the results after a workflow is run. The capacity parameter limits the size of the record. If the capacity is set to zero, then no tokens are recorded, but the total number of input tokens is counted. If the capacity is 1, then only the most recently seen token on each channel is recorded. If the capacity is negative (the default), then the capacity is infinite.

Parameters

capacity	The size of the record for each channel. The value must be an integer. The default is -1 (meaning that the capacity is infinite).

Ports

input	A multiport that receives tokens of any type.

Rectangle (ptolemy.vergil.kernel.attributes. RectangleAttribute)

Author: Edward A. Lee
Version: Unknown

The Rectangle attribute renders a rectangle on the Workflow canvas. Single-click the rectangle to drag the resize handles and adjust the shape's size, or double-click the rectangle to customize its height, width, line width, color, and fill color.

Parameters

height	The vertical extent. The value is a double that defaults to 100.0.
lineWidth	The line width. The value is a double that defaults to 1.0.
rounding	The amount of rounding of the corners. The value is a double that defaults to 0.0, which indicates no rounding.
centered	Indicate whether the shape should be centered on its origin. By default, the location is the upper-left corner.
dashArray	Specify a dash-pattern for dashed or dotted lines. The value consists of an array of doubles that specify the length of the alternating solid and transparent segments. An empty value indicates that the line should not be dashed (the default).
width	The horizontal extent. The value is a double that defaults to 100.0.
lineColor	The line color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. The default is an opaque black, {0.0, 0.0, 0.0, 1.0}
fillColor	The fill color. Specify a string representing an array of four elements: red, green, blue, and alpha, where alpha is transparency. By default, the value is "none."

Regression (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin
Version: Unknown

The Regression actor uses R to run a variance or linear regression analysis. The actor accepts an independent and a dependent variable. If the independent variable is categorical, the actor uses R to run a variance analysis (or a t-test if the variable has only 2 categories). If the independent variable is continuous, a linear regression is run. The actor outputs both a graphical and textual representation of the analysis. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor performs the analysis and saves the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port and/or a Display actor to the intercept and slope ports.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

slope	An output port that broadcasts the slope of the generated result.
graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).

Dependent	The dependent variable (continuous).
intercept	An output port that broadcasts the y-intercept of the generated result.
Independent	The independent variable (continuous or categorical).

Remainder (ptolemy.actor.lib.Remainder)

Author: Edward Lee **Version:** Unknown	The Remainder actor receives an input value, divides the value by a specified divisor, and outputs the remainder (e.g., the remainder of 5 divided by 2 is 1). The remainder is calculated according to the IEEE 754 standard: the remainder is mathematically equal to f1 - f2 * n, where f1 is the dividend, f2 is the divisor, and n is the integer closest to the quotient f1/f2. If two integers are equally close to f1/f2, then n is the even integer. If n is 0, then its sign is the same as the dividend. Note: If either the dividend or the divisor is NaN, or if the dividend is infinite, or if the divisor is 0, the remainder is NaN. Also: If the dividend is finite and the divisor is infinite, then the remainder is the same as the divider.

Parameters

divisor	The divisor for calculating the remainder. The value is a double that defaults to 1.0.

Ports

input	An input port that receives a number used as a dividend. The value is of type double.
output	An output port that broadcasts the remainder. The remainder is of type double.

Repeat (ptolemy.domains.sdf.lib.Repeat)

Author: Shankar Rao, Steve Neuendorffer
Version: Unknown

The Repeat actor reads and repeats a "block" of tokens (i.e., one or more tokens of any type). The actor outputs the block a user-specified number of times. Specify the block size with the blockSize parameter and the number of times to repeat the block with the numberOfTimes parameter. Note: because the actor reads multiple input tokens at each iteration, it causes a sample rate increase by a factor of numberOfTimes, and hence affects the number of invocations of downstream actors.

Parameters

numberOfTimes	The repetition factor. The value is an integer and must be greater than 0. The default is 2.
blockSize	The number of tokens in a block. The value is an integer and must be greater than zero. The default is 1.

Ports

input	An input port that accepts tokens of any type. Specify the number of tokens to read at each iteration with the blockSize parameter.
output	An output port that broadcasts the block of tokens the specified number of times. Specify the number of times to repeat the output with the numberOfTimes parameter.

RequireVersion (ptolemy.kernel.attributes.RequireVersion)

Author: null
Version: Unknown

RequireVersion specifies a particular version of Ptolemy II (e.g., 6.0-devel). The specified value is compared against the version of the currently executing Ptolemy II installation. If the executing version is less than the value set, an error is generated.

Rescaler (util.Rescaler)

Author: Dan Higgins
Version: Unknown

The Rescaler actor reads a matrix of integers and a specified minimum and maximum value. The actor outputs a matrix of integers that have been rescaled to be within the given minimum and maximum values.

Parameters

max	The desired maximum value to use when rescaling. The value can be specified with either the max port or parameter.
min	The desired minimum value to use when rescaling. The value can be specified with either the min port or parameter.

Ports

output	An output port that broadcasts the matrix of rescaled integers. Note that this is a 1D matrix, even though it is returned as a 2D matrix. Only the first dimension is used.
input	An input port that accepts a matrix of integers to rescale. Note that only the first dimension of the matrix is used. The matrix will only be indexed as int[0][i]

Rician Distribution Random Number Generator

(ptolemy.actor.lib.Rician)

Author: Ye Zhou **Version:** Unknown	The RicianDistributionRandomNumberGenerator actor generates a sequence of random numbers with a Rician distribution. A new random number is output every time the actor fires. The generated values are independent and identically distributed with the mean and the standard deviation given by parameters. The default values of xMean and yMean are both zero, creating a distribution that is also known as a Rayleigh distribution. Hence, the actor is by default a Rayleigh random generator. A Rician random variable is defined as follows: Let Z = sqrt(X2 + Y2), where X and Y are statistically independent Gaussian random variables with means given by parameters xMean and yMean respectively, and common variance given by parameter standardDeviation.

Parameters

seed	The seed that controls the random number generation. A seed of zero (the default) means that the seed is derived from the current system time and a Java hash code (i.e., System.currentTimeMillis() + hashCode()). With extremely high probability, the default seed will ensure that two distinct actors will have distinct seeds. However, current time may not have enough resolution to ensure that two subsequent executions of the same model have distinct seeds. The parameter contains a long token, initially with value 0.
xMean	The mean of the random number along the X-axis. The value is a double that defaults to 0.0.
standardDeviation	The standard deviation of the random number. The value is a double that defaults to 1.0.
yMean	The mean of the random number along the Y-axis. The value is a double that defaults to 0.0.
resetOnEachRun	Select to reset the random number generator each time the workflow is run. By default, the generator does not reset.

Ports

output	An output port that broadcasts the random number. The type is double.

trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Round (ptolemy.actor.lib.conversions.Round)

Author: C Fong **Version:** Unknown	The Round actor rounds a number using a specified rounding function. The actor reads a number of type double (e.g., 102.87), and outputs the rounded integer. The actor can use one of four rounding functions: ceil: Round towards positive infinity. floor: Round towards negative infinity. round: (the default) Round towards nearest integer. truncate: Round towards zero.

Parameters

function	The rounding technique to use: ceil, floor, round, or truncate. The default is round.

Ports

output	An export port that broadcasts the rounded number as an integer.
input	An input port that receives a number of type double.

Run Composite Actor **(ptolemy.actor.lib.hoc.RunCompositeActor)**

Documentation coming soon!

Run Job Grid Client (org.sdm.spa.RunJobGridClient)

Author: Ilkay Altintas
Version: Unknown

The RunJobGridClient actor submits an executable or command (a "job") to a Globus host, allowing users to take advantage of remote computational resources. The job output is gathered and sent to the output port as a string. A proxy certificate, generated by a GlobusProxy actor, must be passed to the actor via the certificate port. Specify the location for the job to run with the GlobusHost parameter. The RunJobGridClient actor generates an RSL ("Resource Specification Language") string for a Globus job, executes the job remotely, and outputs the results and a list of output file paths separated by semi-colons. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org.

Parameters

GlobusHost	The name of a Globus host (e.g., griddle.sdsc.edu).
Program name	The name of the remote service to run
Program version	The version number of the remote service.
Queue name	The name of the job queue.
Number of processors	The number of CPUs in the computer system used for calculations.

Ports

outputFiles	An output port that broadcasts a list of output file paths separated by semi-colons.
result	An output port that broadcasts the job results as a string.
inputFiles	An input port that accepts file names.
certificate	An input port that accepts a proxy certificate produced by the GlobusProxy actor.

SI (ptolemy.data.unit.UnitSystem)

Author: Xiaojun Liu
Version: Unknown

SI (International System of Units) defines a unit system that consists of a set of base and derived units. To view or edit the defined units, right-click the SI icon and select Configure Attribute from the drop-down menu.

SProxy (org.srb.SProxy)

Author: Efrat Jaeger
Version: Unknown

The SProxy actor executes a proxy command on a remote Storage Resource Broker (SRB) system. Only a predefined set of SRB commands can be executed via the SProxy actor. To execute a broader range of commands, use the SRBProxyCommands actor. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. Use the SRBConnect actor to open a connection and create a reference that the SProxy actor can use to access the SRB system. Users must have a valid SRB account to use the SProxy actor. Currently supported commands include: 'list directory', 'copy', 'move', 'remove', 'replicate', 'create directory', 'remove directory', and 'change mode'. COMMANDS DESCRIPTIONS: list directory: List the contents of a remote directory. To use this command, input a path specifying the name of the remote directory to list. The actor outputs the contained file paths as a string, consisting of arrays of path information. Select outputEachFileSeparately to output each file path individually. copy: Copy files to a new location. Directories are recursively copied. To use this command, input a path specifying a remote file path and a newPath providing the location to which files should be copied. The actor outputs the new file paths as a string, which consists of arrays of path information. move: Move files to a new location. Directories are recursively moved. To use this command, input a path specifying a remote file path and a newPath providing the location to which files should be moved. The actor outputs the new file paths as a string, which consists of arrays of path information. remove: Remove files from the remote system. To use this command, input a path specifying the remote file path. Select "forward parent directory (for Srm/Srmdir)" to output a list of the removed files' parent directory paths. replicate: Replicate a file or directory. Unlike copied objects, replicated objects are considered equivalent in SRB (i.e., they are identified by the same identifier). To use this command, input a path specifying the remote file path and a newPath providing the resource (storage system) to which files should be replicated. The actor outputs the new resource path as a string. create directory: Create a new directory. To use this command, input a path specifying the new directory path. The actor outputs the new directory path as a string. remove directory: Remove directories from the SRB system. To use this command, input a path specifying the remote directory. Select "-r (for Srm)" to remove a non-empty directory

	recursively. Select "forward parent directory (for Srm/Srmdir)" to output a list of the parent directory paths of removed directories. change mode: Change a specified user's permission to access a file or directory. To use this command, input a path specifying a remote file or directory, a permission string, the username of the person receiving the permission, and the user's mdasDomain.

Parameters

hasTrigger	Select to activate a trigger input port.
command	The proxy command to perform: list directory, copy, move, remove, replicate, create directory, remove directory, change mode.
_r	Specify whether to remove non-empty directories recursively. This parameter is only relevant for remove commands.
outputEachFileSeparately	Specify whether to output the complete directory list at once or to output each file separately. This parameter is only relevant for list directory commands.
forwardParentDir	Specify whether to output the parent directories of removed files/ directories. This parameter is only relevant for remove and remove directory commands.

Ports

listedFiles	An output port that broadcasts the command results. Output varies depending on the selected command and parameters.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
newPath	An input port that accepts a new file path. Use for copy, move, and replicate commands.
mdasDomain	An input port that accepts a user metadata domain, which contains password information (e.g., ~.srb/.MdasAuth). Specify a mdasDomain when using change mode commands.
exitCode	An output port that broadcasts the execution exit code. (e.g. "Success" if the actor has completed its task).
userName	An input port that accepts a user name of the person to whom to grant permission. Specify when using change mode commands.

SRBFileSystem	An input port that accepts a reference to an SRB file system (e. g., srb.sdsc.edu). Use the SRBConnect actor to open a connection and create the reference.
permission	An input port that accepts file permissions. Specify when using change mode commands.
path	An input port that accepts a path to a remote SRB file or directory (e.g., {"/pzone/home/kepler_dev.sdsc/test"})

SRB Add Metadata (org.srb.SRBAddMD)

Documentation coming soon!

SRB Connect (org.srb.SRBConnect)

Author: Bing Zhu and Efrat Jaeger **Version:** Unknown	The SRBConnect actor allows users to connect to a SDSC Storage Resource Broker (SRB). Users must have a valid SRB account. The actor outputs a reference to the SRB system. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. The SRBConnect actor connects to an SRB file system and returns a reference to the connected system. This connection reference can be propagated to all actors accessing the SRB workspace, allowing the actors to access the SRB system. The SRBConnect actor requires the user to specify account information in the connection parameters (srbHost, srbPort, srbUserName, srbPasswd, srbHomeCollection, srbDomainName, and srbDefaultResource)

Parameters

srbMdasDomainHome	Identity of the site or project (e.g., sdsc)
srbHomeCollection	The user's home collection. Each SRB-registered user has a home collection, where the user may read, write, create-sub collections, and grant access permissions.
srbDefaultResource	The name of a SRB resource system capable of storing data objects (e.g., sfs-tape-sdsc).
srbPasswd	SRB user password
srbUserName	SRB username.
srbHost	The SRB hostname (e.g., srb.sdsc.edu)
srbPort	The port number to connect to on the SRB server (e.g., 7510)

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
SRBFileSystem	An output port that broadcasts a reference to the connected SRB workspace. This connection reference can be propagated to all actors accessing the SRB workspace.

SRB Create Query Conditions (org.srb. SRBCreateQueryConditions)

Author: Efrat Jaeger
Version: Unknown

The SRBCreateQueryConditions actor works in conjunction with the SRBCreateQueryInterface and BrowserUI actors to create a set of user-specified query conditions. These conditions are used to query Storage Resource Broker (SRB) metadata. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. To use the SRBCreateQueryConditions actor, first use the SRBCreateQueryInterface actor to output an HTML form containing metadata attribute names. Use the BrowserUI actor to display the HTML form. Users can select attributes to query, and the BrowserUI actor outputs the selections as XML. This XML output is sent to the SRBCreateQueryConditions actor, which creates an array of string conditions that can be used to query the SRB resource via the SRBQueryMetadata actor.

Ports

xmlConditions	An input port that accepts XML query conditions selected by the user.
conditions	An output port that broadcasts sn array of conditions strings that can be used to query an SRB resource.

SRB Create Query Interface (org.srb. SRBCreateQueryInterface)

Author: Efrat Jaeger
Version: Unknown

The SRBCreateQueryInterface actor works in conjunction with the SRBCreateQueryCondition and BrowserUI actors to create a set of user-specified query conditions. These conditions are used to query Storage Resource Broker (SRB) metadata. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. The SRBCreateQueryInterface actor receives three inputs: a string of metadata attributes, the number of query conditions, and a connection reference to an SRB file system. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SRBCreateQueryInterface actor can use. The SRBCreateQueryInterface actor outputs an HTML form containing metadata attribute names. Use the BrowserUI actor to display the HTML form. Users can select attributes to query, and the BrowserUI actor outputs the selections as XML. This XML output is sent to the SRBCreateQueryConditions actor, which creates an array of string conditions that can be used to query the SRB resource via the SRBQueryMetadata actor.

Parameters

attributes	A list of metadata attribute names for querying. A string of SRB metadata can be returned via an SRBGetMetadata actor.
numberOfConditions	The number of query conditions.

Ports

attributes	An input port that accepts a list of metadata attribute names for querying. SRB Metadata can be returned with an SRBGetMetadata actor.
html	An output port that broadcasts an HTML string consisting of the metadata attributes. Use the BrowserUI actor to display the HTML and allow users to select specific attributes to query.
numberOfConditions	An input port that accepts the number of query conditions.
SRBFileSystem	An input port that accepts a reference to an SRB file system. Use the SRBConnect actor to open a connection and create the reference.

SRB Get Metadata (org.srb.SRBGetMD)

Author: Efrat Jaeger **Version:** Unknown	The SRBGetMetadata actor retrieves user-defined metadata for a Storage Resource Broker (SRB) dataset or collection. The actor outputs the metadata of the specified SRB dataset or collection as a string. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. The SRBGetMetadata actor receives two inputs: an SRB file path, and a connection reference to an SRB file system. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SRBGetMetadata actor can use.

Parameters

recursive	Specify whether or not to return metadata recursively for non-empty directories.

Ports

metadata	An output port that broadcasts the SRB dataset metadata, returned as a string.
srbFilePath	An input port that accepts a path to an SRB dataset or collection
SRBFileSystem	An input port that accepts a reference to an SRB file system. Use the SRBConnect actor to open a connection and create this reference.

SRB Get Physical Location (org.srb. SGetPhysicalLocation)

Author: Efrat Jaeger **Version:** Unknown	The SRBGetPhysicalLocation actor takes the logical path of a Storage Resource Broker (SRB) file and outputs the physical location of the file. The physical location can be used to fetch or stage files without requiring an SRB connection. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). SRB collections use a "logical name space" that maps logical paths consisting of collections (directories) and individual data objects (files) to physical files stored on different devices. Users see and interact with the logical paths, and the physical location is handled by the SRB system and administrators. Use the SRBGetPhysicalLocation actor to "translate" a logical path to a physical one. The SRBGetPhysicalLocation actor receives two inputs: a logical path to a remote SRB file, and a connection reference to an SRB file system. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SRBGetPhysicalLocation actor can use. The actor outputs the physical location of the SRB file as well as an exit code status indicating either success or failure/errors in getting file's physical location.

Ports

logicalPath	An input port that accepts a logical path to an SRB file. Logical paths, which consist of SRB collection and data object information, can be mapped to the physical location of a file.
physicalPath	An output port that broadcasts the path to the physical location of the SRB file.
SRBFileSystem	An input port that accepts a reference to an SRB file system. Use the SRBConnect actor to open a connection and create the reference.

SRB Proxy Commands (org.srb.SRBProxyCommand)

Author: Efrat Jaeger **Version:** Unknown	The SRBProxyCommand actor executes a proxy command on a remote Storage Resource Broker (SRB) system. The actor can execute any command supported by SRB. The actor outputs the result of the command execution as a string, as well as the output file path, if one has been specified. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. For a complete list of supported comands, see http://www.sdsc.edu/srb/index.php/Scommand_Manpages. The SRBProxyCommand actor receives several inputs: a command to be executed (e.g., Smkdir or Sget) along with any command options (-r or -al), a connection reference to an SRB file system, and an optional output file name. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SRBProxyCommand actor can use.

Parameters

outputLineByLine	Select to broadcast the output of the command line by line. The parameter is used to format the output if no outputFile is specified.
commandParameter	The command to execute. For a complete list of commands, see http://www.sdsc.edu/srb/index.php/Scommand_Manpages.
outputFile	The name of the output file (optional). If an output file is specified, the output of the command will be written to that file.
hasTrigger	Select to activate a trigger input port.

Ports

output	An output port that broadcasts the result of the command execution as a string.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
outfileHandle	An output port that broadcasts the output file path, if outputFileName has been specified.

arguments	An input port that accepts the command options. Multiple options are concatenated. For more information about commands and their options, see http://www.sdsc.edu/srb/index.php/Scommand_Manpages
outputFileName	An optional input port that accepts the name of the output file. If an output file is specified, the output of the command will be written to that file.
SRBFileSystem	An input port that accepts a connection reference to an SRB file system. Use the SRBConnect actor to open a connection and create the reference.
command	An input port that accepts the command to execute. For a complete list of commands, see http://www.sdsc.edu/srb/index.php/Scommand_Manpages

SRB Query Metadata (org.srb.SRBQueryMD)

Author: Efrat Jaeger
Version: Unknown

The SRBQueryMetadata actor queries the metadata of remote Storage Resource Broker (SRB) files. Only files that meet the conditions specified in the query are returned. Users must have a valid SRB account. The actor outputs an array of all the file paths satisfying the query conditions. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. The SRBQueryMetadata actor receives several inputs: the path to the SRB data to query, the query conditions, and a connection reference to an SRB file system. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SRBQueryMetadata actor can use. Query conditions must use a particular format, and can be generated using the SRBCreateQueryConditions and SRBCreateQueryInterface. See the documentation of those actors for more information.

Ports

filePaths	An output port that broadcasts an array of the file paths satisfying the query conditions.
srbFilePath	An input port that accepts a path to an SRB dataset or collection.
conditions	An input port that accepts the query conditions that files must meet in order to be returned. Use the SRBCreateQueryConditions and SRBCreateQueryInterface actors to generate these conditions.
SRBFileSystem	An input port that accepts a connection reference to an SRB file system. Use the SRBConnect actor to open a connection and create the reference.

SRB SGet (org.srb.SGet)

Author: Efrat Jaeger **Version:** Unknown	The SRBSGet actor fetches one or more file objects from a remote Storage Resource Broker (SRB) and places the fetched objects in the local file system. Users must have a valid SRB account. The actor outputs an array of all the local file paths as well as an exit status (e.g., "success" or a generated error). SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. The SGet actor receives several inputs: an array of SRB files to get, a local directory to place the returned files, and a connection reference to an SRB file system. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SGet actor can use. Choose to overwrite existing files or append the fetched files to existing files of the same name using the actor's parameters.

Parameters

append	Specify whether to append the generated file to the specified file if it already existts or to overwrite it. By default, the actor will overwrite any preexisting file (after asking for permission).
localDir	The local directory in which to place the fetched files.
confirmOverwrite	Specify whether the actor should confirm before overwriting an existing file. By default, the actor will request confirmation before overwriting a file.

Ports

localDir	An input port that accepts the local directory in which to place the fetched files.
filesToGet	An input port that accepts paths of the SRB files to fetch.
exitCode	An output port that broadcasts the exit status of the operation (e.g., "success" or a generated error).
fetchedFiles	An output port that broadcasts an array of the local file paths of the fetched files.
SRBFileSystem	An input port that accepts a connection reference to an SRB file system. Use the SRBConnect actor to open a connection and create the reference.

SRB SPut (org.srb.SPut)

Documentation coming soon!

SRB Stream Get (org.srb.SRBReader)

Author: Bing Zhu and Efrat Jaeger
Version: Unknown

The SRBStreamGet actor reads one or more file objects from a remote Storage Resource Broker (SRB) and outputs the SRB file as a sequence of bytes. Users must have a valid SRB account. SRB is a Grid storage management system providing data access, transfer, and search functionality, as well as persistent archiving (usually for files). For more information about SRB, see http://www.sdsc.edu/srb/. The SRBStreamGet actor receives several inputs: the name of a remote SRB file to read, and a connection reference to an SRB file system. Use the SRBConnect actor to connect to an SRB file system and return a connection reference that the SRBStreamGet actor can use. The actor outputs the remote file as a sequence of bytes. When the end of the file is reached, the actor will indicate that it has finished by outputting true.

Ports

output	An output port that broadcasts the SRB file as a sequence of bytes.
endOfFile	An output port that indicates whether or not the end of the file has been reached. If the end of the file has been reached, the port will produce a true value. Otherwise, the value is false.
remoteFileName	An input port that accepts the path to the SRB file to read. Files are output as a sequence of bytes.
SRBFileSystem	An input port that accepts a connection reference to an SRB file system. Use the SRBConnect actor to open a connection and create the reference.

SRB Stream Put (org.srb.SRBWriter)

Documentation coming soon!

SSH to Execute (org.sdm.spa.Ssh2Exec)

Author: Norbert Podhorszki **Version:** Unknown	The SSHToExecute actor connects to a remote host, executes commands, and returns their output. The actor will keep the SSH session open until it receives a different username and host. It will return an error if an incorrect identity file is given, if the host is unreachable, if the login is unsuccessful, or if the session dies prematurely. Note: Streaming of output during the command execution is not currently implemented.

Parameters

host	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
user	Username for the remote SSH host.
identity	The file path for the SSH identity file. Specify an identity path to connect to a remote SSH host without having to reenter the password.
streaming mode	Specify whether the output should be sent in a streaming mode. Note: Streaming is not implemented yet.

Ports

host	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
user	Username for the remote SSH host.
identity	The file path for the SSH identity file. Specify an identity path to connect to a remote SSH host without having to reenter the password.
command	The command to be executed on the remote host.
stdout	Output of the command as it would output to the standard shell. If there is an SSH connection related error (or timeout) the value will be an empty string.
stderr	Any errors that were reported by the remote execution or while connecting. If there is an SSH connection related error (or timeout) the value will be an empty string.

returncode	The return code of the execution: 0 (zero) if the execution is not successful, and a positive integer if it is successful.
errors	The string representation of actor execution errors, if any.

SVG Concatenate (org.geon.SVGConcat)

Author: Efrat Jaeger
Version: Unknown

The SVGConcatenate actor concatenates an array of SVG (Scalable Vector Graphics) files. The actor reads an array of SVG file names. Using a specified x- and y-offset, the actor concatenates the files and outputs the path of the concatenated file, which can be displayed by the BrowserDisplay actor. SVG files are high resolution graphical images that can be scaled without loss.

Parameters

outputPathParam	The file name or URL of the output file.
confirmOverwrite	Specify whether to confirm before overwriting an existing output file or not.
shiftY	The y-offset to use during concatenation. By default, the value is 0.0.
shiftX	The x-offset to use during concatenation. By default, the value is 0.0.

Ports

outputPath	An input port that accepts the file path for the concatenated output file. See FileParameter for more information about specifying filenames. The path may also be specified with the svgOutputFile parameter.
output	An output port that broadcasts the URL of the concatenated SVG file.
svgFiles	An input port that accepts an array of strings representing SVG filenames.

SVG To Polygon Converter (org.geon.SVGToPolygon)

Author: Efrat Jaeger **Version:** Unknown	The SVGToPolygon actor is a domain specific actor used within the GEON mineral classifier workflow. The actor converts an SVG file into polygon objects. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/.

Parameters

fileOrURL	The name or URL of the SVG file from which to read.

Ports

output	An output port that broadcasts a polygon object.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

SampleDelay (ptolemy.domains.sdf.lib.SampleDelay)

Author: Steve Neuendorffer, Edward A. Lee
Version: Unknown

The SampleDelay actor outputs a set of tokens that are used as initial input values when a workflow starts up. The actor is used to break dependency cycles created by loops in SDF models. Feedback loops, created when an actor cycles its own output into its input port, can create "deadlock" errors under an SDF Director. The deadlock errors occur because the actor depends on its own output value as an initial input. To fix this problem, use a SampleDelay actor to inject an initial value into the workflow. Specify an array of initial values with the initialOutputs parameter. By default, the actor produces a single integer token, {0}.

Parameters

initialOutputs	The values produced when the workflow is initialized. The value must be an array token, and defaults to an array that contains a single integer token, {0}.

Ports

output	An output port that broadcasts the initial tokens.
input	An input port that accepts tokens of any type.

Scale (ptolemy.actor.lib.Scale)

Author: Edward Lee, Steve Neuendorffer **Version:** Unknown	The Scale actor reads any scalar value that supports multiplication (e.g., an integer, double, array, matrix, etc), and outputs a scaled version of the value. The actor transforms the input based on a scaling factor specified with the factor parameter. For data types where multiplication is not commutative (e.g., matrices), the order of the input value and the factor can be specified with the scaleOnLeft parameter. If scaleOnLeft is set to true (the default) the factor is multiplied on the left and the input on the right. The actor automatically assigns an output data type based on the input and factor types. If the input is an array, the output is also an array, with the elements scaled. If the input is an array of arrays, only the elements of the innermost array will be scaled.

Parameters

factor	The scaling factor. The parameter can contain any scalar value that supports multiplication. The default value is 1.
scaleOnLeft	The multiplication order of the factor and the input. The parameter is useful when working with data types that do not support commutative multiplication (e.g., matrices). If scaleOnLeft is set to true (the default) the factor is multiplied on the left and the input on the right.

Ports

output	An output port that broadcasts the scaled value. The actor automatically assigns a data type based on the type of the input and the factor parameter.
input	An input port that receives a scalar value that supports multiplication (e.g., an integer, double, array, matrix, etc.)

Scatterplot (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin **Version:** Unknown	The Scatterplot actor reads an independent and a dependent variable, which are specified as arrays of values. The actor creates a simple scatter plot based on the input, and outputs the path to the generated graph file. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor saves the generated scatterplot to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

graphicsFileName	An output port that broadcasts the file name of the generated scatterplot.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
Dependent	The dependent variable (continuous). The variable is specified as an array of values.
Independent	The independent variable (continuous or categorical). The variable is specified as an array of values.

ScopeExtendingAttribute (ptolemy.data.expr.ScopeExtendingAttribute)

Author: null **Version:** Unknown	An attribute that extends its container's scope. Any parameter contained by such an attribute has the same visibility as parameters of the container of the attribute.

Select **(ptolemy.actor.lib.Select)**

Author: Edward A. Lee **Version:** Unknown	The Select actor "selects" and outputs a token from among its input tokens. The actor accepts one or more values of any type via its input multiport and selects the value on the channel specified via the control port. The value of the selected channel is output each time the actor iterates. Each time the actor iterates, it checks to see if a token is available on its control port. If so, the actor reads the value and uses it to determine which of its input channels to output next. If an input token is available on the specified channel, then that token is read and sent to the output. If the actor has never received a value on the control port, then the actor will read and output channel zero of the input port. If the value of the most recently received token on the control port is out of range (less than zero, or greater than or equal to the width of the input), then the actor will not fire (although it will continue to consume tokens on the control port). The actor never discards input tokens. Note that in DE workflows, where this actor is commonly used, every time the actor receives a new "select" channel via the control port, the actor will output (at the next firing) all previously unread input tokens on the specified channel in the order in which they arrived.

Ports

output	An output port that broadcasts the selected input channel.
input	A multiport that accepts tokens of any type.
control	An input port that accepts integer tokens that indicate which input channel to output (0, 1, 2, 3, etc). If the value is out of range (less than zero, or greater than or equal to the width of the input), then the actor will not fire.

Sequence Plotter (ptolemy.actor.lib.gui.SequencePlotter)

Author: Edward A. Lee, Bart Kienhuis **Version:** Unknown	The SequencePlotter plots received data and displays the graph. The actor reads one or more sequences of doubles via its input multiport and plots each received sequence as a separate data set. By default, the actor updates the display each time it iterates. Specify the scale and starting position of the x-axis with the xUnit and xInit parameters. By default, the x-axis begins at 0 and the increment between samples is 1. Note: updating the display can be costly in terms of system resources. Specify the number of actor iterations that should pass between display updates with the iterationsPerUpdate parameter. For example, if iterationsPerUpdate = 2, then every second time the actor fires, it will update the display (i.e., the actor will update its display on the first firing, the third, the fifth, etc.) The actor will consume its inputs on every firing, regardless of the value of the iterationsPerUpdate parameter.

Parameters

xInit	The start point of the X-axis.
fillOnWrapup	Specify whether or not to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
startingDataset	The starting dataset number to which data is plotted. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).
iterationsPerUpdate	The number of actor iterations that should pass between display updates. The value must be a non-negative integer. The default value is 1.
xUnit	The increment along the X-axis.

Ports

input	A multiport that accepts one or more sequences of doubles.

Sequence To Array (ptolemy.domains.sdf.lib.SequenceToArray)

Author: Yuhong Xiong
Version: Unknown

The SequenceToArray actor reads a sequence of elements and bundles them into a single array, which the actor outputs. The actor can accept inputs of any one type (int, double, etc.) The type must be consistent. The number of elements to be bundled is specified by the arrayLength parameter or port.

Parameters

arrayLength	The size of the output array. The default is the integer 1.

Ports

input	An input port that accepts a sequence of input elements. Elements may be of any one type (int, double, etc). The type must be consistent.
arrayLength	An input port that accepts the size of the output array. The default is the integer 1.
output	An output port that broadcasts an assembled array. The type of the array elements matches that of the input values.

Server Execute (org.geon.ServerExecute)

Documentation coming soon!

ShowLocations (util.ShowLocations)

Author: Dan Higgins **Version:** Unknown	The ShowLocations actor reads a set of points specified in an input text file, and plots them to a specified background image/map file. The actor displays the generated image on the screen. The actor uses the ImageJ application to create the location map, which is an image raster file that ImageJ can display and process. For more information about ImageJ, see http://rsb.info.nih.gov/ij/. The actor accepts input points via an input text file (specified by a file name). A single point location is given per line with the x and y values separated by either a space or a tab. The background map is specified via the fileOrURL parameter. Specify the x and y values of the upper-left corner of the background image, and a multiplicative scale factor to map the point coordinates to the background.

Parameters

X_upperleft	The x-value of the upper-left corner of the background image.
Y_upperleft	The y-value of the upper-left corner of the background image.
fileOrURL	The file name or URL of the background map or image. The value is a string representing the file path of the background image. The input points will be plotted on top of this image.
scale_factor	The scale factor. (x,y) input points are plotted at the pixel nearest the (x-pos, y-pos) where: x-pos = (x-X_upperleft)*scale_factor and y-pos = (y-Y_upperleft) *scale_factor
macroString	The ImageJ macro to execute. The macro may include the expression "_FILE_", which will be replaced by the path of the specified image file. x and y data from the input file will replace "_XPOINTS_" and "_YPOINTS_"

Ports

input	A multiport that accepts a string representing the path to an input text file. The input file contains a space- or tab-delimited set of x and y values (one set per line), which will be plotted by the actor.

Simple File Reader (org.resurgence.actor. SimpleFileReader)

Author: Wibke Sudholt
Version: Unknown

The SimpleFileReader reads and outputs the contents of a file as a single string. The actor is similar to the FileReader, except that the SimpleFileReader can only take its input from another workflow component via an input port, whereas the FileReader actor can use either a port or parameter.

Ports

content	An output port that broadcasts a string of the specified file contents.
file	An input port that accepts the file name or URL of the file to read and output. The file name is passed as a string from another workflow component. Pass a file name with a StringConstant actor, for example.

Sinewave (ptolemy.actor.lib.Sinewave)

Author: Edward A. Lee
Version: Unknown

The Sinewave actor is a composite actor that generates a sine wave. Each time the actor iterates, it outputs a sample from the generated sinusoidal signal. Output values are uniformly sampled.

Parameters

phase	The sinusoid phase in radians. The default is 0.0. The phase may also be specified via the phase port.
samplingFrequency	The sampling frequency in cycles per second (Hertz), a double with default 8000.0.
frequency	The sinusoid frequency in cycles per second (Hertz). The default is 440.0. The frequency may also be specified via the frequency port.

Ports

phase	An input port that accepts a double token representing sinusoid phase in radians. The default is 0.0. The phase may also be specified with the phase parameter.
output	An output port that broadcasts a uniformly sampled sinusoidal signal.
frequency	An input port that accepts a double token representing the sinusoid frequency in cycles per second (Hertz). The default is 440.0. The frequency may also be specified with the frequency parameter.

SoaplabAnalysis (org.sdm.spa.SoaplabAnalysis)

Author: Nandita Mangal **Version:** Unknown	The SoaplabAnalysis actor executes a standard Soaplab operation using a client created by the SoaplabServiceStarter actor and "set operations" specified by a SoaplabChooseOperation actor. The actor outputs a Soaplab client used by the SoaplabChooseResult actor, which specifies which results to return and display. Soaplab is a set of Web services providing access to (mainly) data analysis applications on remote computers. The Soaplab actors require an EBI-registered Web Service Description Language (WSDL) file. WSDL is a format for describing network services--from simple eBay watcher services to complex distributed applications. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services. The actor receives a configured Soaplab client from the SoaplabServiceStarter actor. From the SoaplabAnalysis parameters, select a standard Soaplab operation (e.g., run) from the drop-down menu beside the soaplabMethodName parameter. Standard operations are present in every Soaplab service. Alternatively, enter a non-standard Soaplab method using the "OR Enter another Soaplab Method" parameter. The actor outputs a Soaplab client that is used by the ChooseResultsOperation actor to examine the Soaplab results and specify which results to return.

Parameters

soaplabMethodName	A standard Soaplab operation (e.g., run) to perform. Standard operations are present in every Soaplab service.
OR Enter another Soaplab Method	An alternative Soaplab operation to perform. Operations must be defined by the Soaplab service via the WSDL file.

Ports

clientInput	An input port that accepts information about the Soaplab client and its initial inputs. This information is passed to the actor by the SoaplabServiceStarter actor.

clientOutput	An output port that broadcasts an object representing the Soaplab results. Use the ChooseResultsOperation actor to examine the results and specify the results to return.

SoaplabChooseOperation (org.sdm.spa. SoaplabChooseOperation)

Author: Nandita Mangal **Version:** Unknown	The SoaplabChooseOperation actor specifies the value of a Web service input (e.g., an input sequence) in a format that the Soaplab service can use. The actor accepts the value of the input along with the name of a predefined "set operation" (e.g., set_sequence_data) used by the service. The actor outputs a Soaplab client object. Soaplab is a set of Web services providing access to (mainly) data analysis applications on remote computers. The Soaplab actors require an EBI-registered Web Service Description Language (WSDL) file. WSDL is a format for describing network services--from simple eBay watcher services to complex distributed applications. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services. Specify a Soaplab input (e.g., a format string) via the actor's input port (use a String Constant or Constant actor to create the initial value). In the SoaplabServiceStarter parameters, specify an EBI-registered WSDL via the wsdlUrl parameter and click Commit. When the actor is next opened, the inputSetMethods parameter will contain a drop-down menu containing a complete list of "set operations" used to specify input values so that the Soaplab service can use them. Select an operation (e.g., set_format) and click Commit.

Parameters

wsdlUrl	An EBI-registered Web Service Description Language (WSDL) file. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services.
inputSetMethods	An operation used to set the value of a Soaplab service input.

Ports

input	An input port that accepts a value to pass to the Soaplab service via the inputSetMethod parameter.
output	An output port that broadcasts a Soaplab client. Use the SoaplabServiceStarter to start the client and create an empty job used by the workflow.

SoaplabChooseResultType (org.sdm.spa. SoaplabChooseResultType)

Author: Nandita Mangal **Version:** Unknown	The SoaplabChooseResultType actor specifies a "get operation" used to retrieve and display Soaplab results. The actor receives a Soaplab service client from the SoaplabAnalysis actor. The actor outputs the specified Soaplab results as a string. Soaplab is a set of Web services providing access to (mainly) data analysis applications on remote computers. The Soaplab actors require an EBI-registered Web Service Description Language (WSDL) file. WSDL is a format for describing network services--from simple eBay watcher services to complex distributed applications. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services. The actor receives a Soaplab client from the SoaplabAnalysis actor. In the SoaplabChooseResultType parameters, specify an EBI-registered WSDL via the wsdlUrl parameter and click Commit. When the actor is next opened, the outputGetMethods parameter will contain a drop-down menu containing a complete list of "get operations" used to return and display Soaplab results. Select an operation (e.g., get_report) and click Commit.

Parameters

wsdlUrl	An EBI-registered Web Service Description Language (WSDL) file. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services
outputGetMethods	An operation used to return and display Soaplab results.

Ports

input	An input port that accepts a configured Soaplab client created and passed to the actor by the SoaplabAnalysis actor.
output	An output port that broadcasts the returned Soaplab results, output as a string.

SoaplabServiceStarter (org.sdm.spa. SoaplabServiceStarter)

Author: Nandita Mangal
Version: Unknown

The SoaplabServiceStarter actor starts a Soaplab client and creates an empty job used by a Soaplab workflow. The actor receives a Soaplab service client created by the SoaplabChooseOperation actor, and outputs a started Soaplab client, used by the SoaplabAnalysis actor to run specified operations. Soaplab is a set of Web services providing access to (mainly) data analysis applications on remote computers. The Soaplab actors require an EBI-registered Web Service Description Language (WSDL) file. WSDL is a format for describing network services--from simple eBay watcher services to complex distributed applications. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services. The actor receives a configured Soaplab client from the SoaplabChooseOperation actor. In the SoaplabStartService parameters, specify an EBI-registered WSDL in the wsdlUrl parameter and click Commit. The actor will output a Web service client object used by the SoaplabAnalysis actor to run specified operations.

Parameters

wsdlUrl	An EBI-registered Web Service Description Language (WSDL) file. For a complete list of registered EBI-registered WSDLs, see http://www.ebi.ac.uk/soaplab/services.

Ports

setOperation1	An input port that accepts a Soaplab client configured with a "set operation." The set operations are specified and passed to the actor by the SoaplabChooseOperation actor.
output	An output port that broadcasts a Soaplab client used by the SoaplabAnalysis actor to perform a Soaplab operation.

SSH Directory Creator (org.kepler.actor.ssh. DirectoryCreator)

Author: Norbert Podhorszki **Version:** Unknown	The SSHDirectoryCreator actor creates a local or remote directory. The actor reads the path to a target machine as well as the name of the directory to create. The actor outputs the operation results: true if the directory is created successfully, and false if not. In addition, the actor outputs the text of generated error messages, if any. For remote operations, a relative path is relative to the home directory. For local operations, a relative path is relative to the current directory. If the 'parent' flag is set, intermediate directories in the provided path will be created if necessary.

Parameters

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with ssh. This parameter is read once at initialize.
dir	The path to the directory to be read on the target machines. This parameter is read once at initialize.
parent	Specify whether parent directories should be created recursively if necessary. If the 'parent' flag is set the operation is considered successful even if the directory already exists.

Ports

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. This parameter is read once at initialize.
dir	The path to the target directory. This parameter is read once at initialize.
succ	A Boolean token: true if the operation is successful, false if not.
error	The actor's execution errors, if any; otherwise an empty string.

SSH Directory Listing (org.kepler.actor.io.SshDirectoryList_v1_1)

Author: Norbert Podhorszki
Version: 1.1 Feb. 2008

The SSHDirectoryListing actor gets a directory listing for a local or remote directory. If the directory is remote, the actor uses SSH protocol. The actor returns an array of new files (new since the last time the actor was triggered), new and modified files, all files, or only files that match a specified mask. Each element in the array is a record consisting of a file name, size, and date. If 'newFilesOnly' is set (the default) the file list is updated each time the actor is triggered and only 'new' filenames are output. 'New' files are determined by the difference between the current and the previous directory listings. Files not found in the previous listing are considered new. If 'newFilesOnly' is not set, the entire directory list is returned each time the actor is triggered. Select the 'checkSizeAndDate' parameter to return modified files in addition to new ones. If this parameter is selected, the actor will return all modified files (i.e., files where the size or date has changed between the previous and current listings) in addition to the new files. If 'checkSizeAndDate' is not set (the default), only brand new files are listed. This flag can be used for watching a specific file to look for changes in size or access time. Directory listings are output as an array. Each element in the array is a record containing a file name, size and date (e.g., {name=String, size=long, date=long}). File size is in bytes and date is in UTC seconds. The resolution depends on the 'ls -l' output, i.e. at most a minute, and for old files a day. In the local case, resolution depends on the resolution of Java and the local OS, usually less than a millisecond. The 'sendEmpty' parameter specifies if this actor should emit an empty ArrayToken of RecordTokens in case no files are found. Very useful option under directors that expect a fixed token rate. The 'delay' parameter specifies if the actor should postpone the emission of lists by one step, i.e., emit the results of the previous listing and store the recent one for the next time. 'delay' and 'sendEmpty' cannot be both true; 'sendEmpty' will be considered false if 'delay' is true. This is a strange option. It is used to safely watch a simulation by not listing the newest files which may be still under construction by the simulation. The assumption here is that if we find a new set of files with the recent listing, the "new" files in the previous listing are completely finished by the simulation. Note that the last set of files will not be emitted, except if 'frequencySec' and 'stopmask' are used, see below. The 'frequencySec' and 'stopmask' parameters can be used to make this actor run in a loop itself. If frequencySec is non-zero, the actor will list the directory with the time

	interval of frequencySec until a file matching the stop mask is found. The stop file(s) will be emitted as a separate token after the token of newest files. The actor does not finalize after that, so it can be fired again to start a new file watching loop. Use only in PN, since this mode can block other directors. If 'delayed' is true, the last listing is know: when a stop file is found. In this case, both the previous listing and the current listing and moreover, the stop file(s) will be emitted in the firing, i.e. three tokens are produced.

Parameters

newFilesOnly	Specify whether the output should contain only new files (files not found in previous listing) or all files. By default, only new files are listed.
checkSizeAndDate	Specify whether the output should contain modified files as well as new files. By default, modified files are not included.
sendEmpty	Specify whether an empty ArrayToken (of RecordTokens) should be emitted if no files found. Useful in SDF which awaits a result.
delay	Delay the listings by one step, i.e. emit the list found in the revious step while storing the recent list for the next firing. If true, sendEmpty is considered to be false.
frequencySec	If non-zero, the actor runs in a loop and lists the directory regularly with the given time interval. The actor will stop only when a file is found matching the stop mask. The stop mask usually should not be included in the list of masks. Use only in PN

Ports

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH.
dir	The path of the directory to be read on the target machine.
masks	The list of file masks. Only files and directories that match one of the masks will be listed. "" means all files.
trigger	The actor's trigger port. You need to pass a token to the trigger port to trigger the actor only if you do not connect any other port.
newFiles	The output array of files (new files by default). This port is an output port of type ArrayToken. Each element is a record containing the name of the file, its size and date, e.g., {name=String, size=long, date=long}.
stopmask	The "stop" file mask. This parameter is used only if frequencySec is set to be greater than zero. Then, the actor performs the listing in a loop until a file matching the stopmask is found. Use this mode only in PN, because the actor would block all other models.

SSH Execute cmd (org.kepler.actor.ssh.ExecuteCmd)

Author: Norbert Podhorszki
Version: 1.1 March 2008

The SSHExecuteCmd actor connects to a remote host using SSH protocol and executes a command. If the host is an empty string or "local", the Java Runtime will be used for execution instead of SSH. After the command terminates, the actor returns the stdout, stderr, and exit code of the operation. Set the timeoutSeconds parameter to terminate the command after a specified amount of time. Specify 0 to wait indefinitely for command termination. If connecting to a Unix host, use the cleanupAfterError Parameter to kill the remote process (and all of its children) after an error or timeout. Set this parameter only if connecting to a Unix server (other platforms do not support it). Note: Streaming of output during the command execution is not currently implemented. Third party operation. If the remote command is expected to ask for a password (or passphrase when connecting to a remote host with public-key authentication) set the port/parameter thirdParty for the user@host:port of that third party (it can be the same as target if a sudo command is executed). The authentication to the third party should be the same from the target host and from Kepler's local host. Kepler authenticates (by opening a channel) to the third party and then it provides the password/passphrase used for the authentication to the command on the target host. Therefore, this actor cannot be used to reach a remote host through a proxy machine and execute a command there. The third party execution can be used e.g. to execute and ssh/scp command that connects to another host, also reachable from Kepler's host, to execute external data transfer commands (bbcp, GridFTP, SRM-Lite etc) or sudo commands. The actor will first authenticate Kepler to the third party host (if not yet done by other actors, e.g. SshSession). During the execution of the command, it looks for the appearance of the string 'password' or 'passphrase' in the stdout/stderr streams (case-insensitively). If such string is found, it writes the authentication code stored within Kepler used for the authentication. Therefore, the command must read the password on the standard input, not directly from the terminal device. This process is performed only once! The underlying java code does not have pseudo-terminal emulation, so if you cannot force the command to read passwords from the stdin (e.g. scp command), you have to use an external tool to execute the command through a pseudo-terminal. ptyexec is provided in the org.kepler.ssh package, a C program, that should be compiled and put into the path on the target machine. Then you can execute "ptyexec scp ...". It works on Linux only.

Parameters

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
thirdParty	The third party machine where the command wants to connect and needs to authenticate. Kepler can be used to do the authentication and then provide the password or private-key passphrase to the command. It should be defined as [user@]host[:port].
timeoutSeconds	Timeout in seconds. The command will be timed out after the specified amount of time. Specify 0 to wait indefinitely for command termination.
cleanupAfterError	Kill remote process(es) after an error or timeout. Set this parameter only if connecting to a Unix server (other platforms do not support it). All processes belonging to the same group as the remote command (i.e., its children) will be killed.
streaming mode	Specify whether the output should be sent in a streaming mode. Note: Streaming is not currently implemented.

Ports

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
command	The command to be executed on the remote host, specified as a string (e.g., "uname -a; date")
stdout	Output of the command as it would output to the standard shell. If there is an SSH connection related error (or timeout) the value will be an empty string.
stderr	Any errors that were reported by the remote execution or while connecting. If there is an SSH connection related error (or timeout) the value will be an empty string.
exitcode	The exit code of the command. If there is an SSH connection related error (or timeout) the exitcode will be -32767
errors	The string representation of actor execution errors, if any.

SSH File Copier (org.kepler.actor.ssh.FileCopier)

Author: Norbert Podhorszki
Version: Unknown

The SSHFileCopier actor connects to a remote host using SSH protocol and copies a file or directory to or from the host. Either the source or the target file must be local. This actor cannot copy remote files to remote places. For such operations, use the SSHExecuteCmd actor. The actor performs the copy operation and outputs the operation results (success or failure, as well as internal error messages, if any). The file references should be in the format: [[user@]host:]path. For example, foo.txt The file foo.txt in the current directory on the local machine playdir/foo.txt The relative path to the current dir on the local machine /home/littleboy/ playdir/foo.txt The absolute path to foo.txt on the local machine local: playdir/foo.txt The relative path to $HOME on the local machine localhost:playdir/foo.txt The relative path to $HOME on the 'localhost' machine john@farmachine:playdir/foo.txt The relative path to $HOME on the 'farmachine' machine of user 'john' If the target is an existing file, it will be overwritten. If the target is an existing directory, a subdirectory within the existing directory will be created with the name of the source. If copying a directory, set the parameter 'recursive' to true to copy the entire directory.

Parameters

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
source	The location of the files and directories to copy.
recursive	Specify whether directories can be copied recursively.

Ports

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
source	The location of the files and directories to copy.

succ	A Boolean token indicating if the copy operation was successful: true if ALL matched files and directories are copied; false if not. Note: if copying directories and 'recursive' is not set, the value will be false (files will be copied but directories will not be).
error	The actor's execution errors, if any; otherwise an empty string.

SSH File Remover (org.kepler.actor.ssh.FileRemover)

Author: Norbert Podhorszki **Version:** Unknown	The SSHFileRemover actor connects to a local or remote host and deletes the specified files/directories. If the host is a remote machine, the actor will connect using SSH protocol. If the host is local, all commands will be executed locally using Java Runtime. A file mask indicates which files should be deleted. The actor performs the delete operation and outputs the operation results: success or failure, as well as internal error messages, if any. The file mask identifies which files and directories to remove. Note that the recursive parameter must be selected when removing directories. Relative paths are relative to the home directory in case of remote operations and relative to the current directory in case of local operations. A file mask can contain wildcards. To use wildcards in the mask file, select the allowMask parameter. Note that symbolic links will also be deleted, but if they refer to a directory, they are not followed.

Parameters

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
mask	File mask as string (e.g., "/path/test.txt" or "/path"). Only files and directories that match the mask will be removed. Path expressions are allowed. To use wildcards in the mask, select the allowMask parameter.
recursive	Specify whether directories should be removed recursively.
allowMask	Specify whether wildcards (e.g., an asterisk or question mark) are allowed in the mask.

Ports

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
mask	File mask as string (e.g., "/path/test.txt" or "/path"). Only files and directories that match the mask will be removed. Path expressions are allowed. To use wildcards in the mask, select the allowMask parameter.

succ	A Boolean token indicating if the copy operation was successful: true if ALL matched files and directories are copied; false if not. Note: if copying directories and 'recursive' is not set, the value will be false (files will be copied but directories will not be).
error	The actor's execution errors, if any; otherwise an empty string.

SSH Session (org.kepler.actor.ssh.SshSession)

Author: Norbert Podhorszki **Version:** Unknown	The SSHSession actor creates an SSH session to a remote host. If requested, the session will not be opened until the first actor that uses SSH is invoked by the workflow (e.g., an SSHExecuteCmd actor). This actor is used to provide a private-key for public-key authentication, and to connect to a remote machine at a certain point in the workflow (e.g., at the very beginning so that a password is specified then rather than during execution). The actor outputs a connection reference (the name of the host target), which is retrieved by other actors so that they can use the connection. If the target is an empty string or "local", the actor does nothing, and all commands (in other SSH actors) will be executed locally using Java Runtime. If the parameter postpone is true, the connection is not established until the first remote operation is executed. However, one of the main purposes of this actor is to request passwords and make a connection at the beginning of the workflow. The default is false. If the parameter closeAtEnd is true, the session will be closed at the end of the workflow. If false, the session will remain open. The default is false, as the underlying SSH package has only one session (within Kepler) to a given user@host:port. If two workflows are executing on the same host, and the SSH session is closed at the end of one of the workflows, the other will likely experience a broken operation. The failed output port emits a BooleanToken indicating whether the connection failed: 'true' if the postpone flag is false and the connection failed, otherwise, 'false'. The port is used to throw an exception or stop a workflow that cannot run without a connection, or to successively try out other hosts.

Parameters

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
identity	The file path for the SSH identity file. Specify an identity path to connect to an SSH host without having to reenter the password.
postpone	Specify whether to postpone the connection until the first remote operation is executed. The default is false, as one of the main purposes of this actor is to request passwords and make a connection at the beginning of the workflow.

closeAtEnd	Specify whether the connection to the host should be closed when the workflow terminates. If you run more than one workflow simultaneously on the same remote host, this flag should be false to avoid closing the session in one workflow while the others are still using it.

Ports

target	The machine to be used at job submission. It should be null, "" or "local" for the local machine or [user@]host to denote a remote machine accessible with SSH. If user is not provided, the local username will be used. If port is not provided, the default port 22 will be applied. This parameter is read once at initialize.
identity	The file path for the SSH identity file. Specify an identity path to connect to an SSH host without having to reenter the password.
target_out	A string that identifies the created session. The value is identical to the value of the specified target. If the target is specified as a parameter, you do not need to connect this port to pass the connection reference to downstream actors.
failed	A BooleanToken indicating whether the connection failed: 'true' if the postpone flag is false and the connection failed, otherwise, 'false'.

Start GAMESS Input (org.resurgence.moml. StartGamessInput)

Author: unknown **Version:** Unknown	The StartGamess actor is a composite actor used inside the GamessInputGenerator actor, which is used in computational chemistry workflows. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system. GAMESS (General Atomic and Molecular Electronic Structure System) is a program that can perform a broad range of quantum chemical computations. For more information about GAMESS, see http://www.msg.ameslab.gov/GAMESS/

Ports

trigger	An input port that accepts a trigger.
inputList	An output port that broadcasts an input list.

Status Checker (org.resurgence.actor.StatusChecker)

Author: Yang Zhao, Ilkay Altintas, Wibke Sudholt
Version: Unknown

The StatusChecker actor checks the status of a running job. The actor executes a Unix command (e.g., ls or C:/Program Files/Internet Explorer/ IEXPLORE.EXE) and compares the execution results to a specified regular expression (e.g., *.html). The StatusChecker actor depends on system-specific executables and is operating-system specific. Commands are specified with the command parameter or via the command port and may take an optional argument. Output the execution results to a file by specifying an outputFile. Once the execution is complete, the actor outputs the results as a string via the output port.

Parameters

command	The command to be executed on the remote host. Commands must be specified as a string.
outputFile	The name of the output file (optional). If an output file is specified, the output of the command will be written to that file.
hasTrigger	Select to activate a trigger input port.
sleepTime	The delay, in milliseconds, between checks. The parameter is of type long. The default is 0, meaning that the actor will not sleep between checks.
checkCondition	A regular expression that the actor will compare against the output.
maxChecks	The maximum number of checks. The default value is -1, meaning that the actor will keep checking until the condition is satisfied.

Ports

command	An input port that accepts the command to be executed. Commands must be specified as a string.
arguments	An input port that accepts command arguments (optional).
infileHandle	An input port that accepts the path to an input file (optional). Used if the command accepts an input file instead of a list of arguments.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
outfileHandle	An output port that broadcasts the output file path, if an outputFile has been specified.
output	An output port that broadcasts data generated by the executed command, output as a string after the command has finished executing.

exitCode	An output port that indicates whether or not the command executed successfully. Exit code will be 1 if the command executes successfully.
iterativeOutput	An output port that broadcasts the output of each iteration.

Stop (ptolemy.actor.lib.Stop)

Author: Edward A. Lee
Version: Unknown

The Stop actor stops a workflow after the current iteration is complete. The actor receives Boolean tokens via its input multiport. When the actor receives a true token, it stops the workflow. When the current iteration is complete depends on the workflow director. Under an SDF Director, the current iteration is concluded and the execution is stopped when the Stop actor receives a true token. Under a DE Director, actors may be invoked after Stop is invoked (if an event with time stamp T and value true arrives at the Stop actor, then concluding the current iteration means processing all events in the event queue with time stamp T, some of which may not yet have been processed before the Stop actor is invoked). The Stop actor is not the best way to stop a PN workflow, which has no well-defined notion of an iteration. To stop a PN workflow, design the workflow so that all actors are starved of data when the workflow is to stop. The director will detect this starvation, and halt the workflow. Note: if nondeterminism is acceptable, the Stop actor can be used.

Ports

input	A multiport that accepts Boolean tokens. When the port receives a true token, the workflow is stopped. If the port is not connected, then the actor requests a pause whenever it fires.

String Accumulator (org.resurgence.actor. StringAccumulator)

Author: Wibke Sudholt
Version: Unknown

The StringAccumulator actor reads multiple strings via its input port and outputs a string containing all of the input values. The characters separating the substrings in the output string can be specified with the substringSeparator parameter.

Parameters

Substring separator	The characters used to separate the elements in the output string. By default, the value is "".

Ports

whole	An output port that broadcasts the full string.
parts	A multiport that accepts substrings.

String Compare (ptolemy.actor.lib.string.StringCompare)

Author: Vinay Krishnan, Daniel Lazaro Cuadrado (contributor: Edward A. Lee)
Version: Unknown

The StringCompare actor reads two strings and compares them using a specified comparison function. The actor outputs true if the strings satisfy the comparison criteria, or false if not. The comparison functions are specified with the function parameter: equals: Output true if the strings are equal (the default). startsWith: Output true if the first string starts with second string. endsWith: Output true if the first string ends with the second string. contains: Output true if the first string contains the second string. Strings will be read from the input ports if they are connected; otherwise strings are read from the parameters.

Parameters

secondString	The second string.
firstString	The first string.
function	The comparison function (equals, startsWith, endsWith, contains). The default is "equals".
ignoreCase	Specify whether to ignore case. By default, the actor is case-sensitive. Select the parameter to ignore case.

Ports

secondString	An input port that accepts the second string.
output	An output port that broadcasts the result of the comparison (either true or false).
firstString	An input port that accepts the first string.

String Constant (ptolemy.actor.lib.StringConst)

Author: Edward Lee **Version:** Unknown	The StringConstant actor outputs a string specified via the actor's value parameter. Specifying strings with the StringConstant actor is convenient, as the actor does not require that strings be surrounded by quotes. The actor is often used to specify file paths, which can be selected using the Browse button available in the actor's parameters. Specified string values can include references to parameters within scope (i.e., parameters defined at the same level of the hierarchy or higher). NOTE: If using a PN Director, the 'firingCountLimit' parameter is often set to a finite integer (e.g. '1') so that the workflow will terminate.

Parameters

firingCountLimit	The limit on the number of times the actor will fire. The default value is 'NONE', meaning there is no limit on the number of time the constant will be provided to the output port. Any integer can be provided as a value for this parameter.
value	The value produced by the actor. Specified strings do not require enclosing quotes. (To include a '$' sign in the string, enter '$$'.)

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
output	An output port that broadcasts a string constant specified by the value parameter.

String Function (ptolemy.actor.lib.string.StringFunction)

Author: Mike Kofi Okyere, Ismael M Sarmiento
Version: Unknown

The StringFunction actor reads an input string, applies a string function (e.g., toUpperCase), and outputs the converted string. The name of the string function to apply is specified with the function parameter. The following functions may be specified: trim: Remove leading and trailing whitespace from a string. toUpperCase: Convert all letters in a string to uppercase. toLowerCase: Convert all letters in a string to lowercase.

Parameters

function	The string function to apply to the input string. The possible values are "trim" (the default), "toUpperCase", or "toLowerCase".

Ports

input	An input port that accepts a string.
output	An output port that broadcasts the converted string.

String Index Of (ptolemy.actor.lib.string.StringIndexOf)

Author: Rakesh Reddy, Philip Baldwin, Edward Lee
Version: Unknown

The StringIndexOf actor reads an input string, searches for a specified substring, and outputs the index of the substring, or -1 if the substring is not found. The actor reads an input string via the inText port or parameter. The actor will search the string for the substring specified by the searchFor port or parameter. The search begins at the string index specified with the startIndex port or parameter and ends at either the end or the beginning of the string, depending on the setting of the searchForwards parameter. Note: An index of 0 refers to the first character of the string.

Parameters

ignoreCase	Specify whether to ignore case or not. By default, the case is relevant.
searchForwards	Specify the direction in which to search. By default, the search proceeds forward.
searchFor	A substring to find in the source string. The substring can also be specified via the searchFor port. The default is an empty string.
inText	The string that will be searched. The string can also be specified via the inText port. The default is an empty string.
startIndex	A string index, which determines where to begin the search. The default is the integer 0, representing the first character in the string. The index can also be specified via the startIndex port.

Ports

searchFor	An input port that accepts a substring to find in the source string. The substring can also be specified with the searchFor parameter. The default is an empty string.
inText	An input port specifying the string that will be searched. The string can also be specified with the inText parameter. The default is an empty string, which matches no strings.
startIndex	An input port that accepts a string index, which determines where to begin the search. The default is the integer 0, representing the first character in the string. The index can also be specified with the startIndex parameter.
output	An output port that broadcasts the index of the searchFor substring, or -1 if the substring is not found.

String Length (ptolemy.actor.lib.string.StringLength)

Author: Edward A. Lee
Version: Unknown

The String Length actor reads a string and outputs its length.

Ports

output	An output port that broadcasts an integer representing the length of the input string.
input	An input port that accepts a string.

String Matches (ptolemy.actor.lib.string.StringMatches)

Author: Antonio Yordan-Nones, Colin Cochran (contribuor Edward a lee)
Version: Unknown

The StringMatches actor reads a string and compares it to a pattern. The actor outputs true if the pattern matches the string, and false if it does not. The pattern is specified as a regular expression. For more information about regular expressions, see http://java.sun.com/docs/books/tutorial/extra/regex/index.html.

Parameters

pattern	A regular expression, which is compared to the input string. The default is an empty string. The regular expression may also be specified via the pattern port.
matchString	A string to be compared to a specified regular expression. The string may also be specified via the matchString port.

Ports

pattern	An input port that accepts a regular expression, which is compared to the input string. The default is an empty string. The regular expression may also be specified via the pattern parameter.
matchString	An input port that accepts a string to be compared to a specified regular expression. The string may also be specified via the matchString parameter.
output	An output port that indicates whether the input string and regular expression match. The actor outputs true if there is a match, or false if not.

StringParameter (ptolemy.data.expr.StringParameter)

Author: Edward A. Lee
Version: Unknown

StringParameter specifies a persistent string. Change the name of the StringParameter to better identify the specified value (right-click the parameter and select "Customize Name" from the menu). Other actors may refer to the StringParameter using the $NAME syntax (e.g. $Parameter).

String Replace (ptolemy.actor.lib.string.StringReplace)

Author: Antonio Yordan-Nones, Neil E. Turner, Edward A. Lee
Version: Unknown

The StringReplace actor reads a string and searches it for a specified pattern. The actor replaces either the first instance of the pattern or all occurrences of the pattern with a specified replacement string. The actor outputs the edited string. The actor accepts a source string, a replacement string, and a pattern to search for via either its input ports or its parameters. The pattern is specified as a regular expression. For more information about regular expressions, see http://java.sun.com/docs/books/tutorial/extra/regex/index.html. If the replaceAll parameter is specified, the actor will replace all instances of the string that match the specified pattern. Otherwise, the actor will replace only the first instance that matches. If the actor finds no match, then the actor outputs the original source string, unchanged.

Parameters

replaceAll	Specify whether or not to replace all instances that match the pattern. By default, the actor will replace all instances. Deselect the parameter to replace only the first instance.
pattern	A regular expression, which is compared to the input string. The default is an empty string. The regular expression may be specified via either the pattern port or parameter.
replacement	A replacement string that replaces any matched instance of the specified pattern. The default is an empty string. The replacement string may be specified via either the replacement port or parameter.
stringToEdit	A source string. The actor will edit the string based on the settings specified by the pattern, replacement, and replaceAll parameters. The source string may be specified via either the stringToEdit port or parameter.

Ports

pattern	A port that accepts a regular expression, which is compared to the input string. The default is an empty string. The regular expression may also be specified via the pattern parameter.
replacement	A port accepts a replacement string that replaces any matched instance of the specified pattern. The default is an empty string. The replacement string may also be specified via the replacement parameter.

stringToEdit	A port that accepts a source string. The actor will edit the string according to the settings specified by the pattern, replacement, and replaceAll parameters. The source string may also be specified with the stringToEdit parameter
output	An output port that broadcasts the edited string.

String Splitter **(org.resurgence.actor.StringSplitter)**

Author: Wibke Sudholt
Version: Unknown

The StringSplitter actor reads a string and splits it into segments at specified points. The actor outputs the segments as an array of strings. The actor separates the input string at points specified by the regular expression parameter. For example, if the input string is a URL (http://www.mysite.com/home/archive/stuff) and the Regular expression parameter is /, the actor will output the following array {"http:", "", "home", "archive", "stuff"}. For more information about regular expressions, see http://java.sun.com/docs/books/tutorial/extra/regex/index.html.

Parameters

Regular expression	A regular expression used to identify the points at which the actor should break the string. For more information about regular expressions, see http://java.sun.com/docs/books/tutorial/extra/regex/index.html.

Ports

string	An input port that accepts a string.
array	An output port that broadcasts an array of string segments.

String Substring (ptolemy.actor.lib.string.StringSubstring)

Author: Neil Turner and Edward Lee **Version:** Unknown	The StringSubstring actor reads a string and identifies and outputs a specified segment of the string. The substring to output is identified with the start and stop ports or parameters. Start and stop accept an integer representing a string index (e.g., 0, which identifies the first character in the string). The start character is included in the output string; the stop character is not (i.e., the substring terminates just before the stop character). If the value specified for stop is less than the value specified for start, then the substring starts at start and extends to the end of the string. The default values for start and stop are both 0; the default values identify an empty string.

Parameters

stop	The stop string index, which identifies the position of the last character of the substring. The stop index is one greater than the position of the last letter of the desired substring. The value is an integer that defaults to 0.
start	The start string index, which identifies the position of the first character of the substring. The value is an integer that defaults to 0.

Ports

output	An output that broadcasts the specified substring.
input	An input port that accepts a string.
stop	An input port that accepts an integer string index. The index identifies the position of last character of the substring. The stop index is one greater than the position of the last letter of the desired substring. The default is 0.
start	An input port that accepts an integer string index. The index identifies the position of first character of the substring. The default is 0.

String To Image Converter (util.StringToImage)

Author: Tobin Fricke
Version: Unknown

The StringToImageConverter actor reads a string containing raw data that represents an image in a standard format (e.g., JPEG or PNG) and outputs an image token.

Ports

output	An output port that broadcasts an image token derived from the input string.
input	An input port that accepts a string containing raw data that represents an image in a standard format (e.g., JPEG or PNG).

String To Int (org.resurgence.actor.StringToInt)

Author: Wibke Sudholt
Version: Unknown

The StringToInt actor reads a string, converts it to an integer, and outputs the result. The actor creates the integer by placing one byte (i.e., one character) of the string into the least significant byte of an integer. Typically, this byte is the ASCII code of the character.

Ports

string	An input port that accepts a string.
integer	An output port that broadcasts the converted string as an integer.

String To Long (org.resurgence.actor.StringToLong)

Author: Chad Berkley
Version: 1.0

The StringToLong actor reads a string, converts it to a long, and outputs the result. The actor creates the long by placing one byte (i.e., one character) of the string into the least significant byte of a long. Typically, this byte is the ASCII code of the character.

Ports

string	An input port that accepts a string.
integer	An output port that broadcasts the converted string as a long.

String To Polygon Converter (org.geon.StringToPolygon)

Author: Efrat Jaeger
Version: Unknown

The StringToPolygon actor is a domain specific actor used within the GEON mineral classifier workflow. GEON (Geosciences Network) is a distributed infrastructure for Geoscience research and education. For more information about GEON, see http://www.geongrid.org/. The actor reads a string of the following format: {{x1,y1},{x2,y2},...},region}. The actor outputs an array of coordinates and a string representing the name of a diagram region.

Ports

input	An input port that accepts a string of the following format: {{x1,y1},{x2,y2},...}, region}.
region	An output port that broadcasts a string representing a diagram region name.
coordinates	An output port that broadcasts an array of coordinates representing a region.

String To XML (ptolemy.actor.lib.conversions.StringToXML)

Author: Yang Zhao
Version: Unknown

The StringToXML actor converts a string token to an XML token. XML is a markup language for files containing structured information (content as well as its context, e.g., whether a value is a "header" or "footnote"). For more information about XML, see http://www.w3.org/XML/.

Ports

output	An output port that broadcasts an XML token (i.e., encapsulated XML data that can be sent to other actors).
input	An input port that accepts a string to be transformed into an XML token.

SubsetChooser (org.cipres.kepler.SubsetChooserActor)

Author: Alex Borchers, Zhijie Guan
Version: Unknown

The SubsetChooser actor allows users to select a subset of molecular data sequences or a set of trees via a graphical interface. If the input data is a set of sequences, then it follows the FASTA format. If the input data is a set of trees, it follows the NEXUS format. The actor writes the selected entities to an output file. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Parameters

Output File Path and Name	The name and path of the output file.

Ports

Selected entities file	An output port that broadcasts the user-selected subset of entities (molecular sequences or trees).
Subset Chooser Input File	An input port that accepts the file name of the input file. The input data can be either a set of molecular data sequences or a set of trees and allows users to select a subset of the input values via a graphical interface. If the input data is a set of sequences, then it follows the FASTA format. If the input data is a set of trees, it follows the NEXUS format

Summary (org.ecoinformatics.seek.R.RExpression)

Author: Josh Madin
Version: Unknown

The Summary actor uses R to calculate a specified summary statistic. The actor accepts a number of factors and a variable, and outputs the specified summary statistic (e.g., presence, mean, standard deviation, variance, etc). The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. Input factors are input as an array of values, and might include "date", "site," or "replicate" data. The output summary table is arranged according to the order of the input factors. In addition to factors, a variable (usually a continuous variable such as "height" or "rain fall") must be specified via the variable port. By default, the actor performs a "presence/absence" analysis. To specify another type of analysis, specify a R-function (e.g., mean, sum, max, min, sd, prod, etc.) with a Constant actor and connect it to the method input port. The actor performs the analysis and saves a "summary.txt" text file to the Kepler working directory. On Unix-based machines, this file will be opened automatically during workflow execution. Presently the output of the actor is a dummy output, and to access the summary table the next R actor must load the R workspace using 'load(".RData")' as the first line of the its R script.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

Variable	The variable to be summarized.
Factor5	Any factor of interest (e.g., 'date' or 'site' data)
graphicsFileName	An output port that broadcasts the file name of the generated graph of the results.
Factor4	Any factor of interest (e.g., 'date' or 'site' data)
Factor3	Any factor of interest (e.g., 'date' or 'site' data)
Factor2	Any factor of interest (e.g., 'date' or 'site' data)
Factor1	Any factor of interest (e.g., 'date' or 'site' data)
method	The summary method. The default method is "presence." Use the Constant actor to input another summary method (e.g., mean, max, sum, etc.)
graphicsFileName	An output port that broadcasts the file name of the graph of the generated results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).

SummaryStatistics (org.ecoinformatics.seek.R.RExpression)

Author: Dan Higgins **Version:** Unknown	The SummaryStatistics actor accepts an array of values and uses R to calculate their mean, standard deviation, and variance. The actor outputs both a graphical and textual representation of the summary analysis. The R application must be installed on the local system to use this actor. R is a language and environment for statistical computing and graphics. For more information about R, see http://www.r-project.org/. The actor performs the analysis and saves a graphic of the results to the Kepler working directory. To view the results, connect an ImageJ actor to the graphicsFileName output port and/or a Display actor to the xmean, xstd, and xvar output port.

Parameters

numXPixels	The width of the output graphic in pixels.
Rcwd	The 'R' working directory (the home directory by default).
expression	The expression that is evaluated to produce the output.
graphicsOutput	Specify whether or not to create a graphics output port. By default, the actor will create a graphics output port.
displayGraphicsOutput	Select to display the plot once the actor has generated it.
graphicsFormat	The graphics output format. Currently the actor supports either *.pdf or *.png.
numYPixels	The height of the output graphic in pixels.
save_nosave	Specify whether or not to save the R workspace when R is closed; set to '--save' to retrieve the workspace later in a workflow in another R-actor.

Ports

graphicsFileName	An output port that broadcasts the filename of the JPEG representing the generated results.
output	An output port that broadcasts a copy of the text output that R generates. As the actor communicates with R to run the R function or script, the output port sends back the response (i.e., the values and statistical outputs).
x	An input port that accepts an array of values to evaluate.
xstd	An output port that broadcasts the standard deviation of the input.
xmean	An output port that broadcasts the mean of the input.

xvar	An output port that broadcasts the variance of the input.

Switch (ptolemy.actor.lib.Switch)

Author: Edward A. Lee
Version: Unknown

The Switch actor routes inputs to specified output channels. The actor has two input ports: the input port for data, and the control port, used to select which output channel to use. The actor outputs the input token on the channel most recently received via the control port. Each time the actor iterates, it checks to see if a token is available on its control port. If so, the actor reads the value and uses it to determine which of its output channels to broadcast (0,1,2,3, etc.). If an input token is available on the input port, then that token is read and sent to the specified output channel. If the actor has never received a value on the control port, then the actor will output to channel zero. If the value of the most recently received control token is out of range (less than zero, or greater than or equal to the width of the output), then the actor will not fire and the input token will be lost. The actor is similar to the TokenToSeparateChannelsTransmitter, except that the output channel of the Switch actor is specified via the control port. The TokenToSeparateChannelsTransmitter sends each input token to a different consecutive output channel.

Ports

output	An output port that broadcasts to the channel specified via the control port.
input	An input port that accepts tokens of any type.
control	An input port that accepts integer tokens that indicate which output channel to broadcast (0, 1, 2, 3, etc).

SyncOnTerminator (org.sdm.spa.SyncOnTerminator)

Documentation coming soon!

Temp Actor (org.geon.TempActor)

Documentation coming soon!

Temporary Script Creator (org.resurgence.moml. TemporaryScriptCreator)

Author: unknown **Version:** Unknown	The TemporaryScriptCreator is a composite actor used in computational chemistry workflows. To use the full suite of computational chemistry actors, GAMESS, Open Babel, Babel, and QMView software must be installed on the local system.

Test (ptolemy.actor.lib.Test)

Author: Edward A. Lee, Christopher Hylands, Jim Armstrong
Version: Unknown

The Test actor reads an input value of any type and compares it to a corresponding value in a specified array of values. If the two values match, the actor fires successfully. If the two values do not match, the actor generates an error. The actor outputs a Boolean token: false if the actor's iteration count is less than or equal to the length of the comparison array; true if the iteration count is greater than the length of the array. The Boolean output can be fed to a Stop actor to stop the workflow upon successfully matching test data. Unlike the NonstrictTest actor, the Test actor supports a multiport input. The actor also differs from NonstrictTest in that it requires that all inputs be present. The comparison value is specified with the correctValues parameter. The parameter accepts an array of values, the type of which must match the type of the input (the default array is {true}). The actor cycles through the array values, comparing each consecutive input to the next token in the correctValues array. After each of the values in the correctValues parameter has been matched, any subsequent iteration always succeeds. This behavior allows the actor to be used as a "power-up" test for a model; the actor will check the first few iterations against some known results. The input is a multiport. If more than one input channel exists, then each element of correctValues must itself be an array, with length matching the number of channels. Suppose for example that the input consists of one channel, and the first three inputs should be 1, 2, and 3. Then set correctValues to {1, 2, 3} Suppose instead that the input consists of two channels, and the correct values in the first iteration are 1 on the first channel and 2 on the second. Then on the second iteration, the correct values are 3 on the first channel and 4 on the second. Then set correctValues to {{1, 2}, {3, 4}} Note: With this setting, no tests are performed after the first two iterations of this actor. The input values are checked when the actor fires. If an input value is missing or differs from what it should be, then the actor generates an error. To check the input value after the actor fires, then use the NonstrictTest actor. If the input is a double or complex token, then the comparison "passes" if the value is close to what it should be (i.e., within the specified tolerance). Tolerance is specified with the tolerance parameter, and defaults to 10-9.

Parameters

tolerance	A double token specifying how closely the input must match the value from the correctValues array. The default is 10-9.
correctValues	An array specifying what the input should be. If more than one input channel exists, then each element of the array must itself be an array, with length matching the number of channels. The default is a one-by-one array containing a Boolean true.
trainingMode	Select the trainingMode parameter to collect the input values and place them in the correctValues array. The trainingMode parameter is a shared parameter, meaning that changing it for any one instance of the actor in the model will change all instances.

Ports

output	Boolean output that is false as long as there is data to compare against the input; the value becomes true on the first firing after such data has been exhausted.
input	A multiport that accepts tokens of any type.

Text File Writer (org.resurgence.actor.TextFileWriter)

Author: Wibke Sudholt **Version:** Unknown	The TextFileWriter actor reads a string token and writes it, without any enclosing quotation marks, to a file. The actor outputs the file path of the generated file. The path and name of the output file are specified via input ports. Specify whether the input is appended to the specified file (if it exists), or if an existing file is overwritten via the changeExisting parameter. The TextFileWriter is similar to the FileWriter actor, except that the TextFileWriter actor does not add line breaks to its output, while the FileWriter actor does.

Parameters

Change existing	Specify whether the input is appended to the specified file (if it exists), or if an existing file is overwritten.

Ports

fileToWrite	An input port that accepts the file path of the file to which to write. See FileParameter for more information about specifying file names.
string	An input port that receives a string token to write-one line at a time--to a file.
fileWritten	An output port that broadcasts the name and path of the output file.

Throw Exception (ptolemy.actor.lib.ThrowException)

Author: Edward A. Lee
Version: Unknown

The ThrowException actor generates an error when it receives a true Boolean token on any channel of its input multiport. The text of the error message is specified by the message parameter. The actor is similar to ThrowModelError, except that the ThrowException actor's generated error is local (i.e., it is thrown at the local hierarchy level). The ThrowModelError will pass the exception up the workflow hierarchy rather than immediately throwing it.

Parameters

message	The error message that will be reported.

Ports

input	A multiport that receives Boolean tokens. The actor will generate an error message whenever a true token is received on any channel.

Throw Model Error (ptolemy.actor.lib.ThrowModelError)

Author: Haiyang Zheng **Version:** Unknown	The ThrowModelError actor generates a model error when it receives a true Boolean token on any channel of its input multiport. The text of the error message is specified by the message parameter. A model error is an exception that is passed up the containment hierarchy of the workflow rather than immediately thrown. Any container in the hierarchy may choose to handle the error. By default, containers will pass and delegate the error to their container, if they have one, and throw the exception if they don't. Some containers may do more with the error.

Parameters

message	The error message that will be reported.

Ports

input	A multiport that receives Boolean tokens. The actor will generate an error message whenever a true token is received on any channel.

Time Stamp **(org.sdm.spa.Timestamp)**

Author: Ilkay Altintas **Version:** Unknown	The TimeStamp actor outputs the current date and time. Dates and times can be formatted in a variety of ways. Select a format from the drop-down menu beside the format parameter, or enter a pattern directly into the format field. For more information about date format patterns, see http://java.sun.com/docs/books/tutorial/i18n/format/datepattern.html.

Parameters

format	The date format. Select a format string from the drop-down menu or type in a custom format string directly into the field. For more information about date formatting patterns, see http://java.sun.com/docs/books/tutorial/i18n/format/datepattern.html

Ports

output	An output port that broadcasts the current time stamp as a string.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.

Timed Plotter (ptolemy.actor.lib.gui.TimedPlotter)

Author: Edward A. Lee
Version: Unknown

The TimedPlotter actor reads one or more double tokens via its input multiport and plots each as a separate data set each time the actor iterates. The actor displays the graph on the screen. The horizontal axis represents time.

Parameters

fillOnWrapup	Specify whether to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
startingDataset	The starting dataset number. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the plot graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).

Ports

input	A multiport that accepts double tokens that will be plotted over time.

Timed Scope (ptolemy.actor.lib.gui.TimedScope)

Author: Edward A. Lee **Version:** Unknown	The TimedScope actor plots its input in an oscilloscope style (the horizontal axis is wrapped and the graphed points have a finite persistence). The actor reads one or more double tokens via its input multiport and displays the graphed data on the screen. The horizontal axis represents time. The width parameter is a double that gives the width of the plot. The horizontal axis will be labeled from 0.0 to width. The width defaults to 10.0. The persistence parameter specifies the length of time that points are displayed. If the persistence is positive, then it specifies the amount of time into the past that points are shown. For example, at the default persistence (10), any point older than 10.0 time units is erased and forgotten.

Parameters

fillOnWrapup	Specify whether to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
width	A double that specifies the width of the plot. The horizontal axis will be labeled from 0.0 to width. The default is 10.0.
startingDataset	The starting dataset number. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the plot graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).
persistence	The amount of data displayed at any one time. The default is 10.0 time (i.e., x-axis) units.

Ports

input	A multiport that accepts double tokens that will be plotted over time in an oscilloscope style.

Token Counter (org.resurgence.actor.TokenCounter)

Author: Wibke Sudholt
Version: Unknown

The TokenCounter actor counts the number of received tokens. The actor accepts multiple tokens of any type, and outputs the number of tokens it has received.

Ports

token	An input port that accepts tokens of any type.
number	An output port that broadcasts the number of received tokens.

Token Duplicator (org.geon.Duplicator)

Author: Efrat Jaeger
Version: Unknown

The TokenDuplicator actor reads a token of any type and duplicates it a specified number of times.

Ports

output	An output port that broadcasts the duplicated tokens.
input	An input port that accepts a token of any type.
numCopies	An input port that accepts an integer specifying the number of copies to create.

Token To Expression (ptolemy.actor.lib.conversions.TokenToExpression)

Author: Steve Neuendorffer, Haiyang Zheng
Version: Unknown

The TokenToExpression actor accepts a token of any data type and outputs that token as a quoted string. The output string can be parsed to yield the original input token. For example, if the input is the integer 3, the TokenToExpression actor will output the string "3". If the input value is the string "hello", the actor will output the double-quoted string, ""hello"".

Ports

input	An input port that accepts a data token of any type.
output	An output port that broadcasts the converted input as a string.

Token To String Converter (org.geon.TokenToString)

Author: Efrat Jaeger
Version: Unknown

The TokenToString actor accepts a token of any data type and outputs that token as a quoted string. The output string can be parsed to yield the original input token. The TokenToString actor is similar to the TokenToExpression actor, except that it accepts an input value through either a port or a parameter, whereas the TokenToExpression actor accepts input only via an input port. For example, if the input is the integer 3, the TokenToExpression actor will output the string "3". If the input value is the string "hello", the actor will output the double-quoted string, ""hello"".

Parameters

value	The value to output as a string. Values may be of any data type.

Ports

input	An input port that accepts a data token of any type.
output	An output port that broadcasts the converted input as a string.

Token to Separate Channels Transmitter (org.geon. TokenToSeparateChannels)

Author: Efrat Jaeger **Version:** Unknown	The TokenToSeparateChannelsTransmitter reads tokens of any type and outputs each token to a different consecutive output channel. The actor is similar to the Switch actor, except that the output of the TokenToSeparateChannelsTransmitter is automatically sent to different consecutive output channels. The Switch actor sends its output to the channel specified via its control port.

Ports

output	A multiport that broadcasts the input tokens over different consecutive channels (0, 1, 2, 3, etc).
input	An input port that accepts tokens of any type.

Transitive Closure Database Query (org.geon. TransitiveClosureDBQuery)

Author: Efrat Jaeger
Version: Unknown

The TransitiveClosureDatabaseQuery actor performs database queries against an open database using an initial set of values as a constraint. If a query result does not match a value in the initial constraint set, the actor adds that result to the set. The actor continues to query the database until no new values are added to the constraint set. The actor outputs all values that match the constraint. The actor requires a reference to an open database, which can be created and passed via the OpenDatabaseConnection actor. A query is specified with either the query port or parameter. In addition, an initial set of query constraints is specified via the fieldInSet parameter. The actor outputs all values that match the constraint set, either as an array of values, or as a sequence of separate tokens, depending on the setting of the outputEachRowSeparately parameter.

Parameters

outputEachRowSeparately	Specify whether to display the complete result as an array, or to output each result separately.
fieldInSet	An array of initial query constraint values (e.g., {val1, val2, val3})
query	A database query string.

Ports

query	A database query string.
initialSet	An input port that accepts an array of initial query constraint values (e.g., {val1, val2, val3})
dbcon	An input port that accepts a reference to the database connection, which can be created and passed via the OpenDatabaseConnection actor.
result set	An output port that broadcasts the values that match the constraint set, either as an array of values, or as a sequence of separate tokens, depending on the setting of the outputEachRowSeparately parameter.

TreeDecomposer **(org.cipres.kepler.TreeDecomposer)**

Author: Zhijie Guan **Version:** Unknown	The TreeDecomposer actor receives a phylogenetic tree, which it breaks into subtrees and outputs. The actor, which invokes a CIPRES CORBA service to perform the operation, outputs all subtrees as a single token. CORBA services, much like Web services, are computer programs that run on a remote host and communicate using a standardized protocol that allows them to interoperate. CIPRES CORBA services are specifically designed to help analyze phylogenetic data sets. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The TreeDecomposer actor must also be used with the Initializer actor, which connects to the registry of CORBA services used by the CIPRES actors. The input data tree can be generated by the PhyloDataReader actor, which reads a Nexus file and outputs the data in a form that can be readily used by the TreeDecomposer actor. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

inputTree	An input port that accepts a phylogenetic tree in the format used by CIPRES applications.
outputTrees	An output port that broadcasts the generated subtrees. All subtrees are wrapped in a single token to facilitate the transportation.

TreeImprover (org.cipres.kepler.TreeImprover)

Author: Zhijie Guan **Version:** Unknown	The TreeImprover actor improves a phylogenetic tree based on user-specified settings. The actor, which invokes a CIPRES CORBA service to perform the operation, outputs the improved tree as a single token. CORBA services, much like Web services, are computer programs that run on a remote host and communicate using a standardized protocol that allows them to interoperate. CIPRES CORBA services are specifically designed to help analyze phylogenetic data sets. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The TreeImprover actor must also be used with the Initializer actor, which connects to the registry of CORBA services used by the CIPRES actors. The actor's input values--the data matrix and tree--can be generated by the PhyloDataReader actor, which reads a Nexus file and outputs the data in a form that can be readily used by the TreeImprover actor. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

inputTree	An input port that accepts a CIPRES tree data structure consisting of the tree name, tree score, leaf set, and/or Newick. The tree can be generated by the PhyloDataReader actor, which reads Nexus files and outputs data in a format that CIPRES actors can readily use.
outputTree	An output port that broadcasts the improved tree in the format used by CIPRES applications (tree name, tree score, leaf set, and Newick).
inputMatrix	An input port that accepts a matrix containing character information about the analyzed taxa. The data matrix can be generated by the PhyloDataReader actor, which reads Nexus files and outputs data in a format that CIPRES actors can readily use.

TreeMerger (org.cipres.kepler.TreeMerger)

Author: Zhijie Guan **Version:** Unknown	The TreeMerger actor receives a set of phylogenetic trees, merges them, and outputs the combined tree. The actor uses a CIPRES CORBA service to perform the operation. CORBA services, much like Web services, are computer programs that run on a remote host and communicate using a standardized protocol that allows them to interoperate. CIPRES CORBA services are specifically designed to help analyze phylogenetic data sets. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The TreeMerger actor must also be used with the Initializer actor, which connects to the registry of CORBA services used by the CIPRES actors. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

inputTrees	An input port that accepts a set of phylogenetic trees in the data structure used by CIPRES applications (tree name, tree score, leaf set, and Newick).
outputTree	An output port that broadcasts the merged tree.

TreeParser (org.cipres.kepler.TreeParser)

Author: Zhijie Guan
Version: Unknown

The TreeParser actor parses a tree data structure into its components: the tree name, tree score, leaf set, and Newick. The actor broadcasts each component via a dedicated output port.

Ports

outputLeafSet	An output port that broadcasts the tree's leaf set.
inputTree	An input port that accepts a tree data structure in the format used by CIPRES applications.
outputScore	An output port that broadcasts the tree score.
outputName	An output port that broadcasts the tree name.
outputNewick	An output port that broadcasts the tree's Newick expression.

TreeToString (org.cipres.kepler.TreeToString)

Author: Zhijie Guan
Version: Unknown

The TreeToString actor receives a phylogenetic data set representing a tree expression (a tree name, tree score, leaf set, and/or Newick), and transforms the data set into a single string.

Ports

outputString	An output port that broadcasts a string representing the input tree data.
inputTree	An input port that accepts a phylogenetic data set representing a tree expression. This expression may include the tree name, tree score, leaf set, and/or Newick.

TreeVizForester (org.cipres.kepler.TreeVizForester)

Author: Zhijie Guan **Version:** Unknown	The TreeVizForester actor reads a phylogenetic tree (in Newick format) and displays the tree in a viewing window. The actor uses a CIPRES CORBA service to perform the operation. CORBA services, much like Web services, are computer programs that run on a remote host and communicate using a standardized protocol that allows them to interoperate. CIPRES CORBA services are specifically designed to help analyze phylogenetic data sets. To use the full suite of CIPRES actors, CIPRES software must be installed on the local system. The TreeVizForester actor must also be used with the Initializer actor, which connects to the registry of CORBA services used by the CIPRES actors. The CIPRES (Cyberinfrastructure for Phylogenetic Research) project works to enable large-scale phylogenetic reconstructions that facilitate analyses of datasets containing large numbers of bio molecular sequences. For more information about CIPRES, see http://www.phylo.org/

Ports

inputTreeString	An input port that accepts a tree expression to pass to the Forester tree display package. Tree expressions should be in Newick format. For more information about the Newick data format, see http://evolution.genetics.washington.edu/phylip/newicktree.html.

Trig Function (ptolemy.actor.lib.TrigFunction)

Author: Edward A. Lee
Version: Unknown

The TrigFunction actor takes an input value and computes a specified trigonometric function. The actor outputs a double token representing the result. The function is specified with the function parameter. Functions include: acos: The arc cosine of an angle, in the range from 0.0 through pi. If the argument is NaN or its absolute value is greater than 1, then the result is NaN. asin: The arc sine of an angle, in the range of -pi/2 through pi/2. If the argument is NaN or its absolute value is greater than 1, then the result is NaN. If the argument is positive zero, then the result is positive zero; if the argument is negative zero, then the result is negative zero. atan: The arc tangent of an angle, in the range of -pi/2 through pi/2. If the argument is NaN, then the result is NaN. If the argument is positive zero, then the result is positive zero; if the argument is negative zero, then the result is negative zero. cos: The trigonometric cosine of an angle. If the argument is NaN or an infinity, then the result is NaN. sin: The trigonometric sine of an angle. If the argument is NaN or an infinity, then the result is NaN. tan: The trigonometric tangent of an angle. If the argument is NaN or an infinity, then the result is NaN. If the argument is positive zero, then the result is positive zero; if the argument is negative zero, then the result is negative zero NOTE: The above documentation is adapted from the class documentation for java.lang. Math as released in JDK 1.3.

Parameters

function	The function to compute: acos, asin, atan, cos, sin, or tan. The default is sin.

Ports

output	An output port that broadcasts the computed result of the specified trigonometric function.
input	An input port that accepts a double token representing a value to evaluate.

Type Test (ptolemy.actor.lib.TypeTest)

Author: Steve Neuendorffer
Version: Unknown

The TypeTest actor reads typed values from specified actors and compares them to types stored in the TypeTest actor's parameters. The actor is used to double-check Kepler's type resolution system. After a workflow initializes and Kepler has automatically resolved data types, the TypeTest actor compares the resolved types to the types stored in its parameters. If the types are the same, the actor does nothing; if the types are different, the actor generates an error. The actor stores expected data types in two parameters, portTypes and parameterTypes. Each parameter contains a record consisting of a label corresponding to the name of an actor, and a nested record consisting of the name of a typed object and its type, e.g., {Constant = {value = "long"}, Constant2 = {value = "double"}}. For the portTypes parameter, the typed objects are assumed to be ports; for the parameterTypes parameter, the objects are assumed to be parameters. Note: referenced actors must be at the same hierarchical level as the TypeTest actor. The TypeTest actor only tests type resolution at one level of opaque hierarchy. Hierarchical models should include multiple instances of the actor. Because manually filling in the type parameters is difficult, the actor includes a training mode. Select the trainingMode parameter to automatically fill in the type parameters. It is not necessary to specify the types of all typeable objects. Any objects for which no types are specified will not be checked.

Parameters

portTypes	Values specifying what the port types should be. The format is a record consisting of a label corresponding to the name of an actor, and a nested record consisting of the name of the actor's port and a string representing its type, e.g., {Constant = {output = "long", trigger = "unknown"}, Constant2 = {output = "{double}", trigger = "unknown"}}
parameterTypes	Values specifying what the parameter types should be. The format is a record consisting of a label corresponding to the name of an actor, and a nested record consisting of the name of the actor's parameter and a string representing its type, e.g., {Constant = {value = "long"}, Constant2 = {value = "double"}}.

trainingMode	Select the trainingMode parameter to collect the input types and place them in the proper format in the parameterTypes and portTypes parameters. The trainingMode parameter is a shared parameter, meaning that changing it for any one instance of the actor in the model will change all instances.

Ports

input	A multiport that accepts input of any type.

URL To Local File (util.URLToLocalFile)

Author: Dan Higgins
Version: Unknown

The URLToLocalFile actor reads a URL and copies it to the local file system. The actor can also be used to read a local file and then write it to another location in the file system. Use the optional outputFilePort to provide a name for the copied file. Specify an output file name with either the outputFilePort or the outputFile parameter. When the actor is done reading and saving the file, the output port will produce a true value.

Parameters

overwrite	Select overwrite to replace the content of an existing destination file with the new content.
Parameters @UserLevelDescription	
outputFile	The path of an output file to which to write.
fileOrURL	The file name or URL from which to read. See FileParameter for more information about specifying file names.

Ports

output	An output port that indicates whether or not the end of the file has been reached. If the end of the file has been reached, the port will produce a true value. Otherwise, the value is false.
trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
fileOrURLPort	An optional input port that accepts the file name or URL of a file to be read. When the port is connected, the actor reads the file sent by the previous workflow step. The file name or URL can also be specified using the fileOrURL parameter.
outputFilePort	An optional input port that accepts the name of the output file. The output file name can also be specified using the outputFile parameter.

Unit Converter (ptolemy.actor.lib.conversions.InUnitsOf)

Author: Yuhong Xiong, Xiaojun Liu, Edward Lee **Version:** Unknown	The UnitConverter actor converts an input specified in units (e.g., meters, or cm/sec, etc) to another unit type (e.g., inches, or in/ms). The UnitConverter actor can be used to ensure that numbers are interpreted correctly in a model, and can help prevent certain critical errors. The actor outputs the converted value without units. The actor must be used with a Unit System (BasicUnits or CGSUnitBase, for example). Double-click the Unit System to see the list of the units it defines. The actor accepts a double token with units (e.g., 23.7*cm/sec) and converts it by dividing by the value of the units parameter (e.g., inches/minute). The units parameter must contain units of the same unit category as the input. Unit categories include length, time, mass, or length/time (speed), length*length (area). If the unit categories are different, the actor will generate an error.

Parameters

units	The units to which the input tokens will be converted. The default value of this parameter is 1.0. For a list of workflow units, double-click the Unit System (BasicUnits, CGSUnitBase, etc). A Unit System must be used with the UnitConverter actor.

Ports

input	An input port that accepts a double token with units (e.g., 23.7*cm/sec).
output	An output port that broadcasts the converted value without units.

Uniform Distribution Random Number Generator (ptolemy.actor.lib.Uniform)

Author: Edward A. Lee **Version:** Unknown	The UniformDistributionRandomNumberGenerator generates a sequence of random numbers with a uniform distribution. A new random number is output each time the actor fires. The generated values are independent and identically distributed with bounds defined by the parameters. In addition, the seed can be specified as a parameter to control the sequence that is generated.

Parameters

seed	The seed that controls the random number generation. A seed of zero (the default) means that the seed is derived from the current system time and a Java hash code (i.e., System.currentTimeMillis() + hashCode()). With extremely high probability, the default seed will ensure that two distinct actors will have distinct seeds. However, current time may not have enough resolution to ensure that two subsequent executions of the same model have distinct seeds. The parameter contains a long token, initially with value 0.
upperBound	The upper bound. The value is a double token that defaults to 1.0.
lowerBound	The lower bound. The value is a double token that defaults to 0.0.
resetOnEachRun	Select to reset the random number generator each time the workflow is run. By default, the generator does not reset.

Ports

output	An output port that broadcasts the generated random number.
trigger	An input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
upperBound	An input port that accepts the upper bound. The value is a double token that defaults to 1.0.
lowerBound	An input port that accepts the lower bound. The value is a double token that defaults to 0.0.

Updated Grid FTP (org.sdm.spa.GridFTPUpdated)

Author: Ilkay Altintas **Version:** Unknown	The UpdatedGridFTP actor is used for reading and writing to files on Globus servers. The actor must be used with a GlobusProxy actor, which passes a proxy certificate used to connect to the remote host. Globus is an open source software toolkit used for building Grid systems, which help people share computing power, databases, and other tools. For more information about Globus, see http://www.globus.org. GridFTP is a secure data transfer protocol optimized for wide-area networks.

Parameters

SourceHostname	The name of a Globus server (e.g., "griddle.sdsc.edu")
Full path to source file	The full path to the files to fetch (e.g., "/archive/2000")
DestinationHostname	The name of the destination host (e.g., "localhost")
Full path to destination file	The full path in which to place the fetched files.

Ports

trigger	A multiport that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) If the port is connected, the actor will not fire until the trigger port receives an input token. Connecting the port is optional, but useful when scheduling the actor to perform at a certain time.
input	An input port that accepts a proxy certificate produced by the GlobusProxy actor.
output	An output port that broadcasts the new file paths as a semicolon-separated string.

User Interactive Shell (org.sdm.spa.gui. UserInteractiveShell)

Author: Ilkay Altintas
Version: Unknown

The UserInteractiveShell actor creates two Unix command shells on the screen: one where users can read and write strings, and another where the actor sends its output. The UserInteractiveShell actor depends on system-specific executables and is operating-system specific. Each time the actor fires, it reads the input, displays it, then displays a command prompt, and waits for a command to be typed by the user. The command is terminated by an enter or return character, which then results in the command being produced on the output.

Ports

input	An input port that accepts strings. Each time the actor iterates, it displays any received strings before displaying a command prompt.
output	An output port that broadcasts the command.

Variable Setter (ptolemy.actor.lib.SetVariable)

Author: Edward Lee, Steve Neuendorffer, Jerome Blanc
Version: Unknown

Set the value of a variable. The result may occur at two different times, depending on the value of the delayed parameter. If delayed is true, then the change to the value of the variable is implemented in a change request, and consequently will not take hold until the end of the current toplevel iteration. This helps ensure that users of value of the variable will see changes to the value deterministically (independent of the schedule of execution of the actors). If delayed is false, then the change to the value of the variable is performed immediately. This allows more frequent reconfiguration, and can mimic the operation of PGM's graph variables. Note that the variable name is observed during preinitialize(). If it is changed after that, the change will not take effect until the next time the model is executed. Moreover, the type of the variable is constrained in preinitialize() to be at least that of the input port for this actor. The variable can be either any attribute that implements the Settable interface. If it is in addition an instance of Variable, then the input token is used directly to set the value, and the type of the variable is constrained to be the same as the type of the input. Otherwise, then input token is converted to a string and the setExpression() method on the variable is used to set the value. The variable can occur anywhere in the hierarchy above the current level. If the variable is not found in the container, then the container of the container is checked until we reach the top level.

Vector Assembler (ptolemy.actor.lib.VectorAssembler)

Author: Jie Liu, Elaine Cheong
Version: Unknown

The VectorAssembler accepts double tokens (e.g., 2.3, 15, 11.8) via its input port and assembles and outputs a one-column matrix (e.g., [2.3, 15, 11.8]). The actor's input port is a multiport, meaning that it can accept multiple channels of data. The actor reads, assembles, and outputs one token from each channel of the input port every time it fires.

Ports

input	A multiport that accepts double tokens (2.3, 15, etc) on each channel.
output	An output port that broadcasts one-column matrix consisting of the input values (e.g., [2.3, 15, 11.8])

Vector Disassembler (ptolemy.actor.lib. VectorDisassembler)

Author: Jie Liu, Elaine Cheong **Version:** Unknown	The VectorDisassembler accepts a one-column matrix (e.g., [2.3, 15, 11.8]) and outputs each row as a double token (e.g., 2.3 or 15) via the channels of its output port. The actor's output port is a multiport, meaning that it can broadcast multiple channels of data. If the width of the output port is less than the number of matrix rows, then the extra output tokens are discarded (i.e., only the first elements in the matrix are output via the available channels).

Ports

input	An input port that accepts a one-column matrix (e.g., [2.3, 15, 11.8])
output	A multiport that broadcasts double tokens (2.3, 15, etc) on each channel.

WSWithComplexTypes (org.sdm.spa. WSWithComplexTypes)

Author: Daniel Crawl
Version: Unknown

This actor executes web services defined by WSDLs. Given a web service's URL of a WSDL and an operation name, this actor specializes its input and output ports to reflect the input and output parameters of the operation. For simple web service types, e.g. string, int, or double, this actor's ports are configured to the matching ptolemy type. Otherwise, the ports are set to XMLTOKEN. When this actor fires, it reads each input port, invokes the web service operation and sends the input data, and outputs the response to the output ports. The inputMechanism and outputMechanism parameters control the creation of helper actors for complex/nested web service types. (These parameters have no effect for simple web service types). By setting either to 'composite', a composite actor is created for each parameter that is complex/nested. Each composite actor is populated with necessary XML Assembler or XML Disassembler actors needed to build the nested web service type, and the composite actor ports are all simple ptolemy types. Changing the mechanism back to 'simple' deletes the connected helper actors. If you have made changes to the composite actors and don't want them lost, disconnect them from this actor before changing the mechanism to 'simple'. Limitations: Unused input ports on composite actors must have the corresponding internal links to XML Assembler actors removed. This is because the XML Assembler actors will read a token from each input port whose width is greater than 0. If the input to the web service contains an array of a nested type, the generated composite actors will use SequenceToArray with arrayLength defaulting to 1. For longer arrays, you must manually create them and send to the appropriate XML Assembler actor. Web service responses containing multi-reference values (elements refering to other elements for their content) are not handled. If the WSDL doesn't fully define the operation response, then the corresponding output port is set to XMLTOKEN. You can access the values by using XML Disassembler actor(s). A web service parameter with the WSDL type "any" sets the corresponding actor port type to XMLTOKEN. Multidimensional arrays not handled and the corresponding port is set to XMLTOKEN. You can get or set the values by using XML Assembler or Disassembler actor(s).

Parameters

wsdl	The web service WSDL address.
method	The web service method name to run.
inputMechanism	Setting to composite creates XML Assembler actors for complex (nested) input parameters.
outputMechanism	Setting to composite creates XML Disassembler actors for complex (nested) output parameters.
outputNil	If true, then each output port whose name is not a child element of the web service response XML document outputs a nil token.
username	The username for authentication.
password	The password for authentication.
timeout	The timeout in milliseconds.

Web Service Actor (org.sdm.spa.WebService)

Author: Ilkay Altintas **Version:** Unknown	The WebService actor executes a Web service, which is a computer program that runs on a remote host and communicates using a standardized protocol. The actor invokes the Web service and broadcasts the response through its output ports. The actor is intended to invoke any Web service with certain limitations: only base types may be input and output to the Web service. Base types are defined in the Web Service Description Language (WSDL) file that describes the Web service. WSDL is a format for describing network services--from simple eBay watcher services to complex distributed applications. For more information on WSDL, see http://www.w3.org/TR/wsdl. The WebService actor accepts the URL of a WSDL file and the name of an operation defined by that file. Once the user has selected a WSDL and operation, the actor automatically configures itself to perform that operation by creating the necessary input and output ports. Double-click the actor to start customization. To enter a WSDL URL that is not in the drop-down configuration menu, click the "Preferences" button on the configuration interface and change the type of the wsdUrl parameter to "Text". Then type in the WSDL to use; click "Commit" and double-click the actor again to reconfigure the list of available operations. Please do this every time the WSDL URL is updated.

Parameters

wsdlUrl	The URL of the Web service WSDL. Select a URL from the drop-down menu, or click the Properties button and change the type of the wsdlURL parameter to text to type in a new WSDL. Click Commit and double-click the actor again to reconfigure the list of available operations. Please do this every time the WSDL URL is updated.
methodName	The name of an operation defined by the WSDL (e.g., getXMLEntry or searchParam).
userName	The user name for the Web service, if necessary.
password	The password for the Web service, if necessary.
timeout	The amount of time to wait for the Web service to respond to a request.
hasTrigger	Activate the optional startTrigger port. Please activate the port ONLY when the actor has no input and the startTrigger port is required to schedule the actor.

Ports

startTrigger	An optional input port that has no declared type (in other words, the port can accept any data type: double, int, array, etc.) The port is activated by the hasTrigger parameter. Please enable the startTrigger port ONLY when the actor has no input and the trigger port is required to schedule the actor.
clientExecErrors	An output port that broadcasts execution errors (if any exist), or "NO ERRORS."

Wmsd Actor (org.eol.WmsdActor)

Author: no author given
Version: Unknown

The WmsdActor uses the Grid to process CPU-intensive calculations for the Encyclopedia of Life (EOL), which is working to predict the three-dimensional protein structures for all of the genomes that have been sequenced to date. Because this work requires tremendous computational resources, the actor uses the Grid, which allows people to share computing power, databases, and other tools. The actor uses a Workflow Management System daemon (WMSD) to submit Encyclopedia of Life (EOL) tasks to the Grid via an AppLeS Parameter Sweep Template (APST) program. APST manages the low-level complexities of job submission, heterogeneous resource management, and scheduling. The WMSD selects sequences from an input database, and continuously feeds many thousands of tasks to APST. Modeled protein structures are stored in an on-line, Internet-accessible database. EOL uses the Grid to predict 3D protein structures on entire genomes from homologous experimentally solved structures in support of rational drug design, ultimately in support of the pharmaceutical industry. See more about the Encyclopedia of Life project at SDSC at http://eol.sdsc.edu.

XML Assembler (org.sdm.spa.XMLAssembler)

Author: Daniel Crawl
Version: Unknown

On each firing, read one token from each input port and assemble them into an XML document where the root element name is specified by the output port name.

Parameters

inputNil	If true, then for each unconnected input port an element is created in the output document with an attribute nil whose value is "true".
encloseInputPortName	If true, then each token received will be added to an element with the name of the input port. By setting it false, input XML documents can be merged into a single document.

XML Disassembler (org.sdm.spa.XMLDisassembler)

Author: Daniel Crawl
Version: Unknown

This actor disassembles an XML document into its child elements. The input port name must match the document's root element name and is peeled off. Each child element is sent to the output port with the same name.

Parameters

outputNil	If true, then for each output port whose name is not a child element of the incoming XML document outputs a nil token.
arraysWrapped	If true, each element of an array is wrapped in an additional element.

XML To ADN Converter (org.geon.XMLToADN)

Author: Efrat Jaeger
Version: Unknown

The XMLToADN actor converts a string of XML name/value pairs to an ADN metadata file that describes an earth-systems dataset. The ADN schema is output as a string and can be applied to the dataset when it is registered via a Web service. ADN stands for ADEPT, DLESE, NASA--three organizations that agreed upon the ADN metadata standard to describe earth-systems datasets. For more information about ADN, see http://www.dlese.org/Metadata/adn-item/0.6.50/index.htm. The XMLToADN actor receives the URL of a dataset and a string of XML name/value pairs, which are generated earlier in the workflow. The actor outputs an ADN schema for use with the input dataset.

Ports

datasetURL	An input port that accepts a path to an earth-systems dataset that will use the generated ADN schema.
input	An input port that accepts XML name-value pairs input as a string.
output	An output port that broadcasts an ADN schema, output as a string. The schema can be applied to the input dataset when it is registered via a Web service.

XPath Processor (org.sdm.spa.XPath)

Author: Xiaowen **Version:** Unknown	The XPathProcessor is used to extract a specified tag element from an XML file. The actor outputs an array of selected XML tokens. The actor uses an XPath string, which provides an XML addressing syntax. XML is a markup language for files containing structured information (content as well as its context, e.g., a value and the fact that it is a "header" or a "footnote"). For more information about XML and XPaths, see http://www.w3.org/.

Parameters

xpath	An input port that accepts an xpath string used to identify XML elements (e.g., //tag_name)

Ports

input	An input port that accepts an XML token.
xpath	An input port that accepts an xpath string used to identify XML elements (e.g., //tag_name)
output	An output port that broadcasts an array of selected XML tokens.

XSLT Processor (org.sdm.spa.XSLTActor)

Author: Ilkay Altintas
Version: Unknown

The XSLTProcessor transforms an XML stream into an HTML stream that can be viewed using a BrowserUI actor. The actor performs the transformation using an XSLT file, which specifies how the document should be transformed. For more information about XSLT, see http://www.w3.org/TR/xslt. The actor receives an XML stream via its input port. Specify an XSLT file in the XSLTFilePath parameter. The actor outputs the transformed file as a string that can be displayed by the BrowserUI actor.

Parameters

XSLT File Path	The file path to the XSLT file used to transform the XML input into HTML.

Ports

xmlIn	An input port that accepts an XML input stream, as a string.
htmlOut	An output port that broadcasts an HTML stream, as a string.

XY Plotter (ptolemy.actor.lib.gui.XYPlotter)

Author: Jie Liu **Version:** Unknown	The XYPlotter actor plots (x,y) points and displays the graph. The actor reads x and y double tokens via its two input multiports. Each time the actor iterates, it reads one token from each input channel. X and Y tokens received on the first channel of the input port are plotted together, as are the tokens received on the second channels, etc. The two input port must have the same width (i.e., the same number of input channels).

Parameters

fillOnWrapup	Specify whether to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
startingDataset	The starting dataset number to which data is plotted. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the plot graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).

Ports

inputY	A multiport that accepts doubles representing y-values. The port must have the same width (i.e., number of input channels) as the inputX port.
inputX	A multiport that accepts one or more doubles representing x-values. The port must have the same width (i.e., number of input channels) as the inputY port.

XY Scope (ptolemy.actor.lib.gui.XYScope)

Author: Edward A. Lee **Version:** Unknown	The XYScope actor creates and displays an oscilloscope style graph. The actor reads x and y double tokens via its two input multiports. Each time the actor iterates, it reads one token from each input channel. The actor plots the (x,y) points in an oscilloscope style, meaning that the graphed points have a finite persistence. X and Y tokens received on the first channel of the input port are plotted together, as are the tokens received on the second channels, etc. The two input port must have the same width (i.e., the same number of input channels). The persistence parameter specifies the number of points displayed. For example, at the default persistence (100), any point older than 100 samples is erased and forgotten.

Parameters

fillOnWrapup	Specify whether to rescale the plot so that all data is visible. By default, the actor scales the plot to fill the display area.
startingDataset	The starting dataset number to which data is plotted. The value must be a non-negative integer. The default is 0.
legend	Annotations that will be displayed with the graph. Specify a comma-separated list of values that correspond to the input data sets (e.g., rainfall, temperature, elevation).
persistence	The number of samples from each input channel displayed at any one time (an integer). The default is 100.

Ports

inputY	A multiport that accepts doubles representing y-values. The port must have the same width (i.e., number of input channels) as the inputX port.
inputX	A multiport that accepts one or more doubles representing x-values. The port must have the same width (i.e., number of input channels) as the inputY port.

Zip Files (util.ZipFiles)

Author: Dan Higgins
Version: Unknown

The ZipFiles actor 'zips' multiple files into a single zipped archive. The actor reads a string of file names to be zipped and outputs the zip file name when finished.

Ports

zipFilenamesArray	An input port that accepts string of file names to be zipped.
zippedFileName	An input port that accepts the name to give to the zipped archive.
zippedFileResult	An output port that broadcasts the file name of the zipped archive.

CT Director (ptolemy.domains.ct.kernel. CTMixedSignalDirector)

The Continuous Time (CT) Director is designed to oversee workflows that predict how systems evolve as a function of time (i.e., "dynamic systems"). In CT workflows, the rates of change of parameters are related to the current value or rates of change of other parameters, often in complex and coupled ways that are described by differential equations. When a CT directed workflow is used in Kepler, tokens are passed from one actor to another just as they are in SDF or PN workflows. However, the CT Director keeps track of the "time" of each iteration as well as the time between each iteration (i.e., the "time step"). By insuring that the time step is small enough, the director can use simple extrapolations to estimate new values. The CT Director then iterates the workflow enough times to reach the desired stop time. The entire process is thus just numerical integration. In general, the relevance of the director's parameters varies depending on the type of ODE solver algorithm selected. If the algorithm is fixed-step (FowardEulerSolver and BackwardEulerSolver), the director will use the value specified by the initStepSize as the step size. The specified value is a 'guess' at an initial integration step size. If the integral does not look right, changing the initStepSize might provide a better result. For variable-step-size algorithms (ExplicitRK23Solver and ExplicitRK45Solver), step-size will change based on the rate of change of the original function's values (i.e., derivative values). In other words, time-steps within an integration will change throughout the calculation, and the initStepSize is used only as an initial suggestion. Directors with variable-step-size algorithms use the maxStepSize and minStepSize parameters to set upper and lower bounds for estimated step sizes. These parameters are used for adjusting tradeoffs between accuracy and performance. For simple dynamic systems, setting an upper bound with the maxStepSize parameter helps ensure that the algorithm will use an adequate number of time points. For more complex systems, the minStepSize ensures that the algorithm will not consume too many system resources by using increasingly minute step sizes. The minStepSize is also used for the first step after breakpoints. The timeResolution parameter is also used to adjust the tradeoff between accuracy and speed. In general, one would not change this parameter unless a function is known to change substantially in times of less than the parameter's default value, 1E-10 sec. The parameter helps ensure that variable-step-size algorithms do not use unnecessarily small time

Author: Jie Liu, Haiyang Zheng
Version:

Unknown	steps that would result is long execution times. Reducing the parameter's value might produce more accurate results, but at a performance cost. The errorTolerance parameter is only relevant to CT directors that use variable-step-size algorithms. Workflow actors that perform integration error control (e.g., the Integrator actor) will compare their estimated error to the value specified by the errorTolerance parameter. If the estimated error is greater than the errorTolerance, the director will decide that the step size is inaccurate and will decrease it. In most cases, the default value of the errorTolerance parameter (1e-4) does not require change. The startTime and stopTime parameters specify the initial and final time for the integration. By default, the time starts at 0 and runs to infinity. Note: the startTime and stopTime parameters are only applicable when the CT Director is at the top level. If a CT workflow is contained in another workflow, the CT Director will use the time of its executive director. The maxIterations specifies the number of times the director will iterate to determine a "fixed point." A fixed point is reached if two successive iteration steps produce the "same" result. How close values must be to be considered fixed, is specified with the valueResolution parameter, which defaults to 1e-6. The synchronizeToRealTime and runAheadLength parameters are advanced parameters that are genrally only used when a CT workflow is nested in another workflow. For example, if the CT Director is embedded in an event-based workflow (e.g., a workflow that uses a DE Director), the CT Director will "run ahead" of the global time by the amount specified by the runAheadLength parameter, and prepare to roll back if necessary. The local current time in the sub-workflow is compared with the current time of the executive director. If the local time is later than the global time, then the directed system will rollback to a "known good" state. The "known good" state is the state of the system at the time when local time is equal to the current time of the executive director. In general, the timeResolution and runAheadLength parameters should be left at their default values. For more information about the CT Director, see the Ptolemy II User Manual http://ptolemy.eecs.berkeley.edu/ papers/05/ptIIdesign3-domains/ptIIdesign3-domains.pdf.

Parameters

initStepSize	The initial integration step size. The value is a double that defaults to 0.1.
stopTime	The final time for the integration. By default, the time starts at 0 and runs to infinity.
startTime	Starting time of the integration. The value is a double that defaults to 0.0.
timeResolution	The timeResolution parameter is used to adjust the tradeoff between accuracy and speed. In general, one would not change this parameter unless a function is known to change substantially in times of less than the parameter's default value, 1E-10 sec. The parameter helps ensure that variable-step-size algorithms do not use unnecessarily small time steps that would result is long execution times. Reducing the parameter's value might produce more accurate results, but at a performance cost.
ODESolver	The class name of the normal ODE solver used. The default value is a string: "ptolemy.domains.ct.kernel.solver.ExplicitRK23Solver".
maxIterations	The maxIterations specifies the number of times the director will iterate to determine a "fixed point." A fixed point is reached if two successive iteration steps produce the "same" result. How close values must be to be considered fixed is specified with the valueResolution parameter.
runAheadLength	The runAheadLength parameter is an advanced parameter that is genrally only used when a CT workflow is nested in another workflow. For example, if the CT Director is embedded in an event-based workflow (e.g., a workflow that uses a DE Director), the CT Director will "run ahead" of the global time by the amount specified by the runAheadLength parameter, and prepare to roll back if necessary. The local current time in the sub-workflow is compared to the current time of the executive director. If the local time is later than the global time, then the directed system will rollback to a "known good" state. The "known good" state is the state of the system at the time when local time is equal to the current time of the executive director.
maxStepSize	The maxStepSize parameter sets an upper bound for estimated step sizes. The value is a double that defaults to 1.0.

errorTolerance	The errorTolerance parameter is only relevant to directors that use variable-step-size algorithms. Workflow actors that perform integration error control (e.g., the Integrator actor) will compare their estimated error to the value specified by the errorTolerance parameter. If the estimated error is greater than the errorTolerance, the director will decide that the step size is inaccurate and will decrease it. In most cases, the default value of the errorTolerance parameter (1e-4) does not require change
minStepSize	The minStepSize parameter sets a lower bound for estimated step sizes. The value is a double that defaults to 1e-5.
synchronizeToRealTime	Indicator whether the execution will synchronize to real time. The value is a Boolean token that defaults to false. The synchronizeToRealTime is an advanced parameter that is genrally only used when a CT workflow is nested in another workflow.
valueResolution	Value resolution specifies how close values must be to be considered fixed. The default is 1e-6.

DDF Director (ptolemy.domains.ddf.kernel.DDFDirector)

Author: Gang Zhou
Version: Unknown

The dynamic dataflow (DDF) domain is a superset of the synchronous dataflow(SDF) and Boolean dataflow(BDF) domains. In the SDF domain, an actor consumes and produces a fixed number of tokens per firing. This static information makes possible compile-time scheduling. In the DDF domain, there are few constraints on the production and consumption behavior of actors, and the schedulers make no attempt to construct a compile-time schedule. Instead, each actor has a set of firing rules (patterns) and can be fired if one of them is satisfied, i.e., one particular firing pattern forms a prefix of sequences of unconsumed tokens at input ports. The canonical actors in the DDF domain include Select and Switch, which consume or produce tokens on different channels based on the token received from the control port. (In practice, use DDFSelect and DDFBooleanSelect in the DDF-specific library instead of Select and BooleanSelect in the regular FlowControl library; however, Switch and BooleanSwitch in the regular FlowControl library can be used in DDF domain.) The dynamic scheduler implemented in this director fires all enabled and non-deferrable actors once in a basic iteration. A deferrable actor is one that will not help one of the downstream actors become enabled because that downstream actor either already has enough tokens on the channel connecting those two actors or is waiting for tokens on another channel. If no actor fires so far, which means there is no enabled and non-deferrable actor, then among all enabled and deferrable actors, this director fires those which have the smallest maximum number of tokens on their output channels which satisfy the demand of destination actors. If still no actor fires, then there is no enabled actor. A user can treat several such basic iterations as a single iteration by adding a parameter with name requiredFiringsPerIteration to an actor (which is often a sink actor or an actor directly connected to output port of the composite actor) and specifying the number of times this actor must be fired in a single iteration. If the value of the parameter runUntilDeadlockInOneIteration is a BooleanToken with value true, one single iteration consists of repeating the basic iteration until deadlock is reached (thus overriding the previous definition of one iteration), which is the status of the model where all active actors under the control of this director are unable to fire because their firing rules are not satisfied. However, they may be able to fire again during next iteration when tokens are transferred in from an outside domain. Note runUntilDeadlockInOneIteration can be set to true only when this director is not on the top level. The algorithm implementing one basic iteration goes like this: E = set of enabled actors D = set of deferrable enabled actors One basic(default) iteration consists of: if (E\D != empty set) { fire (E\D) } else if (D != empty set) { fire minimax (D) } else { declare deadlock } The function "minimax(D)" returns a subset of D with the smallest maximum number of tokens on their output channels which satisfy the demand of destination actors. Note that any SDF model can be run with a DDF Director. However, the notion of iteration is different. One could try to imitate the SDF iteration in the DDF domain by controlling the number of firings in one iteration for some actors, such as requiring a plotter to plot a fixed number of points in each iteration. In the DDF domain, the firing rule of any actor is specified by the token consumption rates of its input ports. A general DDF actor could change the consumption rates of its input ports after each firing of this actor. For multiports, an array token could be used to specify different rates for different channels connected to the same multiport. Note that in SDF, all channels connected to the same multiport have the same rate. Based on DDFSimpleSched in Ptolemy Classic, by Edward Lee. See E. A. Lee et al., "The Almagest," documentation for Ptolemy Classic, Vol. 1, Chapter 7, 1997.

Parameters

iterations	Specify the number of times a workflow is iterated. By default, this parameter is set to "0". Note that "0" does not mean "no iterations." Rather, "0" means that the workflow will iterate forever. Values greater than zero specify the actual number of times the director should execute the entire workflow.
maximumReceiverCapacity	A Parameter representing the maximum capacity of each receiver controlled by this director. This is an integer that defaults to 0, which means the queue in each receiver is unbounded. To specify bounded queues, set this to a positive integer.
runUntilDeadlockInOneIteration	A parameter indicating whether one iteration consists of repeated basic iterations until deadlock. If this parameter is true, the model will be executed until deadlock in one iteration. The default value is a BooleanToken with the value false. It cannot be set to true if this director is at the top level.

DE Director (ptolemy.domains.de.kernel.DEDirector)

Author: Lukito Muliadi, Edward A. Lee, Jie Liu, Haiyang Zheng
Version: Unknown

The Discrete Event (DE) Director, which oversees workflows where events occur at discrete times along a time line, is well suited for modeling time-oriented systems, such as queuing systems, communication networks, and occurrence rates or wait times. In DT workflows, actors send "event tokens," which consist of a data token and a time stamp. The director reads these tokens, and places each on a global, workflow timeline. Large event queues or queues that change often are "expensive" in terms of system resources and may have performance issues. All actors in a DE workflow must receive input tokens, even if the tokens are solely used as triggers. Once active, an actor will fire until it has no more tokens in its input ports, or until it returns false. Because DE actors only fire only after they receive their inputs, workflows that require loops (feeding an actor's output back into its input port for further processing) can cause "deadlock" errors. The deadlock errors occur because the actor depends on its own output value as an initial input. To fix this problem, use a TimedDelay actor to generate and inject an initial input token. The DE Director and each event in its workflow contain a tag that consists of a timestamp and additional information that helps the director determine when to process each event. On each iteration, the director will process all events with tags that are equal to its tag (the "model tag"), and then advance its model tag and perform a new set of matching events. Note that "model time" is not "real time." Model time starts from the time specified by startTime parameter, which has a default value of 0.0. The stop time is specified by the stopTime parameter, which has a default value of Infinity, meaning that the execution will run forever. Execution of a DE workflow ends when the timestamp of the earliest event exceeds the stop time. By default, execution also ends when the global event queue becomes empty. To prevent ending the execution when there are no more events (if your workflow relies on user interaction, for example), set the stopWhenQueueIsEmpty parameter to false. If the parameter synchronizeToRealTime is set to true, then the director will not process events until the real time elapsed since the workflow started matches the timestamp of the event. Synchronizing ensures that the director does not get ahead of real time; however, synchronizing does not ensure that the director keeps up with real time. The DE Director's timeResolution parameter is an advanced parameter that is only useful when the DE workflow contains composite components. In general, leave the parameter set to its default value ("1E-10").

Parameters

stopWhenQueueIsEmpty	Specify whether or not the execution stops when the queue is empty. By default, the director will stop when the queue is empty.
timeResolution	The time precision used by the director. All time values are rounded to the nearest multiple of this number. The value is a double token that defaults to "1E-10", which is 10-10. In general, leave the parameter set to its default value.
stopTime	The stop time of the workflow. The value is a double token that defaults to Infinity.
startTime	The start time of the workflow. The value is a double token that defaults to 0.0.
synchronizeToRealTime	Specify whether or not the execution should synchronize to real time. When synchronized with real time, the director will not process events until the elapsed real time matches the time stamp of the events. By default, the director does not synchronize to real time.

PN Director (ptolemy.domains.pn.kernel.PNDirector)

Author: Mudit Goel, Edward A. Lee, Xiaowen Xin **Version:** Unknown	PN Directors are natural candidates for managing workflows that require parallel processing on distributed computing systems. PN workflows are powerful because they have few restrictions. On the other hand, they can be very inefficient. The Process Network (PN) Director is similar to the SDF Director in that it does not have a notion of time. However, unlike the SDF Director, the PN Director does not statically calculate firing schedules. Instead, a PN workflow is driven by data availability: tokens are created on output ports whenever input tokens are available and the outputs can be calculated. Output tokens are passed to connected actors, where they are held in a buffer until that next actor collects all required inputs and can fire. The PN Director finishes executing a workflow only when there are no new data token sources anywhere in the workflow. The same execution process that gives the PN Director its flexibility can also lead to some unexpected results: workflows may refuse to automatically terminate because tokens are always generated and available to downstream actors, for example. If one actor fires at a much higher rate than another, a downstream actor's memory buffer may overflow, causing workflow execution to fail.

Parameters

initialQueueCapacity	The initial size of the queues for each communication channel. The value is an integer that defaults to 1. This is an advanced parameter that can usually be left at its default value.
maximumQueueCapacity	The maximum size of the queues for each communication channel. The value is an integer that defaults to 65536. To specify unbounded queues, set the value to 0. This is an advanced parameter that can usually be left at its default value.

SDF Director (ptolemy.domains.sdf.kernel.SDFDirector)

The SDF Director is often used to oversee fairly simple, sequential workflows in which the director can determine the order of actor invocation from the workflow. Types of workflows that would run well under an SDF Director include processing and reformatting tabular data, converting one data type to another, and reading and plotting a series of data points. A workflow in which an image is read, processed (rotated, scaled, clipped, filtered, etc.), and then displayed, is also an example of a sequential workflow that requires a director simply to ensure that each actor fires in the proper order (i.e., that each actor executes only after it receives its required inputs). The SDF Director is very efficient and will not tax system resources with overhead. However, this efficiency requires that certain conditions be met, namely that the data consumption and production rate of each actor in an SDF workflow be constant and declared. If an actor reads one piece of data and calculates and outputs a single result, it must always read and output a single token of data. This data rate cannot change during workflow execution and, in general, workflows that require dynamic scheduling and/or flow control cannot use this director. Additionally, the SDF Director has no understanding of passing time (at least by default), and actors that depend on a notion of time may not work as expected. For example, a TimedPlotter actor will plot all values at time zero when used in SDF. By default, the SDF Director requires that all actors in its workflow be connected. Otherwise, the director cannot account for concurrency between disconnected workflow parts. Usually, a PN Director should be used for workflows that contain disconnected actors; however, the SDF Director's allowDisconnectedGraphs parameter may also be set to true. The SDF Director will then schedule each disconnected "island" independently. The director cannot infer the sequential relationship between disconnected actors (i.e., nothing forces the director to finish executing all actors on one island before firing actors on another). However, the order of execution within each island should be correct. Usually, disconnected graphs in an SDF model indicate an error. Because SDF Directors schedule actors to fire only after they receive their inputs, workflows that require loops (feeding an actor's output back into its input port for further processing) can cause "deadlock" errors. The deadlock errors occur because the actor depends on its own output value as an initial input. To fix this problem, use a SampleDelay actor to generate and inject an initial input value into the workflow. The SDF Director determines the order in which actors

Author: Steve Neuendorffer

Version: Unknown	execute and how many times each actor needs to be fired to complete a single iteration of the workflow. This schedule is calculated BEFORE the director begins to iterate the workflow. Because the SDF Director calculates a schedule in advance, it is quite efficient. However, SDF workflows must be static. In other words, the same number of tokens must be consumed/produced at every iteration of the workflow. Workflows that require dynamic control structures, such as a BooleanSwitch actor that sends output on one of two ports depending on the value of a 'control', cannot be used with an SDF Director because the number of tokens on each output can change for each execution. Unless otherwise specified, the SDF Director assumes that each actor consumes and produces exactly one token per channel on each firing. Actors that do not follow the one-token-per-channel firing convention (e. g., Repeat or Ramp) must declare the number of tokens they produce or consume via the appropriate parameters. The number of times a workflow is iterated is controlled by the director's iterations parameter. By default, this parameter is set to "0". Note that "0" does not mean "no iterations." Rather, "0" means that the workflow will iterate forever. Values greater than zero specify the actual number of times the director should execute the entire workflow. A value of 1, meaning that the director will run the workflow once, is often the best setting when building an SDF workflow. The amount of data processed by an SDF workflow is a function of both the number of times the workflow iterates and the value of the director's vectorizationFactor parameter. The vectorizationFactor is used to increase the efficiency of a workflow by increasing the number of times actors fire each time the workflow iterates. If the parameter is set to a positive integer (other than 1), the director will fire each actor the specified number of times more than normal. The default is 1, indicating that no vectorization should be performed. Keep in mind that changing the vectorizationFactor parameter changes the meaning of a nested SDF workflow and may cause deadlock in a workflow that uses it. The SDF Director has several advanced parameters that are generally only relevant when an SDF workflow contains composite components. In most cases the period, timeResolution, synchronizeToRealTime, allowRateChanges, timeResolution, and constrainBufferSizes parameters can be left at their default values. For more information about the SDF Director, see the Ptolemy documentation (http://ptolemy.eecs.berkeley.edu/papers/05/ ptIIdesign3-domains/ptIIdesign3-domains.pdf).

Parameters

allowDisconnectedGraphs	Specify whether to allow disconnected actors in the workflow (by default, all actors are required to be connected). If disconnected actors are permitted, the SDF Director will schedule each disconnected 'island' independently. Nothing "forces" the director to finish executing all actors on one island before firing actors on another. However, the order of execution within each island should be correct. Usually, disconnected graphs in an SDF workflow indicate an error.
allowRateChanges	Specify whether dynamic rate changes are permitted or not. By default, rate changes are not permitted, and the director will perform a check to disallow such workflows. If the parameter is selected, then workflows that require rate parameters to be modified during execution are valid, and the SDF Director will dynamically compute a new schedule at runtime. This is an advanced parameter that can usually be left at its default value.
timeResolution	The time precision used by this director. All time values are rounded to the nearest multiple of this number. The value is a double that defaults to "1E-10" (which is 10-10). This is an advanced parameter that can usually be left at its default value.
constrainBufferSizes	Specify whether buffer sizes are fixed. By default, buffers are fixed, and attempts to write to the buffer that cause the buffer to exceed its scheduled size result in an error. This is an advanced parameter that can usually be left at its default value.
iterations	Specify the number of times a workflow is iterated. By default, this parameter is set to "0". Note that "0" does not mean "no iterations." Rather, "0" means that the workflow will iterate forever. Values greater than zero specify the actual number of times the director should execute the entire workflow. A value of 1, meaning that the director will run the workflow once, is often the best setting when building an SDF workflow.
vectorizationFactor	The vectorizationFactor is used to increase the efficiency of a workflow by increasing the number of times actors fire each time the workflow iterates. If the parameter is set to a positive integer (other than 1), the director will fire each actor the specified number of times more than normal. The default is 1, indicating that no vectorization should be performed. Keep in mind that changing the vectorizationFactor parameter changes the meaning of a nested SDF workflow and may cause deadlock in a workflow that uses it.

synchronizeToRealTime	Specify whether the execution should synchronize to real time or not. By default, the director does not synchronize to real time. If synchronize is selected, the director will only process the workflow when elapsed real time matches the product of the period parameter and the iteration count. Note: if the period parameter has a value of 0.0 (the default), then selecting this parameter has no effect. This is an advanced parameter that can usually be left at its default value.
period	The time period of each iteration. The value is a double that defaults to 0.0, which means that the director does not increment workflow time. If the value greater than 0.0, the actor will increment workflow time each time it fires. This is an advanced parameter that can usually be left at its default value.

Alphabetical List of Actors in the Kepler Standard Library

- ANOVA
- ASCToRAW
- AbsoluteValue
- Accumulator
- AddGrids
- AddorSubtract
- Annotation
- ArchiveCounter
- ArrayAccumulator
- ArrayAverage
- ArrayContainsElement
- ArrayElement
- ArrayExtract
- ArrayLength
- ArrayLevelCrossing
- ArrayMaximum
- ArrayMinimum
- ArrayPeakSearch
- ArrayPlotter
- ArrayRemoveElement
- ArraySort
- ArrayToElements
- ArrayToSequence
- AscGridVal
- Authentication
- Average
- Babel
- BarGraph
- Barplot
- BasicUnits
- Bernoulli
- BinaryFileReader
- BinaryFileWriter
- BooleanAccumulator
- BooleanMultiplexor
- BooleanSwitch

- BooleanToAnything
- Boxplot
- BrowserDisplay
- BrowserUI
- CGSUnitBase
- CVHulltoRaster
- CartesianToComplex
- CartesianToPolar
- ClimateChangeFileProcessor
- ClimateFileProcessor
- CloseDatabaseConnection
- ColorParameter
- CommandLineExec
- Comparator
- ComplexToCartesian
- ComplexToPolar
- CompositeActor
- ConcatenateArrays
- Constant
- ConvertImageToString
- ConvertURLToImage
- ConvexHull
- Correlation
- Counter
- CurrentTime
- DDFBooleanSelect
- DarwinCoreDataSource
- DataGroup
- DatabaseQuery
- DatabaseWriter
- DifferentialEquation
- DirectoryListing
- DirectoryMaker
- DiscreteRandomNumberGenerator
- Display
- Distributor
- DocViewer
- Documentation

- DotProduct
- EML2Dataset
- ENMPCP
- ESRIShapeFileDisplayer
- EcogridWriter
- ElectronicUnitBase
- ElementsToArray
- Ellipse
- EmailSender
- EndGAMESSInput
- Equals
- ExperimentMonitor
- ExperimentPreparator
- ExperimentStarter
- Expression
- ExpressionReader
- ExpressionToToken
- ExternalExecution
- FTPClient
- FileCopier
- FileCopy
- FileExistenceMonitor
- FileFetcher
- FileListSequencer
- FileLocationChooser
- FileNameChooser
- FileParameter
- FileReader
- FileStager
- FileToArrayConverter
- FileWriter
- FilterUI
- ForkResourceAdder
- FormattedGroup
- GAMESSAtomDataExtractor
- GAMESSInputGenerator
- GAMESSKeywords
- GAMESSLocalRun

- GAMESSNimrodRun
- GARPAlgorithm
- GARPPresampleLayers
- GARPSummary
- GDALFormatTranslator
- GDALWarpandProjection
- GMLDisplayer
- GUIRunCIPRes
- GarpPrediction
- GaussianDistributionRandomNumberGenerator
- Get2DPoint
- GetPoint
- GlobusJob
- GlobusProxy
- GrassBuffer
- GrassHull
- GrassRaster
- GridFTP
- GridOverlay
- GridRescaler
- GridReset
- GriddlesExec
- GriddlesInputFile
- GriddlesOutputFile
- GriddlesParameter
- IJMacro
- Image
- ImageContrast
- ImageConverter
- ImageDisplay
- ImageJ
- ImageReader
- ImageRotate
- Initializer
- IntRangeParameter
- Integrator
- InteractiveShell
- Interpolator

- IsPresent
- JCOGGridFTP
- JCOGPROXYExec
- JCOGWorkflowExec
- JGridletCreator
- JobCreator
- JobGetDirectory
- JobGetRealJobID
- JobManager
- JobRemover
- JobStatus
- Line
- JobSubmitter
- KeywordGroup
- Limiter
- LineReader
- LineWriter
- LinearModel
- Logger
- LogicFunction
- LongToDouble
- LookupTable
- MappedLogger
- MatlabExpression
- MatrixViewer
- Maximum
- MergeGrids
- MessageDigestTest
- MetadataSource
- Minimum
- MoleculeArrayProducer
- MoleculeSelector
- MonitorImage
- MonitorValue
- MultiplyorDivide
- NextDiagram
- NexusFileReader
- NondeterministicMerge

- NonstrictTest
- ObjectToRecordConverter
- OpenBabel
- OpenDatabaseConnection
- OrbImageSource
- OrbPacketObjectSource
- OrbWaveformSink
- OrbWaveformSource
- PAUPInfer
- Parameter
- ParameterSet
- ParameterizedGlobusJob
- Pause
- PhyloDataReader
- PointInPolygon
- PointInPolygonXY
- PolarToCartesian
- PolarToComplex
- Polygon
- PolygonDiagramsDataset
- PolygonDiagramsTransition
- PortParameter
- PtMatlabExpression
- QMViewDisplay
- Quantizer
- RExpression
- RMean
- RMedian
- RQuantile
- Ramp
- RandomNormal
- RandomUniform
- ReadTable
- RecIDCM3
- RecordAssembler
- RecordDisassembler
- RecordUpdater
- Recorder

- Rectangle
- Regression
- Remainder
- Repeat
- RequireVersion
- Rescaler
- RicianDistributionRandomNumberGenerator
- Round
- RunCompositeActor
- RunJobGridClient
- SI
- SProxy
- SRBAddMetadata
- SRBConnect
- SRBCreateQueryConditions
- SRBCreateQueryInterface
- SRBGetMetadata
- SRBGetPhysicalLocation
- SRBProxyCommands
- SRBQueryMetadata
- SRBSGet
- SRBSPut
- SRBStreamGet
- SRBStreamPut
- SSHtoExecute
- SVGConcatenate
- SVGToPolygonConverter
- SampleDelay
- Scale
- Scatterplot
- ScopeExtendingAttribute
- Select
- SequencePlotter
- SequenceToArray
- ServerExecute
- ShowLocations
- SimpleFileReader
- Sinewave

- SoaplabAnalysis
- SoaplabChooseOperation
- SoaplabChooseResultType
- SoaplabServiceStarter
- SshDirectoryCreator
- SshDirectoryList
- SshExecuteCmd
- SshFileCopier
- SshFileRemover
- SshSession
- StartGAMESSInput
- StatusChecker
- Stop
- StringAccumulator
- StringCompare
- StringConstant
- StringFunction
- StringIndexOf
- StringLength
- StringMatches
- StringParameter
- StringReplace
- StringSplitter
- StringSubstring
- StringToImageConverter
- StringToInt
- StringToLong
- StringToPolygonConverter
- StringToXML
- SubsetChooserActor
- Summary
- SummaryStatistics
- Switch
- SyncOnTerminator
- TempActor
- TemporaryScriptCreator
- Test
- TextFileWriter

- ThrowException
- ThrowModelError
- TimeStamp
- TimedPlotter
- TimedScope
- TokenCounter
- TokenDuplicator
- TokenToExpression
- TokenToStringConverter
- TokentoSeparateChannelsTransmitter
- TransitiveClosureDatabaseQuery
- TreeDecomposer
- TreeImprover
- TreeMerger
- TreeParser
- TreeToString
- TreeVizForester
- TrigFunction
- TypeTest
- URLToLocalFile
- UnitConverter
- UniformDistributionRandomNumberGenerator
- UpdatedGridFTP
- UserInteractiveShell
- VariableSetter
- VectorAssembler
- VectorDisassembler
- WSWithComplexTypes
- WebService
- WmsdActor
- XMLAssembler
- XMLDisassembler
- XMLToADNConverter
- XPathProcessor
- XSLTProcessor
- XYPlotter
- XYScope
- ZipFiles

- CTDirector
- DDFDirector
- DEDirector
- PNDirector
- SDFDirector