

Chapter 1. Ruleset Actions..... 6

- Ruleset Actions - A through E..... 6
 - Absolute..... 6
 - Add..... 6
 - AddDateTime..... 7
 - AddDateTimeWithFields..... 7
 - AddDurationToDateTime..... 8
 - And..... 8
 - AssembleContext..... 8
 - BooleanEquals..... 8
 - BooleanNotEquals..... 9
 - BreakLoop..... 9
 - CalculateDuration..... 9
 - Concatenate..... 10
 - ConcatenateMany..... 10
 - Contains..... 10
 - CreateDateTime..... 11
 - CreateDateTimeFromMilliseconds..... 11
 - CreateDateTimeFromString..... 12
 - CreateDuration..... 12
 - CreateList..... 13
 - CreateNumberFromStringWithLocale..... 13
 - CreateStringFromNumberWithLocale..... 14
 - CreateStorageNodeWithID..... 14
 - CreateRange..... 14
 - DateTimeEquals..... 14
 - DateTimeNotEquals..... 15
 - DecodeString..... 15
 - DivideRoundDown..... 15
 - DivideRoundHalfUp..... 16
 - DurationEquals..... 16
 - EncodeString..... 16

EndsWith.....	17
ExecuteContentRouter.....	17
ExecuteMessageBusinessProcess.....	17
ExecuteRouteBusinessProcess.....	18
Exponentiate.....	18
ExtractAfter.....	19
ExtractBefore.....	19
ExtractDigits.....	19
Ruleset Actions - F through L.....	20
ForceTransformationError.....	20
FormatDateTime.....	20
FormatDateTimeWithTimeZone.....	20
GetApplicationRoutes.....	21
GetCurrentDateTime.....	21
GetDaysBetween.....	21
GetDayOfWeek.....	21
GetDayOfMonth.....	22
GetDayOfYear.....	22
GetHourOfDay.....	22
GetMilliseconds.....	22
GetMinuteOfHour.....	22
GetMonthOfYear.....	22
GetNewUuid.....	22
GetNextNumberFromServer.....	23
GetSecondOfMinute.....	23
GetYear.....	23
GreaterThan.....	23
GreaterThanOrEquals.....	24
Length.....	24
LessThan.....	24
LessThanOrEquals.....	24
ListAdd.....	25

ListGet.....	25
ListSize.....	25
LogInfo.....	25
InsertReferenceData.....	26
IsEmpty.....	26
IsFalse.....	26
IsTrue.....	26
IsNotEmpty.....	27
IsNotNull.....	27
IsNotNullOrWhiteSpaces.....	27
IsNull.....	27
Ruleset Actions - M through T.....	27
Matches.....	28
Max.....	28
Min.....	28
Move.....	28
Multiply.....	30
Negate.....	30
Normalize.....	30
Not.....	30
NotContains.....	31
Not End With.....	31
NotStartsWith.....	31
NumberEquals.....	31
NumberOfMatches.....	31
NumberNotEquals.....	32
NumberToString.....	32
Or.....	32
RawMove.....	32
Remainder.....	33
ReplaceAll.....	33
RestoreContext.....	33

RetrieveScriptIDFromRoute.....	34
Split.....	34
Starts With.....	34
StringEquals.....	34
StringEqualsNormalized.....	35
StringEqualsNotNormalized.....	35
StringNotEquals.....	35
Substring.....	35
Subtract.....	36
SubtractDateTime.....	36
SubtractDateTimeWithFields.....	37
SubstringReplace.....	37
SubtractDurationFromDateTime.....	38
ToLowerCase.....	38
ToUpperCase.....	38
Trim.....	38
TrimLeft.....	38
TrimRight.....	39

Chapter 1. Ruleset Actions

These Ruleset Actions are supported in the new transformation engine. There are some that can be used in both versions of the engine.

Ruleset Actions - A through E

Absolute

Returns the absolute value of a number.

Table 1. Input Properties

Property	Type	Description
Factor1	Number	First number.
RoundingDecimalPositions	Number	Rounds the result "Half Up" if <i>roundingDecimalPositions</i> is not null and ≥ 0 . An example would be rounding 0.745 to 0.75

Add

Adds two numbers together and returns a numeric result.

Table 2. Input Properties

Property	Type	Description
Addend1	Boolean	First number.
Addend2	Boolean	Second number.
RoundingDecimalPositions	Boolean	Rounds the result "Half Up" if <i>roundingDecimalPositions</i> is not null and ≥ 0 . An example would be rounding 0.745 to 0.75

AddDateTime

Adds two DateTime types and returns a new DateTime value.

Table 3. Input Properties

Property	Type	Description
Addend1	DateTime	First DateTime.
Addend2	DateTime	Second DateTime.

- If one of the parameters is null, the other will be returned.
- If both parameters are null then null will be returned.


Related concepts

DateTime Variables ([page](#))

AddDateTimeWithFields

Adds specified values to a DateTime and returns a new DateTime value.

Property	Type	Description
DateTime	DateTime	
Year	Number	The year.
MonthOfYear	Number	The month of the year, from 1 to 12.
DayOfMonth	Number	The day of the month, from 1 to 31.
HourOfDay	Number	The hour of the day, from 0 to 23.
MinuteOfHour	Number	The minute of the hour, from 0 to 59.
Seconds	Number	The second of the minute, from 0 to 59.

 **Note:** This Action replaces the older *AddDays* and *AddHours* Actions.

Related concepts

DateTime Variables ([page](#))

AddDurationToDateTime

This task adds to or increments Duration from the DateTime value.

Table 4. Input Properties

Property	Type	Description
DateTime	DateTime	The DateTime variable that the Duration will be added to.
Duration	Duration	The Duration variable to be added to the DateTime variable.

- If DateTime is null, then returns null.
- If Duration is null, then returns DateTime.

Related concepts

Duration Variables ([page](#))

And

Checks whether up to 10 booleans are true and returns a boolean.

- Returns true only if all input booleans are true.
- If all values are null then returns null.
- Returns the boolean result **Or** for all non null booleans.

AssembleContext

Used in conjunction with the *RestoreContext* action. It is also used in Data Analysis Rulesets to create context; this context holds information about the transformation document for outbound EDI.

Related concepts

[RestoreContext \(page 33\)](#)

BooleanEquals

Compares two strings to see if they are equal, and returns a boolean result.

Table 5. Input Properties

Property	Type	Description
Boolean1	Boolean	First Boolean.

Property	Type	Description
Boolean2	Boolean	Second Boolean.

- If both properties are equal, then returns boolean of **true**.
- If both properties are null, then returns boolean of **true**.
- If only one property is null, then returns a boolean of **false**.

BooleanNotEquals

Compares two strings to confirm that they are not equal, and returns a boolean result.

Table 6. Input Properties

Property	Type	Description
Boolean1	Boolean	First Boolean.
Boolean2	Boolean	Second Boolean.

- If both properties are equal, then returns boolean of **false**.
- If both properties are null, then returns boolean of **false**.
- If only one property is null, then returns a boolean of **true**.

BreakLoop

Exits a composite rule based on number of loop (iteration) values.

Table 7. Input Properties

Property	Type	Description
numberOfLoops	Number	<p>If numberOfLoops is not specified or it is 1, it will exit the current composite rule.</p> <p>If numberOfLoops is greater than 1 it will exit as many parent composite rules and the specified number.</p> <p>If the numberOfLoops is negative, the ruleset will exit with an error.</p>

CalculateDuration

This task calculates the time between two DateTime variables.

Table 8. Input Properties

Property	Type	Description
DateTime1	DateTime	The first DateTime (beginning of the duration).
DateTime2	DateTime	The second DateTime (end of the duration).

- If either DateTime value is null, then returns null.
- If DateTime2 is greater than DateTime1, then returned Duration will be positive.
- If DateTime2 is less than DateTime1, then returned Duration will be negative.

Related concepts

Duration Variables ([page](#))

Concatenate

Links string1 with string2 and returns the result as a concatenated string.

Table 9. Input Properties

Property	Type	Description
SourceString1	String	First string.
SourceString2	String	Second string.

If one of the strings is null then returns the other.

If both are null then returns null.

ConcatenateMany

Concatenates all non-null strings using the specified separator and returns a string result.

Table 10. Input Properties

Property	Type	Description
Separator	String	If null, then defaults to empty.
UseRawValue	Boolean	All strings are trimmed before concatenation when this value is false or null.
SourceString (1-10)	String	Returns null when all are empty.

Contains

Returns boolean of true if the text contains the specified string (str) value.

Table 11. Input Properties

Property	Type	Description
Text	String	Text to search.
ContainsText	String	Text to search for.

CreateDateTime

Creates a DateTime.

Table 12. Input Properties

Property	Type	Description
Year	Number	The year.
MonthOfYear	Number	The month of the year, from 1 to 12.
DayOfMonth	Number	The day of the month, from 1 to 31.
HourOfDay	Number	The hour of the day, from 0 to 23.
MinuteOfHour	Number	The minute of the hour, from 0 to 59.
Seconds	Number	The second of the minute, from 0 to 59.

Notes

Ignores any property that is null.

Throws an `IllegalArgumentException` if one of the fields is out of range.

Related concepts

[DateTime Variables](#) (*page*)

CreateDateTimeFromMilliseconds

Creates a DateTime from milliseconds since 1970, 00:00:00 GMT.

Table 13. Input Properties

Property	Type	Description
DateTimeMilliseconds	Number	Example: 1457586000000 will create a date representing 03/10/2016

Property	Type	Description
CenturyIncluded	Boolean	This property is false by default. Example: If property equals true, then 11457586000000 (note the extra 1) will create the same date in example above.

Related concepts

DateTime Variables ([page](#))

CreateDateTimeFromString

Creates a DateTime from a string.

Table 14. Input Properties

Property	Type	Description
DateTimeString	String	
FormatPattern	String	The pattern syntax. (mostly compatible with java.text.SimpleDateFormat)

Example

DateTimeString: 2016/05/31 11:22:10

FormatPattern: yyyy/MM/dd HH:mm:ss

Return: 2016-05-31T11:22:10.000-04:00

Related concepts

DateTime Variables ([page](#))

CreateDuration

This task creates a Duration variable to be used with a DateTime variable.

Table 15. Input Properties

Property	Type	Description
IsNegative	Boolean	Determines if the duration applies to the future (true) or to the past (false).
Years	Number	Number of years.

Property	Type	Description
Months	Number	Number of months.
Days	Number	Number of days.
Hours	Number	Number of hours.
Minutes	Number	Number of minutes.
Seconds	Number	Number of seconds.
Milliseconds	Number	Number of milliseconds.

- If any value is negative, then causes an exception (no validation).
- At least one digit is required, otherwise returns an exception (no validation).
- 0 in any position is valid. *Example: you can have a duration of 0.*
- Any number from 0 to any positive number is valid.
- When *isNegative* is anything but true (or a true boolean variable), then the value defaults to false.

Related concepts

Duration Variables ([page](#))

CreateList

Returns a new empty list of the defined variable type, and is used with other List-variable type Actions.

Related concepts

List Variables ([page](#))

CreateNumberFromStringWithLocale

Converts a Source Number value as the default Java Locale into a String whose format is user-defined by the Locale parameter. For example, passing a number value of "1000.01" to a String defined with the "nl-NL" (Netherlands) locale will result in the value "1.000,01". The format of the Locale needs to be a language tag.

Table 16. Input Properties

Property	Type	Description
Source Value	String	The source number value.
Source Locale	String	This is the locale that represents the format of the source value.

If the number value or Locale is NULL then returns NULL.

CreateStringFromNumberWithLocale

Converts a Source String value whose format is user-defined by the Locale parameter into a Number formatted according to the default Java Locale. For example, passing a string value of "1.000,01" defined with the "nl-NL" (Netherlands) locale through this rule will result in a number with the value "1000.01". The format of the Locale needs to be a language tag.

Table 17. Input Properties

Property	Type	Description
Source Value	Number	The source string value.
Target Locale	String	This is the locale format that you want the target value to be in

If the number value or Locale is NULL then returns NULL.

CreateStorageNodeWithID

Creates a StorageNode object (assignable to StorageNode variable) using a specified UUID String. Typically used in the De-Envloper Ruleset.

CreateRange

Returns a read-only list of numbers; starting and stopping at user-defined values. This list can then be used to create a looping construct in conjunction with a Composite Rule.

Table 18. Input Properties

Property	Type	Description
rangeStart	Number	The starting position of the range.
rangeEnd	Number	The ending position of the range
rangeStep	Number	The interval at which looping occurs within the range. For example, in a range list of 10, a <i>rangeStep</i> of "2" would result in 5 loops.

Related concepts

List Variables ([page](#))

DateTimeEquals

Checks to see if two DateTimes are equal and returns a boolean result.

Table 19. Input Properties

Property	Type	Description
SourceString1	DateTime	First value.
SourceString2	DateTime	Second value.

- If equal, returns true.
- If both null, returns true.
- If one is null, returns false.

Related concepts

DateTime Variables ([page](#))

DateTimeNotEquals

Checks to see if two DateTimes are not equal and returns a boolean result.

Table 20. Input Properties

Property	Type	Description
SourceString1	DateTime	First value.
SourceString2	DateTime	Second value.

- If both are equal, returns false.
- If both are null, returns false.
- If one is null, returns true.

Related concepts

DateTime Variables ([page](#))

DecodeString

Takes a Base64 encoded string and converts to normal string.

DivideRoundDown

Divides one number (dividend) by another (divisor) and returns a numeric result.

Table 21. Input Properties

Property	Type	Description
Dividend	Number	The number to be divided by another number (divisor).
Divisor	Number	The number to divide another number (dividend).

Property	Type	Description
RoundingDecimalPositions	Number	Rounds the result "down" if roundingDecimalPositions is not null and ≥ 0 . An example would be rounding 23.5 to 23.

- If dividend is null, then returns null.
- If divisor is null, then returns dividend.

DivideRoundHalfUp

Divides one number (dividend) by another (divisor) and returns a numeric result.

Table 22. Input Properties

Property	Type	Description
Dividend	Number	The number to be divided by another number (divisor).
Divisor	Number	The number to divide another number (dividend).
RoundingDecimalPositions	Number	Rounds the result "Half Up" if roundingDecimalPositions is not null and ≥ 0 .

- If dividend is null, then returns null.
- If divisor is null, then returns dividend.

DurationEquals

This task checks to see if two Duration variables are equal and returns a boolean result.

Table 23. Input Properties

Property	Type	Description
SourceDuration1	Duration	First Duration value.
SourceDuration2	Duration	Second Duration value.

- If equal, returns true.
- If both null, returns true.
- If one is null, returns false.

EncodeString

Takes a normal string as parameter and converts to a Base64 encoded String.

EndsWith

Returns true if text ends with the suffix; otherwise returns false. If both strings are null then returns true.

Table 24. Input Properties

Property	Type	Description
text	String	
suffix	String	

ExecuteContentRouter

This older Action was used in Data Analysis Rulesets to match values in order to find and execute Routes.

Table 25. Input Properties

Property	Type	Description
Route Values (1-20)	String	Application routing criteria.
Source Context	Storage Node	This is a container for the internal context values to populate the environment (env) and global (glb) levels of the Variables tab in a Ruleset editor. (For example, the sender ID and receiver ID for an inbound EDI document, as they would be included in the environment level in the Variables tab.)
Source Input Data Fragment	Storage Node	A subset of the file being analyzed.

Recommendation

Use the Application Interface (*page*) object instead of this Action when possible.

ExecuteMessageBusinessProcess

Used to call the Business Process defined for the found Route.

Table 26. Input Properties

Property	Type	Description
ScriptID	String	Identifies the Business Process.
Source Route ID	String	Identifies the Route.

Property	Type	Description
Source Context	Storage Node	This is a container for the internal context values to populate the environment (env) and global (glb) levels of the Variables tab in a Ruleset editor. (For example, the sender ID and receiver ID for an inbound EDI document, as they would be included in the environment level in the Variables tab.)
Source Input Data Fragment	Storage Node	A subset of the file being analyzed.

ExecuteRouteBusinessProcess

Used in conjunction with the *GetApplicationRoutes* and *RetrieveScriptFromIDRoute* Actions to call the Business Process defined for the found Route.

Table 27. Input Properties

Property	Type	Description
ScriptID	String	Identifies the Business Process script.
Source Route ID	String	Identifies the Route.
Source Context	Storage Node	This is a container for the internal context values to populate the environment (env) and global (glb) levels of the Variables tab in a Ruleset editor. (For example, the sender ID and receiver ID for an inbound EDI document, as they would be included in the environment level in the Variables tab.)
Source Input Data Fragment	Storage Node	A subset of the file being analyzed.

Exponentiate

Raises the value of an input parameter by the value of another input parameter.

Table 28. Input Properties

Property	Type	Description
Factor1	Number	
Exponent	Number	

- Returns Factor1 at the power of exponent.
- Returns Factor1 if exponent is null.

- Returns null if Factor1 is null.

ExtractAfter

Returns a new string, based on a sub-string of text after the occurrence of another property.

Table 29. Input Properties

Property	Type	Description
Text	String	
ContainedText	String	

Example

ExtractAfter("2003/01/15","20") returns 03/01/15.

ExtractBefore

Returns a new string, based on a sub-string of text before the occurrence of another property.

Table 30. Input Properties

Property	Type	Description
Text	String	
ContainedText	String	

Example

ExtractBefore("2003/01/15","/") returns 2003.

ExtractDigits

Extracts all digits from a String value.

Table 31. Input Properties

Property	Type	Description
input	String	If null, then returns null.

Notes

This does not include Comma (,), Decimal (.), or Negative Sign (-). This Action replaces much of the functionality in the older MoveNumeric.

Ruleset Actions - F through L

ForceTransformationError


Exits a Ruleset with an error.

FormatDateTime

Used to convert a DateTime to a formatted string.

Table 32. Input Properties

Property	Type	Description
DateTime	DateTime	DateTime to be converted
Pattern	String	The pattern syntax. (mostly compatible with java.text.SimpleDateFormat)

 **Note:** This Action replaces the older *ConvertDateFormat* Action.

Related concepts

DateTime Variables ([page](#))

FormatDateTimeWithTimeZone

Used to convert a DateTime to a formatted string using a specified Time Zone.

Table 33. Input Properties

Property	Type	Description
DateTime	DateTime	DateTime to be converted
Pattern	String	The pattern syntax. (mostly compatible with java.text.SimpleDateFormat)
TimeZone	String	Sources for Time Zone and Daylight Saving Time Data come from the public domain tz database; this is also referred to as the Olson database.

- If both properties are equal, then returns boolean of **false**.
- If both properties are null, then returns boolean of **false**.

- If only one property is null, then returns a boolean of **true**.

GetApplicationRoutes

Used in Data Analysis Rulesets to call attributes in Application Route objects. Works in conjunction with the *ExecuteRouteBusinessProcess* and *RetreiveScriptIDFromRoute* actions. These Actions are used for objects created in previous versions of Clarify.

Table 34. Input Properties


Property	Type	Description
Route values (1-20)	String	Application routing criteria.

Related concepts

Application Route ([page](#))

GetCurrentDateTime

Gets current Date and Time. Returns a DateTime value.

 **Note:** This Action replaces the older *GetCurrentDate* and *GetCurrentTime* Actions, because the return value is being specified.

Related concepts

DateTime Variables ([page](#))

GetDaysBetween

Gets the number of days between two DateTime values, and returns a numeric value.

Table 35. Input Properties

Property	Type	Description
Minuend	DateTime	The number from which another is to be subtracted.
Subtrahend	DateTime	The number that is to be subtracted.

 **Note:** This replaces the older *GetDaysDifference* Action.

Related concepts

DateTime Variables ([page](#))

GetDayOfWeek

Gets the day of the week for the DateTime, and returns a numeric value from 1 to 7.

 **Note:** 1 is Monday and 7 is Sunday. Returns null if the specified DateTime is null.

Related concepts

DateTime Variables ([page](#))

GetDayOfMonth

Gets the day of the month from a DateTime, and returns a numeric value from 1 to 31. Returns null if the specified DateTime is null.

Related concepts

DateTime Variables ([page](#))

GetDayOfYear

Gets the day of year from a DateTime, and returns null if the specified date time is null.

GetHourOfDay

Gets the hour of the day from a DateTime and returns a numeric value from 0 to 23. Returns null if the specified DateTime is null.

GetMilliseconds

Gets the number of milliseconds since 1970, 00:00:00 GMT for the DateTime, and returns numeric value. Returns null if the specified DateTime is null.

GetMinuteOfHour

Gets the minute of the hour from a DateTime and returns numeric value from 0 to 59. Returns null if the specified DateTime is null;

GetMonthOfYear

Gets the month of the year from a DateTime and returns a numeric value from 1 to 12. Returns null if the input is null.

Related concepts

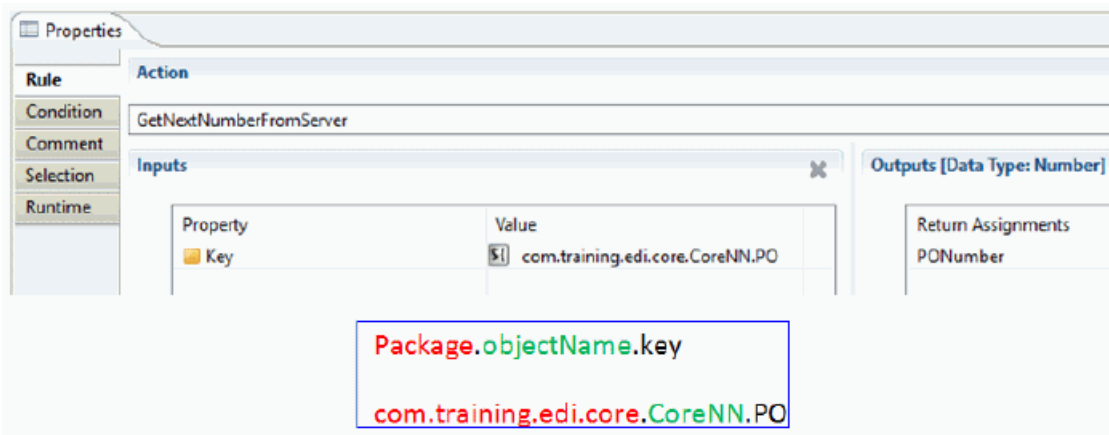
DateTime Variables ([page](#))

GetNewUuid

Creates a new UUID as String. Typically used in the De-Envelope Ruleset.

GetNextNumberFromServer

Retrieves the next value of a Next Number using the specified key. The fully qualified path must be defined as such: `<package name>.<object name>.<next number key name>`.



GetSecondOfMinute

Gets the second of the minute from a DateTime and returns a numeric value from 0 to 59. Returns null if the input is null.

Related concepts

DateTime Variables (page)

GetYear

Gets the year from a DateTime and returns a numeric value.

Related concepts

DateTime Variables (page)

GreaterThan

Checks to see if one number is greater than the other, and returns a Boolean result.

Table 36. Input Properties

Property	Type	Description
numberA	Number	Returns boolean of true if numberA is greater than numberB.
numberB	Number	Returns boolean of false if numberA or numberB is null.

GreaterThanOrEquals

Checks to see if one number is greater than or equal to the other, and returns a Boolean result.

Table 37. Input Properties

Property	Type	Description
numberA	Number	Returns boolean of true if <i>numberA</i> is Greater than or equal to <i>numberB</i> .
numberB	Number	Returns boolean of true if <i>numberA</i> and <i>numberB</i> are null. Returns boolean of false if <i>numberA</i> or <i>numberB</i> is null.

Length

Returns the length of text as a number, and returns null if the text is null.

LessThan

Checks to see if one number is less than another, and returns a Boolean result.

Table 38. Input Properties

Property	Type	Description
numberA	Number	Returns boolean of true if <i>numberA</i> is less than <i>numberB</i> . Returns boolean of false if <i>numberA</i> or <i>numberB</i> is null.
numberB	Number	

LessThanOrEquals

Checks to see if one number is less than or equal to another, and returns a Boolean result.

Table 39. Input Properties

Property	Type	Description
numberA	Number	Returns true if <i>numberA</i> is less than or equals <i>numberB</i> .
numberB	Number	Returns true if <i>numberA</i> and <i>numberB</i> are null. Returns false if <i>numberA</i> or <i>numberB</i> is null.

ListAdd

Adds an element to the specified list.

Table 40. Input Properties

Property	Type	Description
List		The list that is being added to.
Element		The element that is being added to the list.

Related concepts

List Variables ([page](#))

ListGet

Returns an element from a list at a specified position.

Table 41. Input Properties

Property	Type	Description
list	List	The list that being inspected.
index		The position of the element to be listed. The position index starts at 1.

Related concepts

List Variables ([page](#))

ListSize

Returns the number of elements in a specified list.

Table 42. Input Properties

Property	Type	Description
List		The list being measured.

Related concepts

List Variables ([page](#))

LogInfo

Displays additional transformation/application log data in the *Details* section when using the Run Transformation feature in your Ruleset editor.

Related concepts

Run the Transformation directly from the Ruleset (page)

*InsertReferenceData***Table 43. Input Properties**

Property	Type	Description
Log_of_Message_Id	String	
Search_Code	String	
Reference_Data	String	

IsEmpty

Used to condition a Rule based on the non-availability of data. Returns a boolean of true if node is empty.

Table 44. Input Properties

Property	Type	Description
sourceNode		The reference to a piece of data (storage node).
trimValue	Boolean	Trims the data before checking when this property is true of null.

Checks content for data when used with these types: String, DateTime, Element, and Field.

IsFalse

Determines whether an expression is false and returns a boolean.

- If boolean is null, then returns false;
- If boolean is false, then returns true;
- If boolean is true, then returns false.

IsTrue

Determines whether an expression is true and returns a boolean.

- If boolean is null then returns false;
- If boolean is true, then returns true;

- If boolean is false, then returns false.

IsEmpty

Used to condition a Rule based on the availability of data. Returns a boolean of false if node is not empty.

Table 45. Input Properties

Property	Type	Description
sourceNode		The reference to a piece of data (storage node).
trimValue	Boolean	Trims the data before checking when this property is true of null.

Checks content for data when used with these types: String, DateTime, Element, and Field.

IsNull

Returns boolean value of True if sourceNode is not empty.

IsNullOrWhiteSpaces

Contains one property: sourceNode. Returns true for Composite types (Segments, Rows, RecordGroups, Records) if sourceNode is not present in the file.

When used with a simple type (String, DateTime, Element, and Field), this Actions check the contents for the sourceNode. Returns true if the property is null or only contains white spaces. While this may be used with any type of node or variable, it is only useful for simple types.

IsNull

Contains one property: sourceNode. Returns false for Composite types (Segments, Rows, RecordGroups, Records) if sourceNode is present in the file.

When used with a simple type (String, DateTime, Element, and Field), this Actions check the contents for the sourceNode. Returns true if the property is null otherwise it returns false. While this may be used with any type of node or variable, it is only useful for simple types.

Ruleset Actions - M through T

Matches

Returns true if the text matches the specified regular expression.

Table 46. Input Properties

Property	Type	Description
pattern	String	Regular expression.
text	String	

Max

Returns the maximum number of two numbers (Number1 and Number2). For example: *Max*:
 $[Number1=4, Number2=3] \rightarrow [Returns] =4$.

Table 47. Input Properties

Properties	Type	Description
Number1	Number	The first number.
Number2	Number	The second number.

Min

Compares two numbers (Number1 and Number2) and returns the lowest (minimum) number.

Move

Moves data from source to target. (Replaces *MoveNumeric*, *Normalize*, *Trim*, *TrimLeading*, and *TrimTrailing*)

Move and the new Transformation Engine

As part of the evolving engine, improved Move rule functionality reduces the need for manual manipulation of your data during transformation by better inspecting and adhering to source and target Schemas. It is more consistent when reading and writing data; whether moving source data into a Ruleset variable, or directly to target, the Move rule respects the data type and other properties set in the Schema.

Trimming and padding of non-essential characters such as zeros and white spaces

Move automatically executes certain trimming, padding, and spacing behavior when moving data. This can be especially helpful when transforming fixed-length Flat Files. For example:

A field in your Flat File Schema has a starting position of “1” with a Length of “10”. The actual data being moved is “xxxxxxxabc”. (“x” represents 7 empty spaces in front of the data). In the old engine, you would need to use the *Move Trim* rule to handle these spaces. The new Move recognizes these spaces and automatically removes them for you. The target value in this case is “abc”.

Formatting implied decimals in both source and target data

Move also formats implied decimals with less manual intervention. The decimal position is defined in either the source or target Schema for the particular syntax category (Flat File, EDI, or Database, etc). For example:

- For Fixed-length Flat Files, the Move rule uses the Flat File Schema properties *Length* and *Fractional Digits* when reading from and writing to implied decimals.
- For delimited Flat Files, Move uses the Flat File Schema property *Fractional Digits* to read implied decimals.
 - When writing implied decimals, Move uses *Fractional Digits* and *Total Digits*.
 - For target, when the length of data is less than *Total Digits*, the engine automatically pads the output with zeros. However when data length is greater than *Total Digits*, an exception will be thrown.
- For EDI (X12), the Move rule uses the EDI Schema settings *Min Length* and *Max Length* for implied decimals in source and target. Source numeric data values not containing a decimal will have the decimal point added at the implied position. Target numeric values containing decimal positions equal to the implied value have their decimals removed.
- For Database, the Move rule uses the Database Schema property *Fractional Digits* and *Length* when working with *ImpliedDecimal* JDBC type. Values that contain commas (,) or brackets ([]) will not conflict with keywords and delimiters. The engine removes trailing NULL values at the end of the record.

Converting dates into the new DateTime data type

Move now automatically converts a date into the new DateTime data type. It is also capable of reading dates with values not well-formed.

Better error messages provided when moving incompatible data types or violating other Schema properties.

Below are some examples of error messages that may appear when the engine prevents data from being moved.

Cause of error	Error Message
String exceeds <i>maxLength</i> in target	InvalidDataFormatException: Length of value 'ABCDEFGHJIJ' is greater than max length '9' specified in the field 'StringLength10_Max9'
Moving string to number	Incompatible types. Expected Number but was String.

Cause of error	Error Message
Moving decimal to integer	InvalidDataFormatException: The value '555.55' can not be converted to Integer. Field 'IntLength8_Min5'.

Using the new Transformation Engine (page)
[RawMove \(page 32\)](#)

Multiply

Multiplies two values (*Factor1* and *Factor2*) and returns a numeric result.

Table 48. Input Properties

Property	Type	Description
Factor1	Number	First value.
Factor2	Number	Second value.
RoundingDecimalPositions	Number	Rounds the result "Half Up" if <i>roundingDecimalPositions</i> is not null and ≥ 0 .

Negate

Returns the negated value of a number.

Table 49. Input Properties

Property	Type	Description
Factor1	Number	The number to be negated.
RoundingDecimalPositions	Number	Rounds the result "Half Up" if <i>roundingDecimalPositions</i> is not null and ≥ 0 . If null then returns Factor1.

Normalize

Returns a new String replacing all multiple occurrences of white space with a single occurrence. It also trims the original text.

Not

Changes a boolean value from False to True, or True to False.

- If boolean is null, then returns False;

- If boolean is False, then returns True;
- If boolean is True, then returns False.

NotContains

Returns boolean of false if the text contains the specified string (str) value. Returns false if both strings are null.

Table 50. Input Properties

Property	Type	Description
text	String	Specified text.
str	String	Specified string value.

Not End With

Returns false if *text* ends with *suffix*, otherwise returns true. If both strings are null then returns false.

Table 51. Input Properties

Property	Type	Description
text	String	Specified text.
suffix	String	

NotStartsWith

Returns boolean of false if text starts with prefix, otherwise returns true. If both strings are null returns false.

Table 52. Input Properties

Property	Type	Description
text	String	
prefix	String	

NumberEquals

Returns boolean result of true when two numbers are equal (or both are null).

NumberOfMatches

Returns a number.

Table 53. Input Properties

pattern	String	Regular expression. If null, then returns null.
text	String	If null, then returns null.

NumberNotEquals

Returns boolean result of false when two numbers are equal (or both are null).

NumberToString

Takes a number from source and converts it to a sting so that automatic trimming of leading zeros is prevented.

Table 54. Input Properties

Property	Type	Description
Number	Number	The source number to be padded.
PaddingDigits	Number	The total number of digits expected in output. (For example: the source value "1" needs to be padded in the output to be "000001". In this case "6" would be the PaddingDigits.
CountSign	Boolean	Count or not count the minus sign. Default is true .

Or

Checks whether up to 10 booleans are true, and returns a boolean.

- Returns true if at least one input boolean is true.
- If all values are null then returns null.
- Returns the boolean result **Or** for all non null booleans.

RawMove

RawMove transfers data at face value. Unlike Move, it does not facilitate automatic trimming/padding/spacing behavior.

Raw move gets around the new move rule's automatic formatting by moving data literally and regardless of Schema settings. It allows you to retain data intact as it is moved; it can prove helpful when default Move rule formatting is not the desired result.

In this example, the underscore represents empty spaces. RawMove()[_ _ _ _ xyz] -> [_ _ _ _ xyz]

Here the four empty spaces have moved and are intact. Compared to...

```
Move() [_ _ _ _xyx] -> [xyz]
```

Note now that the four empty spaced were trimmed when moving to target.

If you are accustomed to the functionality associated with the Move rule for the older transformation engine (in the version 1 Ruleset), the RawMove can be used instead of Move.

Remainder

Returns the remainder of a division.

Table 55. Input Properties

Property	Type	Description
Addend1	Number	Dividend.
Addend2	Number	Divisor.
RoundingDecimalPositions	Number	Rounds the result "Half Up" if property is not null and >= 0. If null then returns value.

ReplaceAll

Replaces occurrences of *str* in text and returns a new string.

Table 56. Input Properties

Property	Type	Description
text	String	
str	String	
replacement	String	

RestoreContext

Context points can be used to pass RDO and ENV variables between a Business Process and a Ruleset, or to backup and restore the data within the Ruleset. *RestoreContext* updates the variables and the RDO in the Ruleset with the data that is in the Storage Node.

Used in conjunction with the *AssembleContext* action.

Table 57. Input Properties

Property	Type	Description
ContextPointStorageNode	StorageNode	The Storage Node.

Example:

- Set a variable to 10;
- Call *Assemble Context*;
- Change variable to 11;
- Call *RestoreContext*.

Result: the variable returns to 10.

RetrieveScriptIDFromRoute

Used in conjunction with the *ExecuteMessageBusinessProcess* and *ExecuteRouteBusinessProcess* Actions to return the script ID from a Route.

Table 58. Input Properties

Property	Type	Description
RouteID	String	Identifies the Route.

Split

Returns a new list of strings after splitting the specified text into sub-strings based on a regular expression.

Table 59. Input Properties

Property	Type	Description
pattern	String	Regular expression
string	String	String to be split.

Starts With

Returns true if *text* starts with *prefix*, otherwise returns false. Returns true if both strings are null.

StringEquals

Compares two strings for equality and returns a boolean.

Table 60. Input Properties

Property	Type	Description
SourceString1	String	First string.
SourceString2	String	Second string.

StringEqualsNormalized

Returns true if both strings are equal after normalizing them. If both strings are null returns true. If one is null and the other is not returns false.

Table 61. Input Properties

Property	Type	Description
SourceString1	String	
SourceString2	String	

StringEqualsNotNormalized

Returns boolean of true if both strings are equal after normalizing them. If both strings are null then returns true. If one is null then returns false.

Table 62. Input Properties

Property	Type	Description
SourceString1	String	
SourceString2	String	

StringNotEquals

Compares two strings for inequality and returns a boolean.

Table 63. Input Properties

Property	Type	Description
SourceString1	String	First string.
SourceString2	String	Second string.

Substring

Returns a substring of a string. (Used with SubstringReplace). For example, a substring of the word "computer" could be: puter or comp. Another use case would be take a 10-digit phone number and return only certain digits.

Table 64. Input Properties

Property	Type	Description
Text	String	Returns null if text is null. Returns an empty string if the text is empty or only has empty spaces.
Start Position	Number	The position index starts at 1. Defaults to 1 if textStartPosition is null.
Length	Number	Defaults length to the end of the string if it is null.

Subtract

Subtracts the first number (subtrahend) from the second number (minuend) and returns a numeric result.

Table 65. Input Properties

Property	Type	Description
SourceNumber1	Number	subtrahend (number to be subtracted from another).
SourceNumber2	Number	minuend (number from which another is to be subtracted)
RoundingDecimalPositions	Number	Rounds the result "Half Up" if <i>roundingDecimalPositions</i> is not null and ≥ 0 . Defaults both numbers to zero if null.

SubtractDateTime

Subtracts two DateTimes and returns a new DateTime.

Table 66. Input Properties

Property	Type	Description
Minuend	DateTime	The DateTime from which another is to be subtracted.
Subtrahend	DateTime	The DateTime to be subtracted from another.

Returns the minuend if the subtrahend is null;

Returns null if the minuend is null;

Returns null if both parameters are null.

SubtractDateTimeWithFields

Subtracts the specified values from the DateTime and returns a new DateTime.

Table 67. Input Properties

Property	Type	Description
DateTime	DateTime	
Year	Number	The year.
MonthOfYear	Number	The month of the year, from 1 to 12.
DayOfMonth	Number	The day of the month, from 1 to 31.
HourOfDay	Number	The hour of the day, from 0 to 23.
MinuteOfHour	Number	The minute of the hour, from 0 to 59.
SecondOfMinute	Number	The second of the minute, from 0 to 59.

SubstringReplace

Replaces a substring within a string.

Table 68. Input Properties

Property	Type	Description
Text	String	Returns null if text is null.
Text Start Position	Number	The position index starts at 1. Defaults to 1 if textStartPosition is null.
Text Length	Number	Defaults to the end of the string if textLength is null.
Replacement	String	Defaults to empty string if replacement is null
Replacement Start Position	Number	Defaults to zero if start position is null.
Replacement Length	Number	Defaults to the remaining length of replacement if replacementLength is null.

Example

- Text = "abcdef"
- textStartPosition is 3

- `textLength` is 2
- `replacement` is ABCDEF
- `replacementStartPosition` is 1
- and `replacementLength` is 3

Result: "abABCef. The text "cd" has been replaced with "ABC".

SubtractDurationFromDateTime

This task subtracts or decrements Duration from the DateTime value.

Table 69. Input Properties

Property	Type	Description
DateTime	DateTime	The DateTime variable that the Duration will be subtracted from.
Duration	Duration	The Duration variable to be subtracted from the DateTime variable.

- If `DateTime` is null, then returns null.
- If `Duration` is null, then returns `DateTime`.

Related concepts

Duration Variables ([page](#))

ToLowerCase

Returns a new string after converting all characters to lower case.

ToUpperCase

Returns a new string after converting all characters to upper case.

Trim

Returns a new string with leading and trailing spaces removed.

TrimLeft

Returns a new string with leading spaces removed.

TrimRight

Returns a new string with trailing spaces removed.