SONARIS/Framework

SAO Catalogue

Version 1.4.3

ORIMOS S.A.

Innere Gueterstrasse 4
6304 Zug (Switzerland)

Phone	+41-41-725-3570
Fax	+41-41-725-3579
Web	www.ORIMOS.com
Email	info@ORIMOS.com

ORIMOS Group
Berlin · Frankfurt · London · Zurich

Contents

1 Foundation Objects

File:	`saozoo`
Library Version:	1.0
Framework Version:	1.4
No. SAO types:	265

Description:

Fundamental SAOs

1.1 Array Types

The Array Types category contains the following SAOs:

- basicByteArray
- basicIntArray
- basicInt64Array
- basicDoubleArray
- basicStringArray
- basicDateArray
- stringArrayMerge
- goodStringArrayMerge
- intLookup
- stringLookup
- byteArrayIndex
- intArrayIndex
- int64ArrayIndex
- doubleArrayIndex
- stringArrayIndex
- dateArrayIndex
- intArrayBuild
- int64ArrayBuild
- doubleArrayBuild
- stringArrayBuild
- dateArrayBuild
- dynamicIntArrayBuild
- dynamicInt64ArrayBuild
- dynamicDoubleArrayBuild
- dynamicStringArrayBuild
- dynamicDateArrayBuild

1.1.1 basicByteArray

Functional Type:	7
Data Type:	SAF_BYTE_ARRAY
No Inputs:	

Description:

Constant Byte Array

1.1.2 basicIntArray

Functional Type:	8
Data Type:	SAF_INT_ARRAY
No Inputs:	

Description:

Constant 32 bit Integer Array

1.1.3 basicInt64Array

Functional Type:	9
Data Type:	SAF_INT64_ARRAY
No Inputs:	

Description:

Constant 64 bit Integer Array

1.1.4 basicDoubleArray

Functional Type:	10
Data Type:	SAF_DOUBLE_ARRAY
No Inputs:	

Description:

Constant Floating Point Double Array

1.1.5 basicStringArray

Functional Type:	12
Data Type:	SAF_STRING_ARRAY
No Inputs:	

Description:

Constant String Array

1.1.6 basicDateArray

Functional Type:	13
Data Type:	SAF_DATE_ARRAY
No Inputs:	

Description:

Constant Date Array

1.1.7 stringArrayMerge

Functional Type:	42
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Input* *Type:* SAF_STRING_ARRAY *Code:* SAF_OPTIONAL

Description:

Output is combination of all SAF_VALID inputs

1.1.8 goodStringArrayMerge

Functional Type:	43
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Input* *Type:* SAF_STRING_ARRAY *Code:* SAF_OPTIONAL

Description:

Output is combination of all inputs when all inputs are SAF_VALID

1.1.9 intLookup

Functional Type:	98
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Target* *Type:* SAF_INT *Code:* SAF_MANDATORY
Input 1:	*Array* *Type:* SAF_INT_ARRAY *Code:* SAF_MANDATORY

Description:

Search for Target in Array and output the index of the target

1.1.10 stringLookup

Functional Type:	99
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Target* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 1:	*Array* *Type:* SAF_STRING_ARRAY *Code:* SAF_MANDATORY

Description:

Search for Target in Array and output the index of the target

1.1.11 byteArrayIndex

Functional Type:	100
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Index* *Type:* SAF_INT *Code:* SAF_MANDATORY
Input 1:	*Array* *Type:* SAF_BYTE_ARRAY *Code:* SAF_MANDATORY

Description:

Array indexer into a byte array producing an INT

1.1.12 intArrayIndex

Functional Type:	101
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Index*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	*Array*
	Type: SAF_INT_ARRAY
	Code: SAF_MANDATORY

Description:

Array indexer into an int array producing an INT

1.1.13 int64ArrayIndex

Functional Type:	102
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Index*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	*Array*
	Type: SAF_INT64_ARRAY
	Code: SAF_MANDATORY

Description:

Array indexer into a INT64 array producing an INT64

1.1.14 doubleArrayIndex

Functional Type:	103
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Index*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	*Array*
	Type: SAF_DOUBLE_ARRAY
	Code: SAF_MANDATORY

Description:

Array indexer into a double array producing a DOUBLE

1.1.15 stringArrayIndex

Functional Type:	104
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Index* *Type:* SAF_INT *Code:* SAF_MANDATORY
Input 1:	*Array* *Type:* SAF_STRING_ARRAY *Code:* SAF_MANDATORY

Description:

Array indexer into a string array producing a STRING

1.1.16 dateArrayIndex

Functional Type:	105
Data Type:	SAF_DATE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Index* *Type:* SAF_INT *Code:* SAF_MANDATORY
Input 1:	*Array* *Type:* SAF_DATE_ARRAY *Code:* SAF_MANDATORY

Description:

Array indexer into a date array producing a DATE

1.1.17 intArrayBuild

Functional Type:	106
Data Type:	SAF_INT_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	 *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Turn available inputs into an int array. All connected inputs must be available

1.1.18 int64ArrayBuild

Functional Type:	107
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into an int64 array. All connected inputs must be available

1.1.19 doubleArrayBuild

Functional Type:	108
Data Type:	SAF_DOUBLE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into a double array. All connected inputs must be available

1.1.20 stringArrayBuild

Functional Type:	109
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into a string array. All connected inputs must be available

1.1.21 dateArrayBuild

Functional Type:	239
Data Type:	SAF_DATE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into a date array. All connected inputs must be available

1.1.22 dynamicIntArrayBuild

Functional Type:	241
Data Type:	SAF_INT_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into an int array. Ignores unavailable inputs

1.1.23 dynamicInt64ArrayBuild

Functional Type:	242
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into an int64 array. Ignores unavailable inputs

1.1.24 dynamicDoubleArrayBuild

Functional Type:	243
Data Type:	SAF_DOUBLE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into a double array. Ignores unavailable inputs

1.1.25 dynamicStringArrayBuild

Functional Type:	244
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into a string array. Ignores unavailable inputs

1.1.26 dynamicDateArrayBuild

Functional Type:	245
Data Type:	SAF_DATE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Turn available inputs into a date array.Ignores unavailable inputs

1.2 Date and Time

The Date and Time category contains the following SAOs:

- Date
- Timer to Date
- ExTimer to Date
- Time
- DateOnly
- High Resolution Timer2
- System Time
- Date to Seconds
- Time Comparator
- Seconds to Date
- TimeOnly
- Period Timer
- Date to String
- ExDate
- ExGMT to Local
- Date to Milliseconds
- High Resolution Timer

1.2.1 Date

Functional Type:	120
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	0

Description:

Sets output to the current date and time whenever an input changes

1.2.2 System Time

Functional Type:	121
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	0

Description:

Sets output to the current system time in milliseconds whenever an input changes

1.2.3 Date to String

Functional Type:	122
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Date to be formatted* *Type:* SAF_DATE *Code:* SAF_MANDATORY
Input 1:	*Optional control - default dd* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Produces multiple date formats, 0: dd.mm.yyyy 2:dd/mm/yyyy 3:dd/mm/yy 10:mm/yy. Add 100: hh:mm:ss 200:hh:mm:ss.xxx. Add 1000 for UTC.

Input 1: ***Optional control - default dd***

mm.yyyy

1.2.4 Timer to Date

Functional Type:	123
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Milliseconds in the epoch* *Type:* SAF_INT64 *Code:* SAF_MANDATORY

Description:

Converts an interval timer, or milliseconds in the epoch, to a SAF_DATE.

1.2.5 Date to Seconds

Functional Type:	124
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Date to be converted* *Type:* SAF_DATE *Code:* SAF_MANDATORY

Description:

Converts to seconds in epoch (i. e. since 1970)

1.2.6 ExDate

Functional Type:	125
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	0

Description:

No longer supported. Same as Date

1.2.7 ExTimer to Date

Functional Type:	126
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Milliseconds in the epoch* *Type:* SAF_INT64 *Code:* SAF_MANDATORY

Description:

No longer supported. Same as Timer to Date

1.2.8 Time Comparator

Functional Type:	127
Data Type:	SAF_DATE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Source timer* *Type:* SAF_DATE *Code:* SAF_MANDATORY
Input 1:	*Seconds allowed* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Changes value to that of first input if this value is close to current time or when initialised. Uses the second input, in seconds (default 10), to decide how close.

1.2.9 ExGMT to Local

Functional Type:	128
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Date to be converted* *Type:* SAF_DATE *Code:* SAF_MANDATORY

Description:

Redundant SAO. Copies its input.

1.2.10 Time

Functional Type:	129
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	0

Description:

Sets output to the current time whenever an input changes

1.2.11 Seconds to Date

Functional Type:	157
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Seconds in the epoch*
	Type: SAF_INT64
	Code: SAF_MANDATORY

Description:

Converts seconds in the epoch to a SAF_DATE.

1.2.12 Date to Milliseconds

Functional Type:	158
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Date to be converted*
	Type: SAF_DATE
	Code: SAF_MANDATORY

Description:

Converts to milliseconds in epoch (i. e. since 1970)

1.2.13 DateOnly

Functional Type:	165
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Date to be converted*
	Type: SAF_DATE
	Code: SAF_MANDATORY

Description:

Removes any time part leaving only the date

1.2.14 TimeOnly

Functional Type:	166
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Date to be converted*
	Type: SAF_DATE
	Code: SAF_MANDATORY

Description:

Removes any date part leaving only the time

1.2.15 High Resolution Timer

Functional Type:	174
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Sets output to a number in microseconds whenever an input changes

1.2.16 High Resolution Timer2

Functional Type:	175
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Reset*
	Type: SAF_INT64
	Code: SAF_OPTIONAL
Input 1:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Sets output to the number of microseconds since the last use of the reset input whenever any other input changes

1.2.17 Period Timer

Functional Type:	176
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Period start*
	Type: SAF_INT64
	Code: SAF_MANDATORY
Input 1:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Accurate high resolution timer setting its output to the difference between the first input (which should be a High Resolution Timer) and now whenever any input (including the first) changes

1.3 Event and Logging

The Event and Logging category contains the following SAOs:

- fullEvent
- userName
- audit
- lastChangedTimestamp
- userId

1.3.1 fullEvent

Functional Type:	185
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	6
Input 0:	*Control string -see String Processing/format SAO* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 1:	*Sender - field of event* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 2:	*EventId - 3 character field of event* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 3:	*Subject - field of event* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 4:	*Severity - “debug”, “info”, “warning” or “error” (default is info)* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 5:	*Data input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Generates an event whenever an input changes. Text is produced from the control string and data inputs exactly like the String Processing/format SAO. Generates an event on startup or when inputs are connected/disconnected. Output is text of event

1.3.2 audit

Functional Type:	186
Data Type:	SAF_VOID
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Generates an audit event whenever an input changes. Does not generate an event on startup or when inputs are connected/disconnected.

1.3.3 userId

Functional Type:	187
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Obtains the userId of the last user that changed the input.

1.3.4 userName

Functional Type:	188
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_INT *Code:* SAF_MANDATORY

Description:

Obtains the userName of the userId on the x input.

1.3.5 lastChangedTimestamp

Functional Type:	189
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Obtains the timestamp of the last change on the x input.

1.4 Flow Control

The Flow Control category contains the following SAOs:

- dateTriggerOnChange
- compareStrings
- goodInputCounter
- doubleSwitch
- intPriority
- boolPriority
- dateSwitch
- integer64Trigger
- stringTrigger
- doubleInRange
- doubleExRange
- stringArraySwitch
- tickCounter
- doubleDelay
- intArrayDelay
- byteArrayDelay
- delayCoordinator
- integer64TriggerOnChange
- lastChangedIndex
- doubleLastChanged
- isValid
- compareZero
- compareStringsInsensitive
- integerSwitch
- boolSwitch
- int64Priority
- stringPriority
- datePriority
- doubleTrigger
- integerInRange
- integerExRange
- doubleArraySwitch
- intArraySwitch
- intDelay
- boolDelay
- int64ArrayDelay
- stringArrayDelay
- dateDelay
- doubleTriggerOnChange
- intLastChanged
- stringLastChanged
- compare
- compareValues
- storeValues
- integer64Switch
- stringSwitch
- doublePriority
- doubleSequencer
- integerTrigger
- boolTrigger
- integer64InRange
- integer64ExRange
- dateArraySwitch
- int64ArraySwitch
- int64Delay
- stringDelay
- doubleArrayDelay
- dateArrayDelay
- intTriggerOnChange
- stringTriggerOnChange
- int64LastChanged
- dateLastChanged

1.4.1 dateTriggerOnChange

Functional Type:	33
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Propagate input only if changed

1.4.2 compareZero

Functional Type:	70
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

1 if x less than zero, 2 if equal zero, 3 if greater. Numeric comparison with zero suitable for input into switch SAO.

1.4.3 compareValues

Functional Type:	71
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

1 if x less than y; 2 if x equals y; 3 if x greater than y. Numeric comparison suitable for input into switch SAO.

1.4.4 compareStrings

Functional Type:	72
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_STRING *Code:* SAF_MANDATORY

Description:

1 if x less than y; 2 if x equals y; 3 if x greater than y. Case sensitive alphabetic comparison suitable for input into switch SAO.

1.4.5 compareStringsInsensitive

Functional Type:	73
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_STRING *Code:* SAF_MANDATORY

Description:

1 if x less than y; 2 if x equals y; 3 if x greater than y. Case insensitive alphabetic comparison suitable for input into switch SAO.

1.4.6 storeValues

Functional Type:	76
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Trigger Input* *Type:* SAF_INT64 *Code:* SAF_OPTIONAL
Input 1:	*Value* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Copies inputs to children when trigger changes from one good value to a different one. Copies values in parallel - input x to child x-1. Output is time of last copy. Does nothing if input and child numbers do not correspond.

1.4.7 goodInputCounter

Functional Type:	77
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output is number of SAF_VALID inputs. Will be SAF_STALE unless all connected inputs are SAF_VALID

1.4.8 integerSwitch

Functional Type:	80
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.9 integer64Switch

Functional Type:	81
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.10 doubleSwitch

Functional Type:	82
Data Type:	SAF_DOUBLE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.11 boolSwitch

Functional Type:	83
Data Type:	SAF_BOOL
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n* *Type:* SAF_INT *Code:* SAF_MANDATORY
Input 1:	*Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.12 stringSwitch

Functional Type:	84
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n* *Type:* SAF_INT *Code:* SAF_MANDATORY
Input 1:	*Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.13 intPriority

Functional Type:	85
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Uses first Valid input or, if none, first Stale input

1.4.14 int64Priority

Functional Type:	86
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Uses first Valid input or, if none, first Stale input

1.4.15 doublePriority

Functional Type:	87
Data Type:	SAF_DOUBLE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Uses first Valid input or, if none, first Stale input

1.4.16 boolPriority

Functional Type:	88
Data Type:	SAF_BOOL
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Uses first Valid input or, if none, first Stale input

1.4.17 stringPriority

Functional Type:	89
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Uses first Valid input or, if none, first Stale input

1.4.18 doubleSequencer

Functional Type:	92
Data Type:	SAF_DOUBLE
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*Trigger*
	Type: SAF_INT64
	Code: SAF_MANDATORY
Input 1:	*Trigger reduction factor - zero is stop*
	Type: SAF_INT
	Code: SAF_OPTIONAL
Input 2:	*Any change restarts sequence*
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Cycles through valid children copying value on input 0 change

1.4.19 dateSwitch

Functional Type:	95
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.20 datePriority

Functional Type:	96
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Uses first Valid input or, if none, first Stale input

1.4.21 integerTrigger

Functional Type:	130
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propogate x when y changes

1.4.22 integer64Trigger

Functional Type:	131
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propogate x when y changes

1.4.23 doubleTrigger

Functional Type:	132
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propogate x when y changes

1.4.24 boolTrigger

Functional Type:	133
Data Type:	SAF_BOOL
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propogate x when y changes

1.4.25 stringTrigger

Functional Type:	134
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propogate x when y changes

1.4.26 integerInRange

Functional Type:	135
Data Type:	SAF_INT
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 2:	*z* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value of x if (x >= y) and (x <= z) else status of SAF_INVALID

1.4.27 integer64InRange

Functional Type:	136
Data Type:	SAF_INT64
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 2:	*z* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value of x if (x >= y) and (x <= z) else status of SAF_INVALID

1.4.28 doubleInRange

Functional Type:	137
Data Type:	SAF_DOUBLE
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 2:	*z* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value of x if (x >= y) and (x <= z) else status of SAF_INVALID

1.4.29 integerExRange

Functional Type:	138
Data Type:	SAF_INT
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 2:	*z* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value of x if (x < y) or (x > z) else status of SAF_INVALID

1.4.30 integer64ExRange

Functional Type:	139
Data Type:	SAF_INT64
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 2:	*z* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value of x if (x < y) or (x > z) else status of SAF_INVALID

1.4.31 doubleExRange

Functional Type:	140
Data Type:	SAF_DOUBLE
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 2:	*z* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value of x if (x < y) or (x > z) else status of SAF_INVALID

1.4.32 doubleArraySwitch

Functional Type:	141
Data Type:	SAF_DOUBLE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_DOUBLE_ARRAY
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.33 dateArraySwitch

Functional Type:	142
Data Type:	SAF_DATE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_DATE_ARRAY
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.34 stringArraySwitch

Functional Type:	143
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_STRING_ARRAY
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.35 intArraySwitch

Functional Type:	144
Data Type:	SAF_INT_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_INT_ARRAY
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.36 int64ArraySwitch

Functional Type:	145
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Selector in the range 1-n*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 1:	
	Type: SAF_INT64_ARRAY
	Code: SAF_OPTIONAL

Description:

First input selects output 1-n

1.4.37 tickCounter

Functional Type:	169
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	3
Input 0:	*Initial Value*
	Type: SAF_INT
	Code: SAF_OPTIONAL
Input 1:	*Reset*
	Type: SAF_INT
	Code: SAF_OPTIONAL
Input 2:	*Data*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Counts the number of times any Data input changes. Will reset to the initial value (or 0) whenever an input is added or removed. Will reset the count if the Reset input changes.

1.4.38 intDelay

Functional Type:	203
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.39 int64Delay

Functional Type:	204
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.40 doubleDelay

Functional Type:	205
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.41 boolDelay

Functional Type:	206
Data Type:	SAF_BOOL
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.42 stringDelay

Functional Type:	207
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.43 intArrayDelay

Functional Type:	208
Data Type:	SAF_INT_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_INT_ARRAY
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.44 int64ArrayDelay

Functional Type:	209
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_INT64_ARRAY
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.45 doubleArrayDelay

Functional Type:	210
Data Type:	SAF_DOUBLE_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_DOUBLE_ARRAY
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.46 byteArrayDelay

Functional Type:	211
Data Type:	SAF_BYTE_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_BYTE_ARRAY *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.47 stringArrayDelay

Functional Type:	212
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_STRING_ARRAY *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.48 dateArrayDelay

Functional Type:	213
Data Type:	SAF_DATE_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_DATE_ARRAY *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.49 delayCoordinator

Functional Type:	214
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*timer* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*one of the inputs* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Synchronises many inputs to coordinate delays. The output is expected to be used as the trigger input of a set of delay SAOs. Each SAO to be coordinated is connected as an input to the corresponding delay SAOs and to one of the inputs on this SAO. The value of this SAO is the number of sets of input changes that have been merged into the tick.

1.4.50 dateDelay

Functional Type:	215
Data Type:	SAF_DATE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_DATE *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Ignore changes of x until next change of y

1.4.51 intTriggerOnChange

Functional Type:	216
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propagate input only if changed

1.4.52 integer64TriggerOnChange

Functional Type:	217
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propagate input only if changed

1.4.53 doubleTriggerOnChange

Functional Type:	218
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propagate input only if changed

1.4.54 stringTriggerOnChange

Functional Type:	219
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Propagate input only if changed

1.4.55 lastChangedIndex

Functional Type:	233
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Generates index of last changed input or highest numbered input

1.4.56 intLastChanged

Functional Type:	234
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Propagates last changed input or highest numbered input

1.4.57 int64LastChanged

Functional Type:	235
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Propagates last changed input or highest numbered input

1.4.58 doubleLastChanged

Functional Type:	236
Data Type:	SAF_DOUBLE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Propagates last changed input or highest numbered input

1.4.59 stringLastChanged

Functional Type:	237
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Propagates last changed input or highest numbered input

1.4.60 dateLastChanged

Functional Type:	238
Data Type:	SAF_DATE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Propagates last changed input or highest numbered input

1.4.61 isValid

Functional Type:	249
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Checks if the input is valid - result: 1 = valid, 2 = invalid

1.4.62 compare

Functional Type:	250
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Checks if input1 $>$ input2 (out=1), input 1 = input 2 (out=2), input 1 $<$ input 2 (out=3)

1.5 Resource

The Resource category contains the following SAOs:

- int
- int64
- double
- bool
- date
- string
- intArray
- int64Array
- doubleArray
- stringArray
- dateArray

1.5.1 int

Functional Type:	254
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 1:	*Provider of Resource Interface* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.2 int64

Functional Type:	255
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 1:	*Provider of Resource Interface* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.3 double

Functional Type:	256
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected*
	Type: SAF_VOID
	Code: SAF_RESERVED
Input 1:	*Provider of Resource Interface*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.4 bool

Functional Type:	257
Data Type:	SAF_BOOL
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected*
	Type: SAF_VOID
	Code: SAF_RESERVED
Input 1:	*Provider of Resource Interface*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.5 date

Functional Type:	258
Data Type:	SAF_DATE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected*
	Type: SAF_VOID
	Code: SAF_RESERVED
Input 1:	*Provider of Resource Interface*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.6 string

Functional Type:	259
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 1:	*Provider of Resource Interface* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.7 intArray

Functional Type:	260
Data Type:	SAF_INT_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 1:	*Provider of Resource Interface* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.8 int64Array

Functional Type:	261
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected*
	Type: SAF_VOID
	Code: SAF_RESERVED
Input 1:	*Provider of Resource Interface*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.9 doubleArray

Functional Type:	262
Data Type:	SAF_DOUBLE_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected*
	Type: SAF_VOID
	Code: SAF_RESERVED
Input 1:	*Provider of Resource Interface*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.10 stringArray

Functional Type:	263
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected*
	Type: SAF_VOID
	Code: SAF_RESERVED
Input 1:	*Provider of Resource Interface*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.5.11 dateArray

Functional Type:	264
Data Type:	SAF_DATE_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Parent if second input not connected* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 1:	*Provider of Resource Interface* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output from Multivalued calculation SAO. The input is expected to support the ISAOResource interface. This SAO will make it's parents it's input providing the parent is registered with the resource manager. Connecting the second input will disable this link.

1.6 Simple Data Types

The Simple Data Types category contains the following SAOs:

- basicInt
- basicInt64
- basicDouble
- basicBool
- basicDate
- basicString
- basicAmorphous
- basicVoid
- selfEmptyingFolder
- convertToInt
- convertToInt64
- convertToDouble
- convertToBool
- convertToString
- convertToDate
- convertToAmorphous

1.6.1 basicInt

Functional Type:	0
Data Type:	SAF_INT
No Inputs:	

Description:

Constant 32 bit integer

1.6.2 basicInt64

Functional Type:	1
Data Type:	SAF_INT64
No Inputs:	

Description:

Constant 64 bit integer

1.6.3 basicDouble

Functional Type:	2
Data Type:	SAF_DOUBLE
No Inputs:	

Description:

Constant Floating Point Double

1.6.4 basicBool

Functional Type:	3
Data Type:	SAF_BOOL
No Inputs:	

Description:

Constant boolean

1.6.5 basicDate

Functional Type:	4
Data Type:	SAF_DATE
No Inputs:	

Description:

Constant date and/or time

1.6.6 basicString

Functional Type:	5
Data Type:	SAF_STRING
No Inputs:	

Description:

Constant string not convertible to a number

1.6.7 basicAmorphous

Functional Type:	6
Data Type:	SAF_UNDEFINED
No Inputs:	

Description:

Constant amorphous object

1.6.8 basicVoid

Functional Type:	11
Data Type:	SAF_VOID
No Inputs:	

Description:

Object taking no value

1.6.9 selfEmptyingFolder

Functional Type:	14
Data Type:	SAF_VOID
No Inputs:	

Description:

Object taking no value and which deletes all children at startup

1.6.10 convertToInt

Functional Type:	15
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Convert input to 32 bit integer

1.6.11 convertToInt64

Functional Type:	16
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Convert input to 64 bit integer

1.6.12 convertToDouble

Functional Type:	17
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Convert input to Floating Point Double

1.6.13 convertToBool

Functional Type:	18
Data Type:	SAF_BOOL
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Convert input to bool

1.6.14 convertToString

Functional Type:	19
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Convert input to string

1.6.15 convertToDate

Functional Type:	32
Data Type:	SAF_DATE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Convert input to Date

1.6.16 convertToAmorphous

Functional Type:	240
Data Type:	SAF_UNDEFINED
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*input*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Allows an amorphous SAO to be produced from any basic type

1.7 Simple Maths

The Simple Maths category contains the following SAOs:

- integerAdd
- integerSubtract
- integerMultiply
- integerDivide
- intMinus
- intAbs
- exponential
- log10
- floor
- rand
- int64Delta
- intModulo
- integer64Modulo
- integer64Add
- integer64Subtract
- integer64Multiply
- integer64Divide
- integer64Minus
- integer64Abs
- log
- sqrt
- fmod
- average
- doubleDelta
- intIncrement
- integer64Increment
- doubleAdd
- doubleSubtract
- doubleMultiply
- doubleDivide
- doubleMinus
- doubleAbs
- power
- ceil
- round
- intDelta
- roundToDP
- intDecrement
- integer64Decrement

1.7.1 integerAdd

Functional Type:	20
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x+y producing a 32 bit integer

1.7.2 integer64Add

Functional Type:	21
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x+y producing a 64 bit integer

1.7.3 doubleAdd

Functional Type:	22
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x+y producing a floating point double

1.7.4 integerSubtract

Functional Type:	23
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x-y producing a 32 bit integer

1.7.5 integer64Subtract

Functional Type:	24
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x-y producing a 64 bit integer

1.7.6 doubleSubtract

Functional Type:	25
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	x *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	y *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x-y producing a floating point double

1.7.7 integerMultiply

Functional Type:	26
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	x *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	y *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x multiplied by y producing a 32 bit integer

1.7.8 integer64Multiply

Functional Type:	27
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	x *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	y *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x multiplied by y producing a 64 bit integer

1.7.9 doubleMultiply

Functional Type:	28
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

x multiplied by y producing a floating point double

1.7.10 integerDivide

Functional Type:	29
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

x/y producing a 32 bit integer

1.7.11 integer64Divide

Functional Type:	30
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

x/y producing a 64 bit integer

1.7.12 doubleDivide

Functional Type:	31
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

x/y producing a floating point double

1.7.13 intMinus

Functional Type:	55
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value is the integer -x

1.7.14 integer64Minus

Functional Type:	56
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Value is the 64 bit integer -x

1.7.15 doubleMinus

Functional Type:	57
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Value is a floating point double -x

1.7.16 intAbs

Functional Type:	58
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Value is the integer abs(x)

1.7.17 integer64Abs

Functional Type:	59
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Value is the 64 bit integer abs(x)

1.7.18 doubleAbs

Functional Type:	60
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Value is the floating point double abs(x)

1.7.19 exponential

Functional Type:	110
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates e raised to the power of x.

1.7.20 log

Functional Type:	111
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the natural logarithm (base e) of x.

1.7.21 power

Functional Type:	112
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates x to the power y.

1.7.22 log10

Functional Type:	113
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the logarithm (base 10) of x.

1.7.23 sqrt

Functional Type:	114
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the square root of x.

1.7.24 ceil

Functional Type:	115
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates a double value representing the smallest integer that is greater than or equal to x.

1.7.25 floor

Functional Type:	116
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates a double value representing the largest integer that is less than or equal to x

1.7.26 fmod

Functional Type:	117
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the floating-point remainder of x / y

1.7.27 round

Functional Type:	118
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Rounds x by y. Calculates the multiple of y closest to x

1.7.28 rand

Functional Type:	119
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Trigger* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*Resets sequence using given value* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Generates a pseudo random number whenever the trigger changes

1.7.29 average

Functional Type:	190
Data Type:	SAF_DOUBLE
Maximum Inputs:	4
Validated Inputs:	4
Input 0:	*Observation* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*Number of Observations (default 100)* *Type:* SAF_INT *Code:* SAF_OPTIONAL
Input 2:	*Reset when changed* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 3:	*Control (currently unused)* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Calculates the average value of the first input. Also provides a MINIMUM and MAXIMUM SAF_DOUBLE resource. Will be STALE until the observation count is reached

1.7.30 intDelta

Functional Type:	200
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates x minus the previous value of x

1.7.31 int64Delta

Functional Type:	201
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates x minus the previous value of x

1.7.32 doubleDelta

Functional Type:	202
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates x minus the previous value of x

1.7.33 roundToDP

Functional Type:	230
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Rounds x to y decimal places if y given; just copies input if not

1.7.34 intModulo

Functional Type:	246
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

x % y after converting to int, if y has no input it defaults to 2

1.7.35 intIncrement

Functional Type:	247
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Increment x after converting to int if necessary

1.7.36 intDecrement

Functional Type:	248
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Decrement x after converting to int if necessary

1.7.37 integer64Modulo

Functional Type:	251
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

x % y after converting to 64 bit int, if y has no input it defaults to 2

1.7.38 integer64Increment

Functional Type:	252
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Increment x after converting to 64 bit int if necessary

1.7.39 integer64Decrement

Functional Type:	253
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Decrement x after converting to 64 bit int if necessary

1.8 String Processing

The String Processing category contains the following SAOs:

- inputName
- directConcatenation
- spacedConcatenation
- directMerge
- spacedMerge
- format
- rawstring
- stringComparison
- caseInsensitiveStringComparison
- substring
- stringlength
- fullPath

1.8.1 inputName

Functional Type:	44
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output is name of input

1.8.2 directConcatenation

Functional Type:	45
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Joins two strings, x and y, together directly

1.8.3 spacedConcatenation

Functional Type:	46
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Joins two strings, x and y, together with a space between

1.8.4 directMerge

Functional Type:	47
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Joins all strings directly

1.8.5 spacedMerge

Functional Type:	48
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	
	Type: SAF_UNDEFINED
	Code: SAF_OPTIONAL

Description:

Joins all strings together with a space between

1.8.6 format

Functional Type:	49
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Control string* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 1:	 *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Creates a string from multiple inputs. Similar to **printf** . Text is copied and substrings starting `%` are replaced by text from the corresponding input. Format is `%`<**flags**><**letter**> . A `*` in flags (used to specify a length) is not supported and will be unconditionally replaced by `10` . Otherwise, flags are passed to printf except in `D` , `T` and `Z` .

- `U`
  Use a type (`F` , `S` , `D` , `I` , `L` , or `B`) appropriate to the input.
- `F` , `E` and `G`
  Floating point f, e and g.
- `S`
  A string.
- `D`
  A date.
- `T`
  A time.
- `Z`
  A date and time.
- `I`
  A 32 bit integer.
- `L`
  A 64 bit integer.
- `X`
  As `L` , but in hexadecimal.
- `B`
  A boolean.
- `K`
  Skips an input.

- R
  Reuses an input.

- C
  All subsequent inputs with no data available are printed with a default value, either a zero or an empty string. This is the default state.

- A
  All subsequent inputs with no data available are printed as a status value without regard to any other flags e.g. “SAF_INVALID”.

- Q
  All subsequent inputs with no data available are printed as no output at all without regard to any other flags.

Lower case have the same meaning as the upper case value. If in doubt use `%U %U %U` for the number of inputs.

1.8.7 rawstring

Functional Type:	50
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Extracts the raw string from x

1.8.8 stringComparison

Functional Type:	51
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Sets its output to 1 is x is alphabetically less than y, 2 if x equals y and 3 if x is greater than y

1.8.9 caseInsensitiveStringComparison

Functional Type:	52
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY
Input 1:	*y*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Sets its output to 1 is x is alphabetically less than y, 2 if x equals y and 3 if x is greater than y without regard to case

1.8.10 substring

Functional Type:	53
Data Type:	SAF_STRING
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*input*
	Type: SAF_STRING
	Code: SAF_MANDATORY
Input 1:	*maximum length of output*
	Type: SAF_INT
	Code: SAF_MANDATORY
Input 2:	*offset in input at which to start*
	Type: SAF_INT
	Code: SAF_OPTIONAL

Description:

Takes a subset of characters from the input at an optional offset

1.8.11 stringlength

Functional Type:	54
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_STRING
	Code: SAF_MANDATORY

Description:

Output is string length as integer

1.8.12 fullPath

Functional Type:	180
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Output is full path of input. If input is a proxy then SAF_BADINPUT; if no input then its own full path

1.9 System

The System category contains the following SAOs:

- machine
- domain
- getenv
- osVersion
- processMemUsage
- pid
- currentDirectory
- putenv
- memoryUse
- SAFVersion
- user
- application
- cpuUse
- processCpu

1.9.1 machine

Functional Type:	155
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its value to the name of the machine on which it is running.

1.9.2 pid

Functional Type:	156
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its value to the process id of the machine on which it is running.

1.9.3 user

Functional Type:	159
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its value to the name of the user under which it is running.

1.9.4 domain

Functional Type:	160
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its value to the domain of the user under which it is running.

1.9.5 currentDirectory

Functional Type:	161
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Sets its output to that of the current directory. This SAO will not notice a change in current directory unless and until it rechecks due to an input change. It ignores the value of any inputs

1.9.6 application

Functional Type:	162
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its value to the name of the application.

1.9.7 getenv

Functional Type:	163
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Sets its value to that of the environment using its name. The input can be used to refresh the value whenever it changes

1.9.8 putenv

Functional Type:	164
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Value*
	Type: SAF_STRING
	Code: SAF_OPTIONAL

Description:

Sets the environment value using its name to the input value if it is available. With no input works like getenv. Warning: changing the environment can have unexpected side effects

1.9.9 cpuUse

Functional Type:	167
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Whenever the input changes sets its output to the cpu utilization of the machine since the last change

1.9.10 osVersion

Functional Type:	168
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its output to details of the version of the operating system on which it is running

1.9.11 memoryUse

Functional Type:	170
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input*
	Type: SAF_VOID
	Code: SAF_OPTIONAL

Description:

Whenever the input changes sets its output to the number of MBytes of memory currently allocated to all processes.

1.9.12 processCpu

Functional Type:	171
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Whenever the input changes sets its output to the cpu utilization of the process since the last change

1.9.13 processMemUsage

Functional Type:	172
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Any input* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Whenever the input changes sets its output to the number of KBytes of memory currently allocated to the process.

1.9.14 SAFVersion

Functional Type:	173
Data Type:	SAF_STRING
No Inputs:	

Description:

Unconditionally sets its output to a string showing the compiled version of the safsystem library

1.10 Timers

The Timers category contains the following SAOs:

- IntervalTimer
- IntervalTimer2
- TimeOfDayTimer
- TimeOfDayTimer2
- OneShot Timer

1.10.1 IntervalTimer

Functional Type:	151
Data Type:	SAF_INT64
No Inputs:	

Description:

Updates its output every 'n' milliseconds. The name of the SAO takes the form [D][M]n [comment]. D, if present, allows the time to drift i.e. the next period end is calculated from the end rather than the start of the previous one. M, if present, creates a meldable timer i.e. one where ticks may be merged if the machine is too busy to process them all. The period in milliseconds is the SAO's name converted to binary i.e. these SAOs should have names like '250', '1000', '237000 long gap' 'M1000' etc. The SAO's value is the number of milliseconds since the epoch (Jan 1 1970) at each tick.

1.10.2 IntervalTimer2

Functional Type:	152
Data Type:	SAF_INT64
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Ticks every x secs

1.10.3 TimeOfDayTimer

Functional Type:	153
Data Type:	SAF_DATE
No Inputs:	

Description:

Has a name in the form x[,y]. Ticks with a period in minutes specified by x. One of these ticks is chosen so that it occurs y seconds after midnight. x and y default to 60 and 3600 respectively.

1.10.4 TimeOfDayTimer2

Functional Type:	154
Data Type:	SAF_DATE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 1:	*y* *Type:* SAF_DATE *Code:* SAF_OPTIONAL

Description:

Ticks with a period in minutes specified by x. It synchronises so that the time in y specifies the time of one of these ticks on the first day.

1.10.5 OneShot Timer

Functional Type:	177
Data Type:	SAF_INT
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Period in milliseconds* *Type:* SAF_INT *Code:* SAF_OPTIONAL
Input 1:	*Trigger* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Output is always zero except for a timed period after the trigger input changes when it is one. The timer will not operate because of a connection, disconnection or at startup. It will only react to a change in the value of the trigger input, although this change could occur either side of a disconnection. Any change of input during the timed period is totally ignored.

1.11 Trigonometric Functions

The Trigonometric Functions category contains the following SAOs:

- acos
- asin
- atan
- atan2
- cos
- cosh
- sin
- sinh
- tan
- tanh

1.11.1 acos

Functional Type:	220
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Calculates arccosine of x

1.11.2 asin

Functional Type:	221
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Calculates arcsine of x

1.11.3 atan

Functional Type:	222
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x*
	Type: SAF_UNDEFINED
	Code: SAF_MANDATORY

Description:

Calculates the arctangent of x in the range of -pi/2 to pi/2 radians

1.11.4 atan2

Functional Type:	223
Data Type:	SAF_DOUBLE
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*y* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the arctangent of y/x in the range -pi to pi radians

1.11.5 cos

Functional Type:	224
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the cosine of x

1.11.6 cosh

Functional Type:	225
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the hyperbolic cosine of x

1.11.7 sin

Functional Type:	226
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the sine of x

1.11.8 sinh

Functional Type:	227
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the hyperbolic sine of x

1.11.9 tan

Functional Type:	228
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the tangent of x

1.11.10 tanh

Functional Type:	229
Data Type:	SAF_DOUBLE
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*x* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY

Description:

Calculates the hyperbolic tangent of x

2 Base Algorithms

File:	`basealgorithms`
Library Version:	1.0
Framework Version:	1.4
No. SAO types:	3

Description:

Basic mathematical algorithms

2.1 Uncategorized SAOs

The following SAOs are not assigned to any category within this library:

- Mean
- Median
- Interpolation

2.1.1 Mean

Functional Type:	0
Data Type:	SAF_DOUBLE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Single data point* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Calculates mean of all valid inputs

2.1.2 Median

Functional Type:	1
Data Type:	SAF_DOUBLE
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	1
Input 0:	*Single data point* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Calculates median of all valid inputs

2.1.3 Interpolation

Functional Type:	2
Data Type:	SAF_DOUBLE
Maximum Inputs:	42
Validated Inputs:	42
Input 0:	*x* *Type:* SAF_DOUBLE *Code:* SAF_MANDATORY
Input 1:	*Interpolation method, if empty linear* *Type:* SAF_INT *Code:* SAF_OPTIONAL
Input 2:	*x0* *Type:* SAF_DOUBLE *Code:* SAF_OPTIONAL
Input 3:	*y0* *Type:* SAF_DOUBLE

Code: SAF_OPTIONAL

Input 4: *x1*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 5: *y1*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 6: *x2*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 7: *y2*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 8: *x3*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 9: *y3*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 10: *x4*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 11: *y4*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 12: *x5*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 13: *y5*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 14: *x6*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 15: *y6*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 16: *x7*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 17: *y7*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 18: *x8*
Type: SAF_DOUBLE
Code: SAF_OPTIONAL

Input 19: *y8*
Type: SAF_DOUBLE

	Code: SAF_OPTIONAL
Input 20:	*x9*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 21:	*y9*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 22:	*x10*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 23:	*y10*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 24:	*x11*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 25:	*y11*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 26:	*x12*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 27:	*y12*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 28:	*x13*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 29:	*y13*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 30:	*x14*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 31:	*y14*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 32:	*x15*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 33:	*y15*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 34:	*x16*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 35:	*y16*
	Type: SAF_DOUBLE

	Code: SAF_OPTIONAL
Input 36:	*x17*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 37:	*y17*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 38:	*x18*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 39:	*y18*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 40:	*x19*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL
Input 41:	*y19*
	Type: SAF_DOUBLE
	Code: SAF_OPTIONAL

Description:

Calculates interpolated value

3 External System Interface

File:	`eszoo`
Library Version:	1.2
Framework Version:	1.4
No. SAO types:	19

Description:

External System data retrieval and contribution interface SAOs. An External System is an implementation of either or both the *ExternalSystemRetriever (ESR)* and *ExternalSystemSender (ESS)* interfaces. Hierarchies created using these SAOs may be used to access data available from an External System (via the ESR interface) and contribute values back to that system (via the ESS interface).

3.1 Array

The Array category contains the following SAOs:

- Incoming Double Array
- Incoming Date Array
- Incoming Int Array
- Incoming Int64 Array

3.1.1 Incoming Double Array

Functional Type:	8
Data Type:	SAF_DOUBLE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 2:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 3:	*Pseudo Parent* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Receives double array data from an External System. Maintains a value based on a single item of data supplied by the External System. Can serve as a parent to other SAOs in nested data structures and can contribute supplied values back to the External System when used in conjunction with an Outgoing SAO in a similar way to the Contributing SAO.

Input 0: ***Configuration Parameters***

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-r` *depth*
  The setting of some options can be passed recursively down a hierarchy of *Incoming* SAOs. Setting the *depth* value will control how far down a hierarchy such options should inherit. The default is `0` . Setting to *-1* results in infinite inheritance. Options which may be recursively inherited are marked as such below.

- `-c` *[t|p|u][Mm]*
  Automatically create children as indicated by *Child List* information supplied by the ESR. The `p` modifier makes such new child SAOs permanent. The `t` modifier indicates the creation of transient SAOs. The `u` modifier causes the 'uncreation' of child SAOs which

may then subsequently be recreated using the `p` or `t` modifiers. The `m` modifier makes new child SAOs created via the `p` or `t` modifiers 'managed' such that if the ESR removes a child name from the Child List the corresponding child SAO and any tree beneath it will be deleted. The `M` modifier disables the effect of the `m` modifier. This option and its modifiers can be recursively inherited depending on the setting of the `-r` option.

- `-F`
  Enable failover. Reestablishes subscription to the data if a SAF_DROPPED status is received which indicates that no further updates can be supplied. This is enabled by default. This option can be recursively inherited depending on the setting of the `-r` option.

- `-f`
  Disable failover (reverse of `-F` , also inheritable).

- `-s` *suffix*
  The *suffix* string is used as an extension to the name of the SAO for the purposes of the SAO's subscription to an item of data in the ESR. This may be convenient where the desired behaviour would otherwise require the creation of a number of placeholder SAOs simply to achieve a complex hierarchical name (e.g. an SQL query subscription). Double quotes may be used to supply space separated words, double quotes within strings may be escaped with backslash (\ ).

- `-S` *suffix*
  Works the same as the `-s` option, but the *suffix* will be used to replace rather than extend the name of the SAO.

- `-n` *new-name*
  In contrast to the `-S` option, this will cause the name of the SAO to actually be changed to the *new-name* supplied.

- `-m`
  Disables the `meldable` property of this SAO to ensure that every update delivered to it via the container is supplied (i.e. no update merging is performed in the container queue for this SAO). This may be desirable for updates to certain critical subscriptions, and is essential for "query" type subscriptions which result in a stream of updates which represent a result set (e.g. a database query) as merging updates will corrupt the stream.

- `-M`
  Enables the `meldable` property of this SAO. This is enabled by default (reverse of `-m` .

- `-i`
  Treat updates received from a subscription as independent to the current subscription. Ordinarily any update received must result from the current subscription i.e. not result from a previous subscription which has been cancelled. To ensure this, no resubscription is attempted until any existing subscription has completed or been cancelled and that cancellation has been confirmed. Further, any updates which are received that do not match the current subscription are discarded. This flag suppresses these checks. It also allows a new subscription request to be submitted before the current one has completed. This is useful in situations where the ESR will complete all subscriptions but cannot cancel subscriptions once they are made or where a series of subscription requests will be submitted.

- `-I`
  Enforce update checks and cancellations (default, the reverse of `-i`).

- `-q` *suffix*
  A shorthand for the `-mfiS` series of flag settings commonly used for "query" style subscriptions which produce update streams that will be processed by SAOs in the *Stream* category of this library.

- `-v` *[s|v|i]*
  Translate any SAF_STALE updates received according the the modifier specified. The *s* modifier causes SAF_STALE updates to be stored as SAF_STALE (the default). The *v* modifier causes SAF_STALE updates to be stored as SAF_VALID, and the *i* modifier causes SAF_STALE updates to be stored as SAF_INVALID. This option can be recursively inherited depending on the setting of the `-r` option.

Input 3: ***Pseudo Parent***

Any SAO supplied on this input will be treated in preference to the parent of this SAO for the purposes of establishing the hieararchical position of this SAO when initiating a subscription to data from the ESR. This allows instances of this SAO to be placed outside the main hierarchy of SAOs related to a particular ESR without the need to form a new connection to the ESR.

Input 4: ***Contribution Value***

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.1.2 Incoming Date Array

Functional Type:	9
Data Type:	SAF_DATE_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 2:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 3:	*Pseudo Parent* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Receives date array data from an External System. Maintains a value based on a single item of data supplied by the External System. Can serve as a parent to other SAOs in nested data structures and can contribute supplied values back to the External System when used in conjunction with an Outgoing SAO in a similar way to the Contributing SAO.

Input 0: ***Configuration Parameters***

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-r` *depth*
  The setting of some options can be passed recursively down a hierarchy of *Incoming* SAOs. Setting the *depth* value will control how far down a hierarchy such options should inherit. The default is `0` . Setting to *-1* results in infinite inheritance. Options which may be recursively inherited are marked as such below.

- `-c` *[t|p|u][Mm]*
  Automatically create children as indicated by *Child List* information supplied by the ESR. The `p` modifier makes such new child SAOs permanent. The `t` modifier indicates the creation of transient SAOs. The `u` modifier causes the 'uncreation' of child SAOs which may then subsequently be recreated using the `p` or `t` modifiers. The `m` modifier makes new child SAOs created via the `p` or `t` modifiers 'managed' such that if the ESR removes a child name from the Child List the corresponding child SAO and any tree beneath it will be deleted. The `M` modifier disables the effect of the `m` modifier. This option and its modifiers can be recursively inherited depending on the setting of the `-r` option.

- `-F`
  Enable failover. Reestablishes subscription to the data if a SAF_DROPPED status is received which indicates that no further updates can be supplied. This is enabled by default. This option can be recursively inherited depending on the setting of the `-r` option.

- `-f`
  Disable failover (reverse of `-F` , also inheritable).

- `-s` *suffix*
  The *suffix* string is used as an extension to the name of the SAO for the purposes of the SAO's subscription to an item of data in the ESR. This may be convenient where the desired behaviour would otherwise require the creation of a number of placeholder SAOs simply to achieve a complex hierarchical name (e.g. an SQL query subscription). Double quotes may be used to supply space separated words, double quotes within strings may be escaped with backslash (\ ).

- `-S` *suffix*
  Works the same as the `-s` option, but the *suffix* will be used to replace rather than extend the name of the SAO.

- `-n` *new-name*
  In contrast to the `-S` option, this will cause the name of the SAO to actually be changed to the *new-name* supplied.

- `-m`
  Disables the `meldable` property of this SAO to ensure that every update delivered to it via the container is supplied (i.e. no update merging is performed in the container queue for this SAO). This may be desirable for updates to certain critical subscriptions, and is essential for "query" type subscriptions which result in a stream of updates which represent a result set (e.g. a database query) as merging updates will corrupt the stream.

- `-M`
  Enables the `meldable` property of this SAO. This is enabled by default (reverse of `-m` .

- `-i`
  Treat updates received from a subscription as independent to the current subscription. Ordinarily any update received must result from the current subscription i.e. not result from a previous subscription which has been cancelled. To ensure this, no resubscription is attempted until any existing subscription has completed or been cancelled and that cancellation has been confirmed. Further, any updates which are received that do not match the current subscription are discarded. This flag suppresses these checks. It also allows a new subscription request to be submitted before the current one has completed. This is useful in situations where the ESR will complete all subscriptions but cannot cancel subscriptions once they are made or where a series of subscription requests will be submitted.

- `-I`
  Enforce update checks and cancellations (default, the reverse of `-i`).

- `-q` *suffix*
  A shorthand for the `-mfiS` series of flag settings commonly used for "query" style subscriptions which produce update streams that will be processed by SAOs in the *Stream* category of this library.

- `-v` *[s|v|i]*
  Translate any SAF_STALE updates received according the the modifier specified. The *s* modifier causes SAF_STALE updates to be stored as SAF_STALE (the default). The *v* modifier causes SAF_STALE updates to be stored as SAF_VALID, and the *i* modifier causes SAF_STALE updates to be stored as SAF_INVALID. This option can be recursively inherited depending on the setting of the `-r` option.

Input 3: *Pseudo Parent*

Any SAO supplied on this input will be treated in preference to the parent of this SAO for the purposes of establishing the hieararchical position of this SAO when initiating a subscription to data from the ESR. This allows instances of this SAO to be placed outside the main hierarchy of SAOs related to a particular ESR without the need to form a new connection to the ESR.

Input 4: *Contribution Value*

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.1.3 Incoming Int Array

Functional Type:	10
Data Type:	SAF_INT_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 2:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 3:	*Pseudo Parent* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Receives int array data from an External System. Maintains a value based on a single item of data supplied by the External System. Can serve as a parent to other SAOs in nested data structures and can contribute supplied values back to the External System when used in conjunction with an Outgoing SAO in a similar way to the Contributing SAO.

Input 0: ***Configuration Parameters***

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-r` *depth*
  The setting of some options can be passed recursively down a hierarchy of *Incoming* SAOs. Setting the *depth* value will control how far down a hierarchy such options should inherit. The default is `0` . Setting to *-1* results in infinite inheritance. Options which may be recursively inherited are marked as such below.

- `-c` *[t|p|u][Mm]*
  Automatically create children as indicated by *Child List* information supplied by the ESR. The `p` modifier makes such new child SAOs permanent. The `t` modifier indicates the creation of transient SAOs. The `u` modifier causes the 'uncreation' of child SAOs which may then subsequently be recreated using the `p` or `t` modifiers. The `m` modifier makes new child SAOs created via the `p` or `t` modifiers 'managed' such that if the ESR removes a child name from the Child List the corresponding child SAO and any tree beneath it will be deleted. The `M` modifier disables the effect of the `m` modifier. This option and its modifiers can be recursively inherited depending on the setting of the `-r` option.

- `-F`

Enable failover. Reestablishes subscription to the data if a SAF_DROPPED status is received which indicates that no further updates can be supplied. This is enabled by default. This option can be recursively inherited depending on the setting of the `-r` option.

- `-f`
  Disable failover (reverse of `-F` , also inheritable).

- `-s` *suffix*
  The *suffix* string is used as an extension to the name of the SAO for the purposes of the SAO's subscription to an item of data in the ESR. This may be convenient where the desired behaviour would otherwise require the creation of a number of placeholder SAOs simply to achieve a complex hierarchical name (e.g. an SQL query subscription). Double quotes may be used to supply space separated words, double quotes within strings may be escaped with backslash (\ ).

- `-S` *suffix*
  Works the same as the `-s` option, but the *suffix* will be used to replace rather than extend the name of the SAO.

- `-n` *new-name*
  In contrast to the `-S` option, this will cause the name of the SAO to actually be changed to the *new-name* supplied.

- `-m`
  Disables the `meldable` property of this SAO to ensure that every update delivered to it via the container is supplied (i.e. no update merging is performed in the container queue for this SAO). This may be desirable for updates to certain critical subscriptions, and is essential for "query" type subscriptions which result in a stream of updates which represent a result set (e.g. a database query) as merging updates will corrupt the stream.

- `-M`
  Enables the `meldable` property of this SAO. This is enabled by default (reverse of `-m` .

- `-i`
  Treat updates received from a subscription as independent to the current subscription. Ordinarily any update received must result from the current subscription i.e. not result from a previous subscription which has been cancelled. To ensure this, no resubscription is attempted until any existing subscription has completed or been cancelled and that cancellation has been confirmed. Further, any updates which are received that do not match the current subscription are discarded. This flag suppresses these checks. It also allows a new subscription request to be submitted before the current one has completed. This is useful in situations where the ESR will complete all subscriptions but cannot cancel subscriptions once they are made or where a series of subscription requests will be submitted.

- `-I`
  Enforce update checks and cancellations (default, the reverse of `-i`).

- `-q` *suffix*
  A shorthand for the `-mfiS` series of flag settings commonly used for "query" style subscriptions which produce update streams that will be processed by SAOs in the *Stream* category of this library.

- `-v` *[s|v|i]*
  Translate any SAF_STALE updates received according the the modifier specified. The *s* modifier causes SAF_STALE updates to be stored as SAF_STALE (the default). The *v* modifier causes SAF_STALE updates to be stored as SAF_VALID, and the *i* modifier causes SAF_STALE updates to be stored as SAF_INVALID. This option can be recursively inherited depending on the setting of the `-r` option.

Input 3: *Pseudo Parent*

Any SAO supplied on this input will be treated in preference to the parent of this SAO for the purposes of establishing the hieararchical position of this SAO when initiating a subscription to data from the ESR. This allows instances of this SAO to be placed outside the main hierarchy of SAOs related to a particular ESR without the need to form a new connection to the ESR.

Input 4: *Contribution Value*

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.1.4 Incoming Int64 Array

Functional Type:	11
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 2:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 3:	*Pseudo Parent* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Receives int64 array data from an External System. Maintains a value based on a single item of data supplied by the External System. Can serve as a parent to other SAOs in nested data structures and can contribute supplied values back to the External System when used in conjunction with an Outgoing SAO in a similar way to the Contributing SAO.

Input 0: ***Configuration Parameters***

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-r` *depth*
  The setting of some options can be passed recursively down a hierarchy of *Incoming* SAOs. Setting the *depth* value will control how far down a hierarchy such options should inherit. The default is `0` . Setting to *-1* results in infinite inheritance. Options which may be recursively inherited are marked as such below.

- `-c` *[t|p|u][Mm]*
  Automatically create children as indicated by *Child List* information supplied by the ESR. The `p` modifier makes such new child SAOs permanent. The `t` modifier indicates the creation of transient SAOs. The `u` modifier causes the 'uncreation' of child SAOs which may then subsequently be recreated using the `p` or `t` modifiers. The `m` modifier makes new child SAOs created via the `p` or `t` modifiers 'managed' such that if the ESR removes a child name from the Child List the corresponding child SAO and any tree beneath it will be deleted. The `M` modifier disables the effect of the `m` modifier. This option and its modifiers can be recursively inherited depending on the setting of the `-r` option.

- `-F`
  Enable failover. Reestablishes subscription to the data if a SAF_DROPPED status is received which indicates that no further updates can be supplied. This is enabled by default. This option can be recursively inherited depending on the setting of the `-r` option.

- `-f`
  Disable failover (reverse of `-F` , also inheritable).

- `-s` *suffix*
  The *suffix* string is used as an extension to the name of the SAO for the purposes of the SAO's subscription to an item of data in the ESR. This may be convenient where the desired behaviour would otherwise require the creation of a number of placeholder SAOs simply to achieve a complex hierarchical name (e.g. an SQL query subscription). Double quotes may be used to supply space separated words, double quotes within strings may be escaped with backslash (\ ).

- `-S` *suffix*
  Works the same as the `-s` option, but the *suffix* will be used to replace rather than extend the name of the SAO.

- `-n` *new-name*
  In contrast to the `-S` option, this will cause the name of the SAO to actually be changed to the *new-name* supplied.

- `-m`
  Disables the `meldable` property of this SAO to ensure that every update delivered to it via the container is supplied (i.e. no update merging is performed in the container queue for this SAO). This may be desirable for updates to certain critical subscriptions, and is essential for "query" type subscriptions which result in a stream of updates which represent a result set (e.g. a database query) as merging updates will corrupt the stream.

- `-M`
  Enables the `meldable` property of this SAO. This is enabled by default (reverse of `-m` .

- `-i`
  Treat updates received from a subscription as independent to the current subscription. Ordinarily any update received must result from the current subscription i.e. not result from a previous subscription which has been cancelled. To ensure this, no resubscription is attempted until any existing subscription has completed or been cancelled and that cancellation has been confirmed. Further, any updates which are received that do not match the current subscription are discarded. This flag suppresses these checks. It also allows a new subscription request to be submitted before the current one has completed. This is useful in situations where the ESR will complete all subscriptions but cannot cancel subscriptions once they are made or where a series of subscription requests will be submitted.

- `-I`
  Enforce update checks and cancellations (default, the reverse of `-i`).

- `-q` *suffix*
  A shorthand for the `-mfiS` series of flag settings commonly used for "query" style subscriptions which produce update streams that will be processed by SAOs in the *Stream* category of this library.

- `-v` *[s|v|i]*
  Translate any SAF_STALE updates received according the the modifier specified. The *s* modifier causes SAF_STALE updates to be stored as SAF_STALE (the default). The *v* modifier causes SAF_STALE updates to be stored as SAF_VALID, and the *i* modifier causes SAF_STALE updates to be stored as SAF_INVALID. This option can be recursively inherited depending on the setting of the `-r` option.

Input 3: *Pseudo Parent*

Any SAO supplied on this input will be treated in preference to the parent of this SAO for the purposes of establishing the hieararchical position of this SAO when initiating a subscription to data from the ESR. This allows instances of this SAO to be placed outside the main hierarchy of SAOs related to a particular ESR without the need to form a new connection to the ESR.

Input 4: *Contribution Value*

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.2 Connection

The Connection category contains the following SAO:

- Link

3.2.1 Link

Functional Type:	0
Data Type:	SAF_UNDEFINED
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Connection Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 2:	*Additional Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 3:	*Additional Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Forms a link to an External System. Establishes and maintains a bidirectional connection with an External System, and acts as a root for a hierarchy of data SAOs which can receive and contribute data. In all other respects this SAO then functions like an *Incoming* SAO and once it has established a connection to the External System will make a subscription to the data at the top level of the namespace presented by that system.

Input 0: ***Connection Parameters***

A space separated string of connection parameters may be supplied on this input to enable connection to an External System. Standard command line parsing rules apply to these parameters - spaces and \ characters must be escaped or enclosed in single quotes.

- `-l` *ESLibraryName*
  Shared library which provides an implementation of the External System appropriate to the *ESName* specified with the `-n` option. Specifying a library in this way will ensure that the library is loaded locally in preference to any other implementation of *ESName* which may be available remotely.

- `-c` *name=value*
  Additional configuration information to be passed to the External System, supplied as a name/value pair. In most cases this will only be passed to a locally loaded instance of the

External System specified using the `-l` option.

- `-r`
  Configure the link to retrieve data only (unless the `-s` option is also specified).

- `-s`
  Configure the link to publish data only (unless the `-r` option is also specified).

- `-d` *path*
  Specify the current working directory in cases where the ESLibrary must run in a particular location. Spaces or `'\'` characters must be escaped or defended inside single quotes. **Use with caution!** Only use if there is *no other way* of appropriately loading the ESLibrary. Changing current working directory may conflict with the requirements of other SAOs or ESLibraries hosted by this or other containers within the current process. Be aware that other components may alter the current working directory once the ESLibrary has been initialised.

Note If neither the `-r` or `-s` flags are specified the link will be configured to both retrieve and publish data. If the *ESLibraryName* specified does not support both functions initialisation will fail. Where it is known that the link is required to perform only one function (either sending or retrieving data) appropriate specification of the `-r` or `-s` flags can be used to avoid the creation of superfluous communication sessions associated with the function which is not required.

Input 1: *Configuration Parameters*

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-r` *depth*
  The setting of some options can be passed recursively down a hierarchy of *Incoming* SAOs. Setting the *depth* value will control how far down a hierarchy such options should inherit. The default is `0` . Setting to *-1* results in infinite inheritance. Options which may be recursively inherited are marked as such below.

- `-c` *[t|p|u][Mm]*
  Automatically create children as indicated by *Child List* information supplied by the ESR. The `p` modifier makes such new child SAOs permanent. The `t` modifier indicates the creation of transient SAOs. The `u` modifier causes the 'uncreation' of child SAOs which may then subsequently be recreated using the `p` or `t` modifiers. The `m` modifier makes new child SAOs created via the `p` or `t` modifiers 'managed' such that if the ESR removes a child name from the Child List the corresponding child SAO and any tree beneath it will be deleted. The `M` modifier disables the effect of the `m` modifier. This option and its modifiers can be recursively inherited depending on the setting of the `-r` option.

- `-F`
  Enable failover. Reestablishes subscription to the data if a SAF_DROPPED status is received which indicates that no further updates can be supplied. This is enabled by default. This option can be recursively inherited depending on the setting of the `-r` option.

- `-f`
  Disable failover (reverse of `-F` , also inheritable).

- `-s` *suffix*
  The *suffix* string is used as an extension to the name of the SAO for the purposes of the SAO's subscription to an item of data in the ESR. This may be convenient where the desired behaviour would otherwise require the creation of a number of placeholder SAOs simply to achieve a complex hierarchical name (e.g. an SQL query subscription). Double quotes may be used to supply space separated words, double quotes within strings may be escaped with backslash (\ ).

- `-S` *suffix*
  Works the same as the `-s` option, but the *suffix* will be used to replace rather than extend the name of the SAO.

- `-n` *new-name*
  In contrast to the `-S` option, this will cause the name of the SAO to actually be changed to the *new-name* supplied.

- `-m`
  Disables the `meldable` property of this SAO to ensure that every update delivered to it via the container is supplied (i.e. no update merging is performed in the container queue for this SAO). This may be desirable for updates to certain critical subscriptions, and is essential for "query" type subscriptions which result in a stream of updates which represent a result set (e.g. a database query) as merging updates will corrupt the stream.

- `-M`
  Enables the `meldable` property of this SAO. This is enabled by default (reverse of `-m` .

- `-i`
  Treat updates received from a subscription as independent to the current subscription. Ordinarily any update received must result from the current subscription i.e. not result from a previous subscription which has been cancelled. To ensure this, no resubscription is attempted until any existing subscription has completed or been cancelled and that cancellation has been confirmed. Further, any updates which are received that do not match the current subscription are discarded. This flag suppresses these checks. It also allows a new subscription request to be submitted before the current one has completed. This is useful in situations where the ESR will complete all subscriptions but cannot cancel subscriptions once they are made or where a series of subscription requests will be submitted.

- `-I`
  Enforce update checks and cancellations (default, the reverse of `-i`).

- `-q` *suffix*
  A shorthand for the `-mfiS` series of flag settings commonly used for "query" style subscriptions which produce update streams that will be processed by SAOs in the *Stream* category of this library.

- `-v` *[s|v|i]*
  Translate any SAF_STALE updates received according the the modifier specified. The *s* modifier causes SAF_STALE updates to be stored as SAF_STALE (the default). The *v* modifier causes SAF_STALE updates to be stored as SAF_VALID, and the *i* modifier causes SAF_STALE updates to be stored as SAF_INVALID. This option can be recursively inherited depending on the setting of the `-r` option.

Input 2: *Additional Configuration Parameters*

Additional configuration parameters (refer to the input description of `Configuration Parameters` for details). Values supplied on this input will be appended to any supplied on the `Configuration Parameters` input.

Input 4: *Contribution Value*

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.3 Data

The Data category contains the following SAOs:

- Incoming
- Outgoing
- Contributing
- Snapshot

3.3.1 Incoming

Functional Type:	1
Data Type:	SAF_UNDEFINED
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 2:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 3:	*Pseudo Parent* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Receives data from an External System. Maintains a value based on a single item of data supplied by the External System. Can serve as a parent to other SAOs in nested data structures and can contribute supplied values back to the External System when used in conjunction with an Outgoing SAO in a similar way to the Contributing SAO.

Input 0: *Configuration Parameters*

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-r` *depth*
  The setting of some options can be passed recursively down a hierarchy of *Incoming* SAOs. Setting the *depth* value will control how far down a hierarchy such options should inherit. The default is `0` . Setting to *-1* results in infinite inheritance. Options which may be recursively inherited are marked as such below.

- `-c` *[t|p|u][Mm]*
  Automatically create children as indicated by *Child List* information supplied by the ESR. The `p` modifier makes such new child SAOs permanent. The `t` modifier indicates the creation of transient SAOs. The `u` modifier causes the 'uncreation' of child SAOs which

may then subsequently be recreated using the `p` or `t` modifiers. The `m` modifier makes new child SAOs created via the `p` or `t` modifiers 'managed' such that if the ESR removes a child name from the Child List the corresponding child SAO and any tree beneath it will be deleted. The `M` modifier disables the effect of the `m` modifier. This option and its modifiers can be recursively inherited depending on the setting of the `-r` option.

- `-F`
  Enable failover. Reestablishes subscription to the data if a SAF_DROPPED status is received which indicates that no further updates can be supplied. This is enabled by default. This option can be recursively inherited depending on the setting of the `-r` option.

- `-f`
  Disable failover (reverse of `-F` , also inheritable).

- `-s` *suffix*
  The *suffix* string is used as an extension to the name of the SAO for the purposes of the SAO's subscription to an item of data in the ESR. This may be convenient where the desired behaviour would otherwise require the creation of a number of placeholder SAOs simply to achieve a complex hierarchical name (e.g. an SQL query subscription). Double quotes may be used to supply space separated words, double quotes within strings may be escaped with backslash (\ ).

- `-S` *suffix*
  Works the same as the `-s` option, but the *suffix* will be used to replace rather than extend the name of the SAO.

- `-n` *new-name*
  In contrast to the `-S` option, this will cause the name of the SAO to actually be changed to the *new-name* supplied.

- `-m`
  Disables the `meldable` property of this SAO to ensure that every update delivered to it via the container is supplied (i.e. no update merging is performed in the container queue for this SAO). This may be desirable for updates to certain critical subscriptions, and is essential for "query" type subscriptions which result in a stream of updates which represent a result set (e.g. a database query) as merging updates will corrupt the stream.

- `-M`
  Enables the `meldable` property of this SAO. This is enabled by default (reverse of `-m` .

- `-i`
  Treat updates received from a subscription as independent to the current subscription. Ordinarily any update received must result from the current subscription i.e. not result from a previous subscription which has been cancelled. To ensure this, no resubscription is attempted until any existing subscription has completed or been cancelled and that cancellation has been confirmed. Further, any updates which are received that do not match the current subscription are discarded. This flag suppresses these checks. It also allows a new subscription request to be submitted before the current one has completed. This is useful in situations where the ESR will complete all subscriptions but cannot cancel subscriptions once they are made or where a series of subscription requests will be submitted.

- `-I`
  Enforce update checks and cancellations (default, the reverse of `-i`).

- `-q` *suffix*
  A shorthand for the `-mfiS` series of flag settings commonly used for "query" style subscriptions which produce update streams that will be processed by SAOs in the *Stream* category of this library.

- `-v` *[s|v|i]*
  Translate any SAF_STALE updates received according the the modifier specified. The *s* modifier causes SAF_STALE updates to be stored as SAF_STALE (the default). The *v* modifier causes SAF_STALE updates to be stored as SAF_VALID, and the *i* modifier causes SAF_STALE updates to be stored as SAF_INVALID. This option can be recursively inherited depending on the setting of the `-r` option.

Input 3: ***Pseudo Parent***

Any SAO supplied on this input will be treated in preference to the parent of this SAO for the purposes of establishing the hieararchical position of this SAO when initiating a subscription to data from the ESR. This allows instances of this SAO to be placed outside the main hierarchy of SAOs related to a particular ESR without the need to form a new connection to the ESR.

Input 4: ***Contribution Value***

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.3.2 Outgoing

Functional Type:	2
Data Type:	SAF_UNDEFINED
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Data To Be Contributed* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED

Description:

Arranges to send input value to an External System. This SAO adopts the value of its input as it's own value, registers with the External System to supply that value via the ESS interface, and forms a connection with another SAO above it in the hierarchy which will perform the contribution whenever that value changes.

The indentity of the item of data being contributed to the External System is determined by reference to the names of the SAOs in the path between the *Link* SAO and this SAO's parent.

The type of most of the SAOs in that path is arbitrary although in practice they will generally be *Incoming* SAOs. At least one of them *must* be capable of performing the contribution (*Incoming* , *Contributing* or *Link* SAOs can do this) and be located at a point in the hierarchy appropriate for contribution to the External System in question.

Input 0: ***Data To Be Contributed***

The value, type and status of any SAO connected on this input will be contributed to the External System whenever any one of them changes.

3.3.3 Contributing

Functional Type:	4
Data Type:	SAF_UNDEFINED
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*Configuration Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 2:	*Link Status* *Type:* SAF_VOID *Code:* SAF_RESERVED
Input 3:	*Pseudo Parent* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 4:	*Contribution Value* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Contributes data to an External System. Can serve as a parent to other SAOs in nested data structures and can contribute supplied values back to the External System when used in conjunction with an Outgoing SAO.

Input 0: ***Configuration Parameters***

A space separated string of configuration parameters may be supplied on this input to control the behaviour of this SAO.

- `-f` *"fieldname1 fieldname2 ..."*
  Names to associate with inputs for contribution. If supplied, the registration of inputs with the ESS for contribution will be done by this SAO. This allows direct connection of the SAOs whose values are to be published to the *Contribution Value* inputs (as opposed to using instances of *OutgoingSAO* for each input). If such a configuration is used it is important *not* to attempt to connect the same SAO to multiple *Contribution Value* inputs. Names may be prefixed with a default value which is contributed if the value of the input

to be contributed is unavailable. Such a prefix is supplied between brackets, and the type of the default value is determined from the default value:

- Strings are enclosed in double quotes e.g. `(‘‘na’’)NAME` .
- Doubles include a decimal point e.g. `(0.0)BID` .
- Dates are supplied in ISO8601 format e.g. `(2005-12-31)SETTLEMENT` .
- Times are supplied in ISO8601 format e.g. `(12:23:56)LASTUPDATE` .
- Combined DateTimes may be supplied in ISO8601 format e.g. `(2005-12-31 12:23:56)LASTUPDATE` .
- Integers are supplied as simple digit strings e.g. `(0)LASTUPDATE` .

All default values are sent as SAF_STALE unless the first character of the default value is ˜ , in which case the default is sent with SAF_VALID status e.g. `(˜0.0)BID` . If no default value is supplied but ˜ is specified (e.g. `(˜)BID`) the input is ignored completely if it becomes unavailable, or is supplied as SAF_VALID if it is stale. An empty prefix (e.g. `()BID`) behaves in the normal way as though no prefix is specified.

- `-n` *new-name*
  Change the SAO name to the *new-name* supplied.

Input 3: ***Pseudo Parent***

Any SAO supplied on this input will be treated in preference to the parent of this SAO for the purposes of establishing the hieararchical position of this SAO when initiating a subscription to data from the ESR. This allows instances of this SAO to be placed outside the main hierarchy of SAOs related to a particular ESR without the need to form a new connection to the ESR.

Input 4: ***Contribution Value***

The value of an SAO connected on this input will be sent to the ESS (if available) tagged with the SAOId of that SAO. It is expected that this Id will have been registered with the ESS for the purposes of contribution. The `Outgoing` SAO type is able to manage this automatically and so is the usual choice for an SAO to connect to this input.

3.3.4 Snapshot

Functional Type:	12
Data Type:	SAF_UNDEFINED
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Value to record* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 1:	*Parent* *Type:* SAF_UNDEFINED *Code:* SAF_RESERVED

Description:

Adopts the first usable value available from its input or parent if no input is connected. In addition if it has not adopted a value (ie remains SAF_UNINITIALIZED) and its input or parent gets a terminal status (SAF_COMPLETED, SAF_DROPPED or SAF_CANCELLED) it sets itself SAF_INVALID. The value will only change thereafter if a change is made to the input connection (i.e. a different input source is connected or the existing one disconnected), or the SAO is reset to an invalid status (e.g. SAF_UNINITIALIZED) via the delivery of an update.

If the *Snapshot* SAO is used with the *Incoming* SAO to preserve snapshot data or required to preserve the very first usable value from another SAO then consider disabling the melding flag on the parent or input SAO. Snapshot data will typically have a valid value followed by the terminal status and, depending on how busy the container is, the status and value can get melded.

3.4 Mirrors

The Mirrors category contains the following SAO:

- MultiMirror

3.4.1 MultiMirror

Functional Type:	7
Data Type:	SAF_UNDEFINED
Maximum Inputs:	7
Validated Inputs:	7
Input 0:	*Basic Updating Input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 1:	*Basic Snapshot Input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 2:	*Constructing updating Input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 3:	*Constructing Snapshot input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 4:	*Templating Updating Input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 5:	*Templating Snapshot Input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 6:	*Template Input* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Mirrors the value and status of its input. This SAO has 6 mirroring inputs and will behave differently depending on which of them is connected. Only one of these 6 inputs may be connected at any one time. On forming a connection to one of the mirroring inputs any previous connection to any of the others will be automatically removed. **Note** Only SAOs which are of fixed basic type or are SAF_AMORPHOUS may be mirrored. It is not possible to mirror array types. Further, only the *Basic Updating Input* and *Basic Snapshot Input* will work where the input SAO is in a different container to the mirror.

Input 0: ***Basic Updating Input***

Connecting another SAO to this input will cause it to simply adopt the type, value and status of its input, and update accordingly should the input SAO change.

Input 1: ***Basic Snapshot Input***

Works in exactly the same manner as *Basic Updating Input* with the exception that once it has adopted the type, value and status of the input SAO the input connection will be automatically broken with the effect that the value cannot update should the input SAO change value again. Note that if the input SAO's status is SAF_UNINITIALIZED the connection will be retained until that SAO is initialized with some value or status, at which point the above behaviour will be invoked and the connection will be removed.

Input 2: ***Constructing updating Input***

Works in exactly the same manner as *Basic Updating Input* with the exception that it will automatically connect any of its children which are MultiMirror SAOs to correspondingly named children of the input SAO. If such a child of the input SAO does not exist it will be created as either an SAO of the same type as the input SAO, or according to the type of an SAO connected to the *Template* input. In either case the SAO created will be a transient SAO. Using this input can, therefore, drive the construction of an SAO hierarchy according the the shape of a hierarchy of MultiMirror SAOs.

Input 3: ***Constructing Snapshot input***

Works in exactly the same way as *Constructing Updating Input* with the exception that it will employ the same behaviour as the *Basic Snapshot Input* in that it will disconnect its input once it has adopted a value or status, and will cause its children to do the same.

Input 4: ***Templating Updating Input***

Works in a similar manner to the *Constructing Updating Input* , but rather than cause the construction of suitable input SAOs for its children, will instead create and connect new child MultiMirror SAO instances for itself where necessary to match the children of the input SAO, and will cause its children to do likewise. When this input is used, the input SAO connected will automatically be connected to the *Template* input as well.

Input 5: ***Templating Snapshot Input***

Works in exactly the same way as *Templating Updating Input* with the exception that it will employ the same behaviour as the *Basic Snapshot Input* in that it will disconnect its input once it has adopted a value or status, and will cause its children to do the same.

Input 6: ***Template Input***

The type of this SAO will be used to determine the type of any new input SAO created when either of the *Constructing* inputs is used.

3.5 Streams

The Streams category contains the following SAOs:

- StreamCount
- ColumnCount
- RowCount
- HTMLTable
- StreamTabulator
- ArrayStreamTabulator

3.5.1 StreamCount

Functional Type:	13
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Stream* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Counts the changes to its input until the status of the input becomes SAF_COMPLETED at which point the count is adopted as the SAOs value.

Input 0: *Stream*

An SAO which provides a stream of updates which describe a data set.The sequence of expected updates used as input to this SAO is as follows -

UpdateNo	Type	Value	Description
0	SAF_INT	*n*	Number of columns in the set.
1	SAF_STRING	f1	First column name.
...			
n	SAF_STRING	f*n*	Last column name.
n +1	any	any	First row, first value.
...			
2*n*	any	any	First row, last value.
...			
xn	SAF_INT	0	SAF_COMPLETED end of stream.

3.5.2 ColumnCount

Functional Type:	14
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Stream* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Counts the number of fields whose values are supplied on its input.

Input 0: ***Stream***

An SAO which provides a stream of updates which describe a data set.The sequence of expected updates used as input to this SAO is as follows -

UpdateNo	Type	Value	Description
0	SAF_INT	*n*	Number of columns in the set.
1	SAF_STRING	f1	First column name.
...			
n	SAF_STRING	f*n*	Last column name.
n +1	any	any	First row, first value.
...			
2*n*	any	any	First row, last value.
...			
xn	SAF_INT	0	SAF_COMPLETED end of stream.

3.5.3 RowCount

Functional Type:	15
Data Type:	SAF_INT
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Stream* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Counts the number of rows whose values are supplied on its input.

Input 0: ***Stream***

An SAO which provides a stream of updates which describe a data set.The sequence of expected updates used as input to this SAO is as follows -

UpdateNo	Type	Value	Description
0	SAF_INT	*n*	Number of columns in the set.
1	SAF_STRING	f1	First column name.
...			
n	SAF_STRING	f*n*	Last column name.
n +1	any	any	First row, first value.
...			
2*n*	any	any	First row, last value.
...			
xn	SAF_INT	0	SAF_COMPLETED end of stream.

3.5.4 HTMLTable

Functional Type:	16
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Stream* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Collects the stream of values supplied on its input and presents them as an HTML table.

Input 0: ***Stream***

An SAO which provides a stream of updates which describe a data set.The sequence of expected updates used as input to this SAO is as follows -

UpdateNo	**Type**	**Value**	**Description**
0	SAF_INT	*n*	Number of columns in the set.
1	SAF_STRING	f1	First column name.
...			
n	SAF_STRING	f*n*	Last column name.
n +1	any	any	First row, first value.
...			
2*n*	any	any	First row, last value.
...			
xn	SAF_INT	0	SAF_COMPLETED end of stream.

3.5.5 StreamTabulator

Functional Type:	17
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Stream* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 1:	*Key* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Creates and maintains a hierarchy of SAOs to present data supplied by a query stream using simple data types.

Input 0: ***Stream***

An SAO which provides a stream of updates which describe a data set.The sequence of expected updates used as input to this SAO is as follows -

UpdateNo	Type	Value	Description
0	SAF_INT	*n*	Number of columns in the set.
1	SAF_STRING	f1	First column name.
...			
n	SAF_STRING	f*n*	Last column name.
n +1	any	any	First row, first value.
...			
2*n*	any	any	First row, last value.
...			
xn	SAF_INT	0	SAF_COMPLETED end of stream.

Input 1: *Key*

The number of fields from each row of the stream which will be treated as key fields. This will be used to determine the hierarchical structure of the SAOs which will hold the values of the stream.

3.5.6 ArrayStreamTabulator

Functional Type:	18
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Stream* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 1:	*Key* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Creates and maintains an SAO hierarchy to present data supplied by a query stream using array data types.

Input 0: *Stream*

An SAO which provides a stream of updates which describe a data set.The sequence of expected updates used as input to this SAO is as follows -

UpdateNo	Type	Value	Description
0	SAF_INT	*n*	Number of columns in the set.
1	SAF_STRING	f1	First column name.
...			
n	SAF_STRING	f*n*	Last column name.
n +1	any	any	First row, first value.
...			
2*n*	any	any	First row, last value.
...			
xn	SAF_INT	0	SAF_COMPLETED end of stream.

Input 1: ***Key***

The number of fields from each row of the stream which will be treated as key fields. This will be used to determine the hierarchical structure of the SAOs which will hold the values of the stream.

4 Monitoring Utilities

File:	`monitorsaolib`
Library Version:	1.0
Framework Version:	1.4
No. SAO types:	4

Description:

SAOs for monitoring and measuring running S/AF Applications.

4.1 Latency

The Latency category contains the following SAOs:

- StopWatch
- CheckPoint

4.1.1 StopWatch

Functional Type:	0
Data Type:	SAF_INT64
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	2
Input 0:	*Start* *Type:* SAF_INT64 *Code:* SAF_MANDATORY
Input 1:	*Stop* *Type:* SAF_INT64 *Code:* SAF_OPTIONAL

Description:

Show elapsed time between SAO value or status changes. Reports the elapsed time in milliseconds between a millisecond time value supplied on the *Start* input and the return of that same value on all other connected *Stop* inputs. For example, an instance of the *SystemTime* SAO could be used to establish the time of some event within a container (such as a value change) and this SAO then connected to the *Start* input of a *StopWatch* SAO. The *SystemTime* SAO can then be connected to instances of *EventCheckPoint* SAOs in other containers which are themselves connected back to the *Stop* inputs of the *StopWatch* .

Note A measurement can only be made if the updates on the *Start* input and the later *Stop* responses occur in different container ripples. This will be the case if the relationship between the triggering SAO and the results crosses a container boundary or is in some other way deferred and returned via the container input queue.

4.1.2 CheckPoint

Functional Type:	2
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Check value* *Type:* SAF_INT64 *Code:* SAF_MANDATORY
Input 1:	*Trigger* *Type:* SAF_VOID *Code:* SAF_MANDATORY

Description:

Adopts the value of the *Check value* input only when a simultaneous change occurs on both this input and the *Trigger* input. This SAO may be used in conjunction with instances of the *StopWatch* SAO to associate changes to some SAO (connected on the *Trigger* input) with a time stamp value supplied on the *Check value* input in order to stop the *StopWatch* .

4.2 Statistics

The Statistics category contains the following SAOs:

- UpdStat
- StopWatch Statistics

4.2.1 UpdStat

Functional Type:	1
Data Type:	SAF_STRING
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Data* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 1:	*Reset* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Create a simple statistic about received updates. Reports number of updates, min, average and max of time between updates in milliseconds.

4.2.2 StopWatch Statistics

Functional Type:	3
Data Type:	SAF_INT64
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Timings* *Type:* SAF_INT64 *Code:* SAF_MANDATORY
Input 1:	*Period* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

SAO to collect statistics based on latency timings taken from a StopWatch SAO over a specified time period. Automatically creates and connects to three child SAOs which receive values for the maximum, minimum and mean latency timings seen. The value of this SAO indicates the number of updates received during the time period.

Input 0: *Timings*

Latency timings (in milliseconds) received from a StopWatch SAO.

Input 1: *Period*

Value (in seconds) for a rolling time period over which statistics are to be kept (default is 300 seconds).

5 Hierarchy Utilities

File:	`hierarchy`
Library Version:	1.0
Framework Version:	1.4
No. SAO types:	9

Description:

Hierarchy Monitoring and Operations

5.1 TemplateCopy

The TemplateCopy category contains the following SAOs:

- Monitor
- AtLeast
- MonitorIf
- Matching
- MonitorExcept
- QueryMonitor

5.1.1 Monitor

Functional Type:	0
Data Type:	SAF_INT
Maximum Inputs:	4
Validated Inputs:	4
Input 0:	*SAO being monitored* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 1:	*Local SAO being targeted* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 2:	*Local template SAO* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 3:	*Extension flags* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Keep two trees of SAOs in step.

Input 3: ***Extension flags***

Not used yet.

5.1.2 MonitorIf

Functional Type:	1
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	6
Input 0:	*SAO being monitored* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 1:	*Local SAO being targeted* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 2:	*Local template SAO* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 3:	*Extension flags* *Type:* SAF_INT *Code:* SAF_OPTIONAL
Input 4:	*Path to SAO to be checked* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 5:	*Value to be checked against* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Keep two trees of SAOs in step if an input matches.Will take pairs of inputs (Path and Value) and assume the first is a path in the source tree and the second a value. If the node exists and the value matches (a value of SAF_VOID counts as matching) then the copy will take place.

Input 3: ***Extension flags***

Not used yet.

5.1.3 MonitorExcept

Functional Type:	2
Data Type:	SAF_INT
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	6
Input 0:	*SAO being monitored* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 1:	*Local SAO being targeted* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 2:	*Local template SAO* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 3:	*Extension flags* *Type:* SAF_INT *Code:* SAF_OPTIONAL
Input 4:	*Path to SAO to be checked* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 5:	*Value to be checked against* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Keep two trees of SAOs in step unless an input matches.Will take pairs of inputs (Path and Value) and assume the first is a path in the source tree and the second a value. If the node exists and the value matches (a value of SAF_VOID counts as matching) then the copy will not take place.

Input 3: ***Extension flags***

Not used yet.

5.1.4 AtLeast

Functional Type:	3
Data Type:	SAF_INT
Maximum Inputs:	4
Validated Inputs:	4
Input 0:	*SAO being monitored* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 1:	*Local SAO being targeted* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 2:	*Local template SAO* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 3:	*Extension flags* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Keep two trees of SAOs together ensuring the destination contains at least all trees in the source.

Input 3: ***Extension flags***

Not used yet.

5.1.5 Matching

Functional Type:	4
Data Type:	SAF_INT
Maximum Inputs:	4
Validated Inputs:	4
Input 0:	*SAO being monitored* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 1:	*Local SAO being targeted* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 2:	*Local template SAO* *Type:* SAF_VOID *Code:* SAF_MANDATORY
Input 3:	*Extension flags* *Type:* SAF_INT *Code:* SAF_OPTIONAL

Description:

Keep two trees of SAOs together child for child.Warning. Giving this SAO a full set of inputs will remove all children in the destination that are not present in the source

Input 3: ***Extension flags***

Not used yet.

5.1.6 QueryMonitor

Functional Type:	5
Data Type:	SAF_INT
Maximum Inputs:	4
Validated Inputs:	4
Input 0:	*Local Destination SAO*
	Type: SAF_VOID
	Code: SAF_MANDATORY
Input 1:	*Local Template SAO being copied*
	Type: SAF_VOID
	Code: SAF_OPTIONAL
Input 2:	*Query results stream*
	Type: SAF_VOID
	Code: SAF_OPTIONAL
Input 3:	*0x1=transient copy, 0x2=remove*
	Type: SAF_INT
	Code: SAF_OPTIONAL

Description:

Keep a destination tree in line with the results of a query.

5.2 TreeMonitor

The TreeMonitor category contains the following SAOs:

- LeafMonitor
- LeafExtender
- StringLeafArrayBuilder

5.2.1 LeafMonitor

Functional Type:	6
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	2
Validated Inputs:	2
Input 0:	*Root* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 1:	*Pattern* *Type:* SAF_STRING_ARRAY *Code:* SAF_OPTIONAL

Description:

Monitor the descendants of the SAO specified via the *Root* input which match the pattern supplied on the *Pattern* input. Holds the Ids of those SAOs within the SAF_INT64_ARRAY value of the SAO.

Input 0: *Root*

SAO at the root of a hierarchy which is to be monitored.

Input 1: *Pattern*

Regular expressions detailing the area of hierarchy to be monitored. e.g. `[0-9]/[abc]/foo` will match any SAO called `foo` whose parent is a single character named `a` , `b` or `c` , which is itself parented by an SAO whose name is a single digit and is a child of the SAO specified by the *Root* input. The `/` character is used as a path separator. It can be escaped using the \ character.

5.2.2 LeafExtender

Functional Type:	7
Data Type:	SAF_INT64_ARRAY
Maximum Inputs:	3
Validated Inputs:	3
Input 0:	*Root* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 1:	*Pattern* *Type:* SAF_STRING_ARRAY *Code:* SAF_OPTIONAL
Input 2:	*Template* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Monitor the descendants of the SAO specified via the *Root* input which match the pattern supplied on the *Pattern* input. Extend the SAOs which match using the template specified, holding the Ids of those SAOs within the SAF_INT64_ARRAY value of the SAO.

Input 0: ***Root***

SAO at the root of a hierarchy which is to be monitored.

Input 1: ***Pattern***

Regular expressions detailing the area of hierarchy to be monitored. e.g. `[0-9]/[abc]/foo` will match any SAO called `foo` whose parent is a single character named `a` , `b` or `c` , which is itself parented by an SAO whose name is a single digit and is a child of the SAO specified by the *Root* input. The `/` character is used as a path separator. It can be escaped using the \ character.

Input 2: ***Template***

Template to be used to extend newly detected matching SAOs.

5.2.3 StringLeafArrayBuilder

Functional Type:	8
Data Type:	SAF_STRING_ARRAY
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	4
Input 0:	*Root* *Type:* SAF_VOID *Code:* SAF_OPTIONAL
Input 1:	*Pattern* *Type:* SAF_STRING_ARRAY *Code:* SAF_OPTIONAL
Input 2:	*Parameters to modify behaviour of the SAO* *Type:* SAF_STRING *Code:* SAF_OPTIONAL
Input 3:	*Monitored SAO* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Monitor the descendants of the SAO specified via the *Root* input which match the pattern supplied on the *Pattern* input. Those SAOs which match are automatically connected to the next available *Monitored SAO* input and their values are used to populate the SAF_STRING_ARRAY value of this SAO.

The array value of this SAO consists of the values of the leaf SAOs it finds, in the order they are found. The SAO monitors them for value changes and updates its array as those values change.

When a new leaf is detected its value is added in to the array at an index that corresponds to the order that it is added to its parent. When a leaf is removed the corresponding value is removed from the array. Unless monitoring status the SAO will only put the values from SAF_STRING SAOs with a SAF_VALID status into the value array.

Note Changes to the leaf state of the SAO and the update to the value of the SAO do not occur in the same ripple. The leaf state change is detected and processed in the ripple *after* the leaf is added or removed. Generally, these ripples occur 'back to back', and for dependant SAOs or clients the effect is of a simultaneous change.

Input 0: *Root*

SAO at the root of a hierarchy which is to be monitored.

Input 1: *Pattern*

Regular expressions detailing the area of hierarchy to be monitored. e.g. `[0-9]/[abc]/foo` will match any SAO called `foo` whose parent is a single character named `a` , `b` or `c` , which is itself parented by an SAO whose name is a single digit and is a child of the SAO specified by the *Root* input. The `/` character is used as a path separator. It can be escaped using the \ character.

Input 2: *Parameters to modify behaviour of the SAO*

- `-s`
  The array builder will only include items from SAOs with a *non* SAF_VALID status.
- `-d`
  Debugging information (quite a lot of it) is output to the log file.
- `-f`
  A quote enclosed set of formatting characters used to control the content of array elements. Similar to **printf** in that formatting characters are preceeded with % and other characters are copied to the array element.
  - f
    The full name of the leaf SAO.
  - v
    The value of the leaf SAO as returned by the SAO's getString() method.
  - n
    The name of the leaf SAO.
  - i
    The SAOId of the leaf SAO in the form (<cid>,<oid>).
  - s
    The status of the leaf SAO.

Input 3: *Monitored SAO*

Connections to leaf SAOs within the hierarchy monitored by this SAO. These connections are formed automatically. Manually altering these connections will interfere with the operation of this SAO.

6 Record Publishing

File:	`srzoo`
Library Version:	1.0
Framework Version:	1.4
No. SAO types:	4

Description:

Provision of Publish/Subscribe mechanism

6.1 Uncategorized SAOs

The following SAOs are not assigned to any category within this library:

- Control
- RecordByInputs
- RecordByConfiguration
- Index

6.1.1 Control

Functional Type:	0
Data Type:	SAF_STRING
Maximum Inputs:	1
Validated Inputs:	1
Input 0:	*Control Parameters* *Type:* SAF_STRING *Code:* SAF_OPTIONAL

Description:

Provides control for a set of Record SAOs each of which must be a child. Specifies the way the records are named and built.

Input 0: ***Control Parameters***

A space separated string of control parameters may be supplied to determine the way records are created.

- `-m` *Use SAO name as record name*
  Uses the record SAO's name as the name of the record. This is the default.

- `-i` *Use input as name*
  Uses the value of the first field input to provide the name of the record. This value should be a string.

- `-p` *Use SAO path as record name*
  Uses the record SAO's full path as the name of the record.

- `-d` *Use SAO Id as record name*
  Uses the record SAO's SAOId as the name of the record.

- `-n` *Don't send field names*
  Use this if you do not want the overhead of sending field names

- `-r` *Send raw strings*
  Default off. If set then raw strings will be sent where present.

- `-e` *Send error text*
  Default off. If set then error text will be sent where present.

- `-D` *Send SAOId for each field*
  Default off. If set then each field will have the SAOId included.

- `-b` *Back compatible mode*
  Default off. Set this if you want versions prior to 1.4.1 to be able to receive the records.

6.1.2 RecordByInputs

Functional Type:	1
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	3
Input 0:	*Parent* *Type:* SAF_STRING *Code:* SAF_RESERVED
Input 1:	*Throttle* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 2:	*Field* *Type:* SAF_VOID *Code:* SAF_OPTIONAL

Description:

Creates a record from multiple inputs. Controlled by its parent which must be a control SAO. Field names will be taken from the names of the inputs which must therefore be unique. Each input, except the first if used to specify the name, is used to produce a field in a record. The record will be updated whenever a change in its inputs can be reflected in an update (which is a relatively small message) and refreshed otherwise. The record will be refreshed when any of the following occurs:-

- The parent changes configuration.
- Any input is connected or disconnected, which results in fields being added to or removed from the record.
- Any input changes type or a SAF_STRING or array grows in size.
- Error text or raw strings are being sent and an SAO first gets one of these strings or one grows in length
- The name changes. This is not strictly a refresh because the previous request must be dropped.

The record will be updated, not refreshed, when an input goes unavailable or when it is restored provided it does not change type at the same time. Not all increases in size will result in refreshes because an expansion gap of 25% will be sent in the original record.

The throttle input, if connected, will be used to ensure that changing input data will not cause more than one update in any one period between ticks

6.1.3 RecordByConfiguration

Functional Type:	2
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	4
Input 0:	*InputNames* *Type:* SAF_STRING_ARRAY *Code:* SAF_OPTIONAL
Input 1:	*Throttle* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL
Input 2:	*Parent* *Type:* SAF_STRING *Code:* SAF_RESERVED
Input 3:	*Field* *Type:* SAF_VOID *Code:* SAF_RESERVED

Description:

Creates a record from input array. Controlled by its parent which must be a control SAO. Inputs and fields names will be take from the array in the first input, which contains strings of the form <field name>;<path>. Blank lines and lines starting with a semicolon are ignored. The path is relative to this SAO (usually it will start ../). Each resultant input, except the first if used to specify the name, is used to produce a field in a record. The record will be updated whenever a change in its inputs can be reflected in an update (which is a relatively small message) and refreshed otherwise. The record will be refreshed when any of the following occurs:-

- The parent changes configuration.
- The control array changes, which results in fields being added to or removed from the record.
- Any input changes type or a SAF_STRING or array grows in size.
- Error text or raw strings are being sent and an SAO first gets one of these strings or one grows in length
- The name changes. This is not strictly a refresh because the previous request must be dropped.

The record will be updated, not refreshed, when an input goes unavailable or when it is restored provided it does not change type at the same time. Not all increases in size will result in refreshes because an expansion gap of 25% will be sent in the original record.

The throttle input, if connected, will be used to ensure that changing input data will not cause more than one update in any one period between ticks

6.1.4 Index

Functional Type:	3
Data Type:	SAF_VOID
No Inputs:	

Description:

Creates an index of SAFRecords. The name of this SAO is used as the name of the index. The index is a SAFRecord containing a set of SAF_VOID fields the names of which are the names of all the SAFRecords produced by the process in which it is running. This record will be refreshed whenever a SAFRecord is created or destroyed. By convention using a name of index0, index1 will allow a client to establish the names of all the available records. Be aware that two records in a process will each contain the same set of fields.

7 SQL Utilities

File:	`sqlzoo`
Library Version:	1.3
Framework Version:	1.4
No. SAO types:	8

Description:

SQL database data retrieval

7.1 SQL

The SQL category contains the following SAO:

- SQLWriter

7.1.1 SQLWriter

Functional Type:	7
Data Type:	SAF_STRING
Maximum Inputs:	SAF_UNLIMITED_INPUTS
Validated Inputs:	5
Input 0:	*SQL Formats* *Type:* SAF_STRING_ARRAY *Code:* SAF_MANDATORY
Input 1:	*Primary Key Inputs* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 2:	*Updating Inputs* *Type:* SAF_STRING *Code:* SAF_MANDATORY
Input 3:	*Link Availability* *Type:* SAF_UNDEFINED *Code:* SAF_MANDATORY
Input 4:	*Field Value* *Type:* SAF_UNDEFINED *Code:* SAF_OPTIONAL

Description:

Generates SQL statements based on *Field Value* input values and proforma SQL commands that may be used to maintain those values in the row of a database. The identity of the row is determined using the *Field Value* inputs whose indexes are specified by the *Primary Key Inputs* input in order to construct a suitable WHERE clause to the generated SQL statements. If the value of any of the primary key fields changes, a DELETE command followed by an INSERT command is generated which would cause the row to be replaced. Fields other than the primary key fields which are expected to update are specifield using the *Updating Inputs* input. Changes to these inputs cause an UPDATE command to be generated.

Input 0: *SQL Formats*

SQL INSERT, UPDATE and DELETE command formats in the style of *format* SAO formats (see that SAO for details). These usually contain the name of the table to be updated (and, in the case of the UPDATE command, the names of the columns), and also a suitable WHERE clause which uses the inputs identified via the *Primary Key Inputs* input to match appropriate rows of the table for UPDATE and DELETE commands. The UPDATE command should SET those fields whose values are supplied on inputs defined using the *Updating Inputs* input. Note that use of the %s format code provides implicit convertion of the supplied string to SQL format (i.e. the string is enclosed in single quotes and any single quotes within the field value are escaped). An optional prefix common to each of the above formats can be specified as the fourth element of this array (e.g. the `-q .` option to an *Incoming* SAO which will use the value of this SAO to drive a request to an External System interface).

Example:
The following values are suitable for use with a database whose schema defines a table `Bonds` with five columns; `Id(INT), Isin(STRING), Bid(REAL), Ask(REAL), LastChange`. In this case `Id` is the primary key field.

- "INSERT INTO Bonds VALUES (%I, %s, %f, %f, now())"
- "UPDATE Bonds SET Bid = %f, Ask = %f, LastChange=now() WHERE Id = %I"
- "DELETE FROM Bonds WHERE Id = %I"
- -q

Input 1: *Primary Key Inputs*

Indexes of inputs which represent primary key values for the table to be written.
Example: In the case of the example described in the *SQL Formats* input description, this should be the index of the input to which the `Id` field value is connected.

Input 2: *Updating Inputs*

Indexes of inputs for which an UPDATE can be issued (in preference to a DELETE/INSERT) supplied as a space separated list of input numbers.
Example: In the case of the example described in the *SQL Formats* input description, this should be the indexes of the inputs to which the `Bid` and `Ask` field values are connected.

Input 3: *Link Availability*

The status of this input is used to determine the availability of the connection to the database which will process the SQL statements produced by this SAO. Usually this input is simply connected to the *Link* SAO responsible for the database connection.

Input 4: *Field Value*

Value to write to substitute at the appropriate point in one of the SQL commands whose format is supplied on the *SQL Formats* input.