

Information and/or specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein are covered by one or more U.S. or foreign patents. This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc. Complete Confidentiality, Trademark, Copyright and Patent notifications can be found at: <http://www.tridium.com/galleries/SignUp/Confidentiality.pdf>. © 2011 Tridium, Inc. JACE, Niagara Framework, Niagara AX Framework and the Sedona Framework are trademarks of Tridium, Inc.

NiagaraAX Conversion Links

Starting in AX-3.6, linking components no longer requires that slots have matching data types, nor usage of special components found in the “Conversion” folder in the kitControl palette. Now in most cases, you simply link between a source slot and target slot, regardless of data types, and a conversion link is automatically created to handle conversion.

Each conversion link has a “BIConverter” implementation as a child, which is used to manage and customize the conversion process.

Conversion links can simplify control logic by reducing amounts of needed components. Conversion links can also help to de-clutter wire sheets.

Note: *Conversion links require AX-3.6 or higher on “both ends” when engineering control logic. That is, the host (JACE) running the station must be at AX-3.6 or higher, and Workbench must be at AX-3.6 or higher. Otherwise, conversion links are unavailable.*

The following sections provide more details:

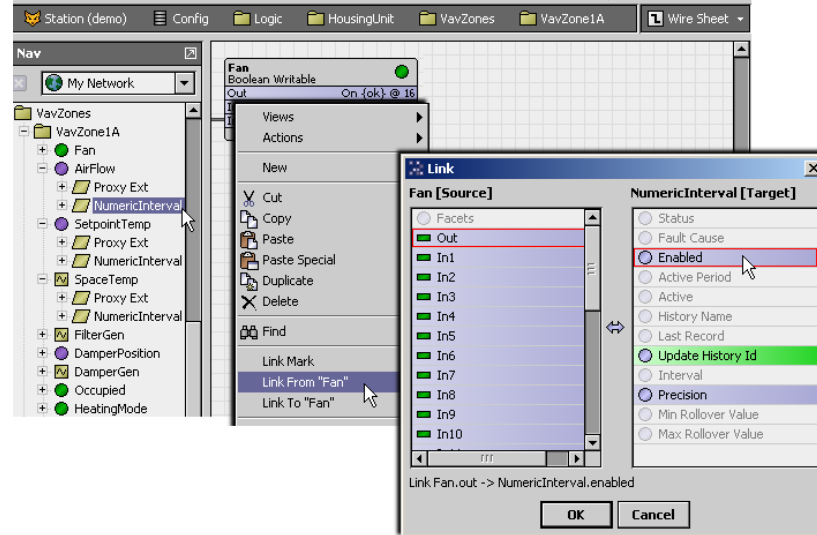
- “Conversion link usage” on page 2
 - “Typical conversion link usage” on page 2
 - “Data transformation via conversion link” on page 3
- “Supported conversion link types” on page 4
 - “Converter components” on page 5
 - “Converter properties” on page 6
- “Conversion link “From” notes” on page 7
 - “Links from string” on page 7
 - “Links from boolean” on page 7
 - “Links from double, float, long, integer” on page 7
 - “Links from statusBoolean” on page 8
 - “Links from statusNumeric” on page 8
 - “Links from statusEnum” on page 9
 - “Links from statusString” on page 9
 - “Links from time” on page 9
 - “Links from absTime” on page 10
 - “Links from relTime” on page 10
- “For more information” on page 10
- “Document change log” on page 10

Conversion link usage

Typical conversion link usage

The most expected usage of a conversion links is to support a link between a “status type” numeric or boolean to a “simple type” numeric or boolean, or vice versa. [Figure 1](#) shows an example.

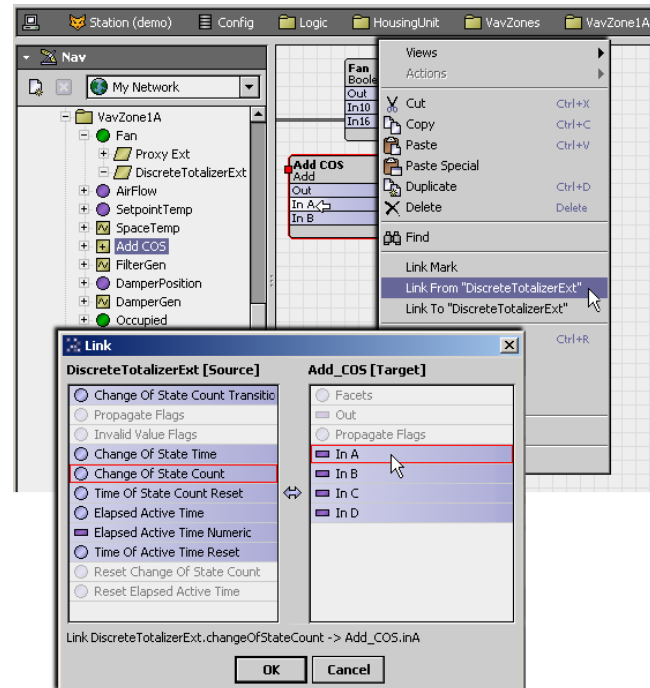
Figure 1 Example link from status type to simple type (statusBoolean to Boolean)



In the example above, the statusBoolean “out” of a BooleanWritable (Fan control) point is needed to enable/disable several history extensions, where the “Enabled” property is a simple boolean type. Before AX-3.6, this required an intervening “StatusBooleanToBoolean” component. However, now you simply link between the two dissimilar slots, as allowed in the **Link** editor.

[Figure 2](#) shows another example, this time linking from a simple data type to a status type.

Figure 2 Example link from simple type to status type (integer to statusNumeric)

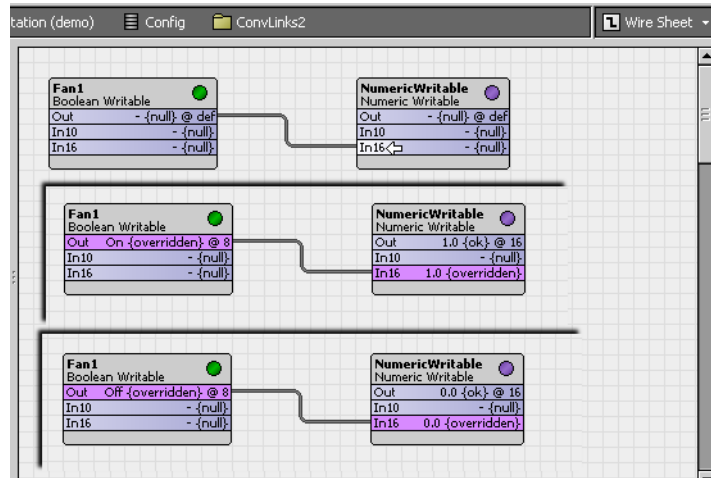


In the example above, the integer “Change Of State Count” slot value of a DiscreteTotalizerExt of a BooleanWritable is needed as an input in a Math component, say an “Add” component. Before AX-3.6, this required an intervening “IntegerToStatusNumeric” component. Again, now you can simply link the two slots, as allowed in the **Link** editor.

Data transformation via conversion link

Many types of conversion links go beyond the “simple-to-status type” or “status type-to-simple” model used in kitControl “Conversion” components. [Figure 3](#) shows one example.

Figure 3 Example link from statusBoolean to statusNumeric

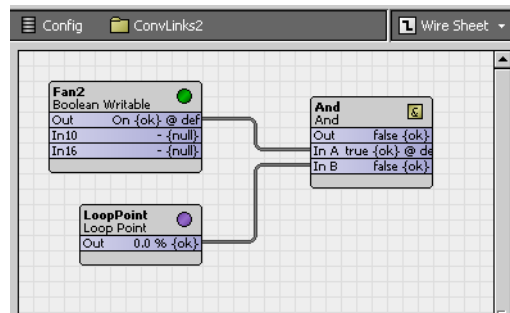


In the example above, the statusBoolean “out” of a BooleanWritable is linked to the statusNumeric “In16” slot of another component. With no other configuration, this transforms the boolean “active” value to a numeric “1”, a boolean “inactive” value to a numeric “0”, and passes through a “null” value.

This sort of “value transform” may be useful in downstream logic. Note before AX-3.6, this type of conversion was often done via a custom Program component.

To reverse this example, you can link the statusNumeric “out” of a component to a statusBoolean slot of another component, for example a kitControl “Logic” component. See [Figure 4](#) below.

Figure 4 Example link from statusNumeric to statusBoolean



In this case, the background “Status Numeric To Status Boolean” converter works like this:

- Numeric value of “0” is boolean “false”.
 - Numeric value *not* “0” is boolean “true” (note this means value > 0, and also a negative value, i.e. < 0).
- Many other link converters are used in various link combinations, including many string-related ones and time-related ones.

For more details, see the following topics:

- [“Supported conversion link types”](#) on page 4 for a matrix of allowed links by data types.
- [“Converter components”](#) on page 5 for details on the converter that is automatically selected.
- [“Converter properties”](#) on page 6 for information on possible converter properties.
- [“Conversion link “From” notes”](#) on page 7 for details on linking from some specific data types.

Supported conversion link types

In AX-3.6 and later, a conversion link automatically results if you link two slots with dissimilar data types.

[Table 1](#) lists those conversion-supported (Y) links, by “From” and “To” slots, by data type.

Table 1 Supported conversion links, by “From” data type and “To” data type

		From															
		string	boolean	double	float	long	integer	frozenEnum	dynamicEnum	statusBoolean	statusNumeric	statusEnum	statusString	ord	time	absTime	relTime
To	string	—	Y*	Y*	Y*	Y*	Y*	Y	Y*	Y	Y	Y	Y	Y	Y*	Y*	Y
	boolean	Y*	—	Y*	Y*	Y*	Y*	Y	Y*	Y	Y*	Y*	Y*	—	—	—	—
	double	Y	Y	—	Y	Y	Y	Y	Y	Y*	Y	Y	Y	—	Y	Y	Y
	float	Y	Y	Y	—	Y	Y	Y	Y	Y*	Y	Y	Y	—	Y	Y	Y
	long	Y	Y	Y	Y	—	Y	Y	Y	Y*	Y	Y	Y	—	Y	Y	Y
	integer	Y	Y	Y	Y	Y	—	Y	Y	Y*	Y	Y	Y	—	Y	Y	Y
	frozenEnum	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	dynamicEnum	—	Y	Y	Y	Y	Y	Y	—	Y	Y	Y	—	—	—	—	—
	statusBoolean	Y*	Y*	Y*	Y*	Y*	Y*	Y	Y*	—	Y	Y*	Y	—	—	—	—
	statusNumeric	Y	Y	Y	Y	Y	Y	Y	Y	Y	—	Y	Y	—	—	Y	Y
	statusEnum	—	Y	Y	Y	Y	Y	Y	Y	Y	Y	—	—	—	—	—	—
	statusString	Y	Y	Y*	Y*	Y*	Y*	Y	Y	Y	Y*	Y	—	Y	—	Y*	Y
	ord	Y	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—
	time	—	—	Y	Y	Y	Y	—	—	—	—	—	—	—	—	Y	—
	absTime	Y	—	Y	Y	Y	Y	—	—	—	Y	—	Y	—	Y	—	—
	relTime	—	—	Y	Y	Y	Y	—	—	—	Y	—	—	—	—	—	—

Each of the “From-To” links made above results in a conversion link, where that link (component) has a specific type of child *Converter* component.

Note: Y* indicates that the Converter component of the link has one or more properties. For more details, see [“Converter components”](#) on page 5 and [“Converter properties”](#) on page 6.

For specific details on many of the “From” data type links, see [“Conversion link “From” notes”](#) on page 7, which contains the following sections:

- [“Links from string”](#) on page 7
- [“Links from boolean”](#) on page 7
- [“Links from double, float, long, integer”](#) on page 7
- [“Links from statusBoolean”](#) on page 8
- [“Links from statusNumeric”](#) on page 8
- [“Links from statusEnum”](#) on page 9
- [“Links from statusString”](#) on page 9
- [“Links from time”](#) on page 9
- [“Links from absTime”](#) on page 10
- [“Links from relTime”](#) on page 10

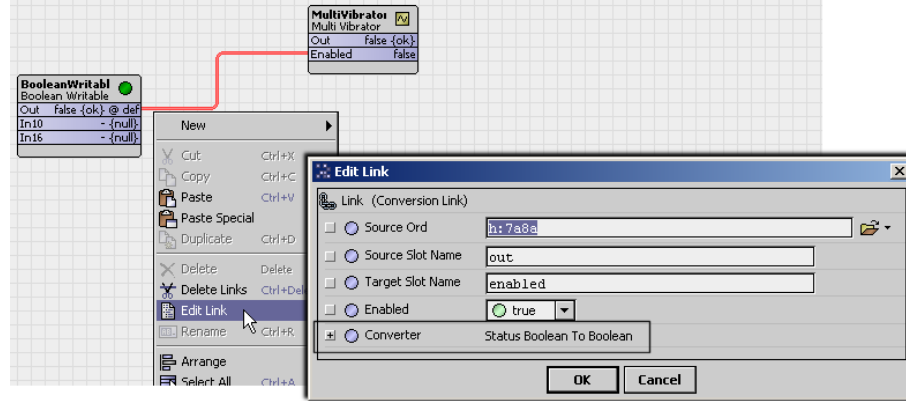
For an overview of conversion links, see [“NiagaraAX Conversion Links”](#) on page 1.

Converter components

Any conversion link has a child “Converter” component. To see it in Workbench, open the **Edit** dialog for the link.

- If the link shows as a *wire* on the wire sheet, *right-click* it and select “**Edit Link**” (Figure 5).

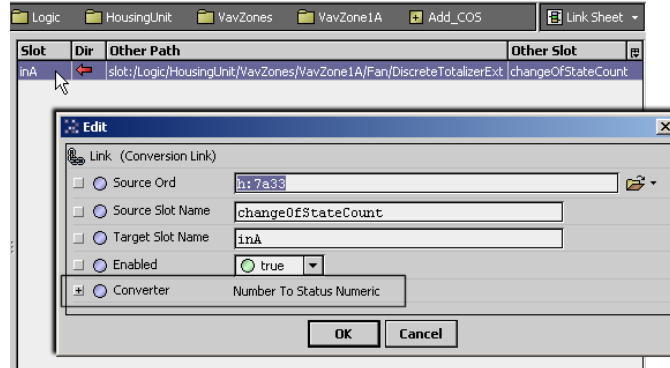
Figure 5 Right-click wire for conversion link and select “Edit Link” to see converter



The conversion link above uses converter type “Status Boolean To Boolean”.

- If the link shows on the wire sheet as a *knob* on a component, go to the component’s *link sheet* view, then double-click the desired link for its **Edit** dialog. See Figure 6.

Figure 6 Go to link sheet of component and double-click conversion link for “Edit” dialog



The link above is a “Number to Status Numeric” converter, allowing the integer “Change Of State Count” value from a DiscreteTotalizeExt to be used in a Math component (from the example shown in Figure 2 on page 2).

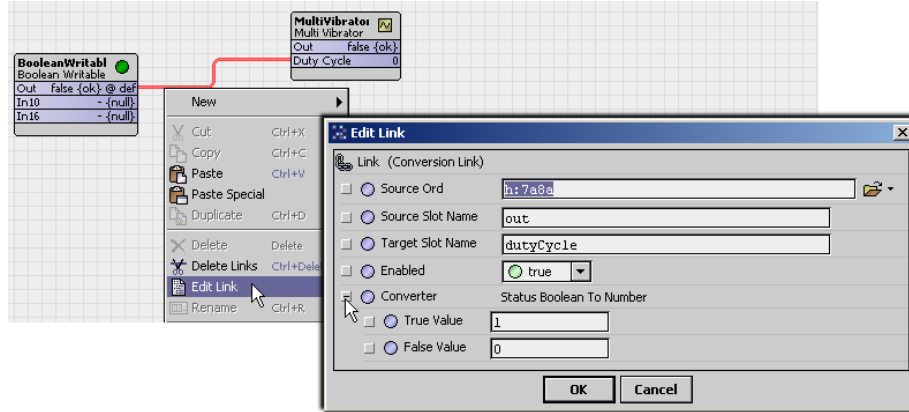
Typically, you seldom need to edit links between components in NiagaraAX. One exception is when linking from the dynamically-created components of a station’s PlatformServices to other components, where you need to edit the “Source Ord” property from “Handle” to “Slot”. For related details, refer to the NiagaraAX Platform Guide section “PlatformServices binding and link caveats”.

Another possible exception may be with some conversion links, where a child Converter component has one or more properties (such links are indicated with a “Y*” in the Table 1 matrix of supported links). See “Converter properties” on page 6.

Converter properties

Most conversion links have no properties under the link's Converter component. This is true for all conversion links between "like" data types, that is, status type-to-simple and vice versa. However, conversion links that transform values like string to boolean or numerical, or the reverse (to string) may have one or more Converter properties. [Figure 7](#) shows one example.

Figure 7 Example Converter with properties (Status Boolean To Number)



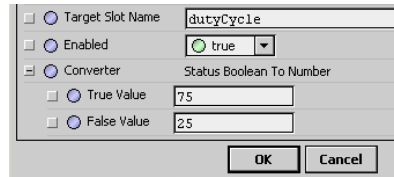
The example above reflects default values for a link made from a statusBoolean slot (in this case, "out" of a BooleanWritable) to an Integer slot (in this case, "Duty Cycle" of a MultiVibrator). The converter type automatically used is "Status Boolean To Number".

Expanding the Converter component in the **Edit Link** dialog reveals two properties:

- True Value — default value of 1
- False Value — default value of 0

The duty cycle range of a MultiVibrator is from 0 to 100, so defaults here are not typically appropriate.

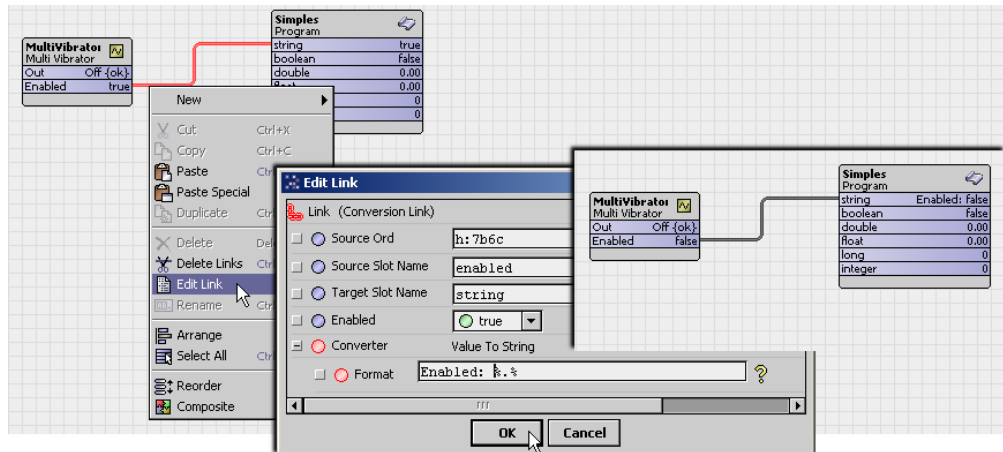
Figure 8 Example Converter properties edited to non-default values



In this example, converter properties were edited to true at 75, and false at 25, as shown in [Figure 8](#) above.

In cases where the link's slot target (To) data type is string or statusString, the Converter property, if present, is typically "Format". If a default "% .%" value, this means it is BFormat (Baja Format).

Figure 9 Example Format property of "Boolean To String" Converter edited from default "% .%"



[Figure 9](#) above shows the static text "Enabled:" prepended on the default % .% Format property value of a "Boolean to String" link Converter.

Conversion link "From" notes

These notes describe some conversion link behaviors, grouped by source (From) data types:

- [“Links from string”](#) on page 7
- [“Links from boolean”](#) on page 7
- [“Links from double, float, long, integer”](#) on page 7
- [“Links from statusBoolean”](#) on page 8
- [“Links from statusNumeric”](#) on page 8
- [“Links from statusEnum”](#) on page 9
- [“Links from statusString”](#) on page 9
- [“Links from time”](#) on page 9
- [“Links from absTime”](#) on page 10
- [“Links from relTime”](#) on page 10

Also see [“Supported conversion link types”](#) on page 4, and an overview [“NiagaraAX Conversion Links”](#) on page 1.

Links from string

- A string-to-boolean link or string-to-statusBoolean link has a “False Value” converter property, with a default value of `false`. Any other source string value, instead of (or in addition to) this “False Value” string (case insensitive) results in the target boolean or statusBoolean value to be *true*. Note that a *blank* string source value is also a boolean or statusBoolean true (unless the “False Value” is blank).
- A string-to-double or string-to-float link requires a source string using decimal numerals only, expressed as a decimal number or integer, either positive or negative. Any other source sting characters (or blank string) results in the linked double or float slot to have an “nan” (not a number) value.
- A string-to-long or string-to-integer link requires a source string expressed as an integer only, either positive or negative. Any additional characters in the source string results in the linked long or integer slot to have a 0 value, or the last non-zero value.
- A string-to-statusNumeric link requires a source string using decimal numerals only, expressed as a decimal number or integer, either positive or negative. Any other source sting characters (or blank string) results in the linked statusNumeric slot to have a “fault” condition, with no value change.
- A string-to-statusString link results in that string value in the statusString.
- A string-to-ord link provides no Niagara ord format/error checking—use sparingly.
- A string-to-absTime link requires an ISO-formatted string—otherwise, the linked absTime value remains null. The ISO format is “`yyyy-mm-ddThh:mm:ss.mmm[+/-]hh:mm`”, for example, “2011-01-31T13:23:53.772-05:00”, which an absTime slot could show as “31-Jan-2011 01:23 PM EST”.

Links to other [“Conversion link "From" notes”](#) on page 7.

Links from boolean

- A boolean-to-string results in either a “false” string value if boolean false, or “true” string value if boolean true. The link has a “Format” converter property. See [Figure 9](#) on page 6 for an example.
- A boolean-to-simple number data type link (to-double, to-float, to-long, to-integer) results in a 0 value for a boolean false, or 1 if a boolean true.
- A boolean-to-statusBoolean link has a “False Value” converter property, with a default value of 0. The default 0 keeps the statusBoolean value in synch with the source boolean value. If “False Value” is set to 1, the linked statusBoolean value is opposite (NOT) the source.
- A boolean-to-statusNumeric link or boolean-to-statusEnum link results in a 0 value for a boolean false, or 1 if a boolean true.
- A boolean-to-statusString has a string value `false` if false, or string value `true` if boolean true.

Links to other [“Conversion link "From" notes”](#) on page 7.

Links from double, float, long, integer

The following summarize links from one of the simple number data types (double, float, long, or integer, described below as “number”), to another data type:

- A number-to-string or number-to-statusString link results in the string value of that number, for example “78.30” The link also has a “Format” converter property based on a *text string*, with a blank default value. The blank Format outputs all existing digits with no formatting. In the Format property, you can enter a Format value using pound signs (#) for digits, 0 numeral(s) for leading/trailing zero(s), and placeholder separators, such as a comma (,) for grouping and/or period (.) as decimal separator. Some Format value examples:
 - ###,###.### — where a source number 123456.789 is string formatted as 123,456.789

- ###, ## — where a source number 123456.789 is string formatted as 123456.79
- 00000.000 — where a source number 123.78 is string formatted as 000123.780
- A number-to-boolean link or number-to-statusBoolean link has a “False Value” converter property, with a default value of 0, such that any number value other than 0 results in a boolean true. If needed, “False Value” can be edited to specify a different value to associate with false.
- A number-to- “different simple number type” link results in that number, unless outside the range of the target data type. For example, a double-to-integer link with a source value of 2147484000 (exceeding max integer value of 2147483647) will result in the integer value of “max” (2147483647).
- A number-to-statusNumeric link results in that number.
- A number-to-statusEnum link results in that number, unless outside the ordinal (integer) range of a statusEnum data type. For example, a double-to-statusEnum link with a source value of 2147484000 (exceeding max integer value of 2147483647) will result in an Enum (ordinal) value of 2147483647.
- A number-to-Time link results in that number of milliseconds added to the base time of 12:00am midnight (00:00). For example, a float value of 900000 is seen as Time 12:15am (00:15).
- A number-to-absTime link adds that number of milliseconds to the Java “epoch” date/timestamp of December 31 1969 7pm EST. For example, a long value of 1296509138929 results in an absTime value of January 31 2011 4:25pm EST.
- A number-to-relTime link adds that number of milliseconds to 0ms. For example, an integer value of 4800000 results in an relTime value of 1hour 20mins, or -108000 results in -1min 48sec.

Links to other “[Conversion link "From" notes](#)” on page 7.

Links from statusBoolean

- A statusBoolean-to-string results in either a “false” string value if boolean false, or “true” string value if boolean true. A statusBoolean “null” value does not change the target string value.
- A statusBoolean-to-simple number data type link (to-double, to-float, to-long, to-integer), by default, results in a 0 value for a boolean false, or 1 if a boolean true. A link’s Converter has two editable child properties: “True Value” (default = 1) and “False Value” (default = 0), to specify non-default values.
Typically, a statusBoolean “null” does not change a target long or integer link 0 value; however, a “null” changes a linked double or float value to “nan” (not a number).
- A statusBoolean to statusNumeric or statusEnum link results in a 0 value for a boolean false, or 1 if a boolean true. A statusBoolean “null” changes the statusNumeric or statusEnum target to “null”.
- A statusBoolean-to-statusString, by default, has a string value false if false, or string value true if boolean true. A statusBoolean “null” changes the statusString target to “null”.

Links to other “[Conversion link "From" notes](#)” on page 7.

Links from statusNumeric

- A statusNumeric-to-string link results in the string value of that number, for example “78.30”. A statusNumeric “null” value does not change the target string value.
- A statusNumeric-to-boolean link has a “False Value” converter property, with a default value of 0.0, such that *any number value other than 0* results in a boolean true. If needed, “False Value” can be edited to specify a different value to associate with false.
Typically, a statusNumeric “null” value does not change the target boolean value.
- A statusNumeric to “different simple number type” link results in that number, unless outside the range of the target data type. For example, a statusNumeric-to-integer link with a source value of 2147484000 (exceeding max integer range) will result in the integer value of “max” (2147483647). A statusNumeric “null” value does not change/affect a linked long or integer value; however, a “null” changes a linked double or float value to “nan” (not a number).
- A statusNumeric-to-statusBoolean link has a “False Value” converter property, with a default value of 0.0, such that *any number value other than 0* results in a boolean true. If needed, “False Value” can be edited to specify a different value to associate with false.
A statusNumeric “null” value changes the statusBoolean target to “null”.
- A statusNumeric-to-statusEnum link results in that number, unless outside the ordinal (integer) range of a statusEnum data type, whereby it is “clamped” at that max or min value.
A statusNumeric “null” value changes the statusEnum target to the “null value” (often 0).
- A statusNumeric-to-statusString link results in the string value of that number, for example “78.30”. The link also has a “Format” converter property, based on a *text string*, with a blank default value. The blank Format outputs all existing digits with no formatting.
In the Format property, you can enter a Format value using pound signs (#) for digits, 0 numeral(s) for leading/trailing zero(s), and placeholder separators, such as a comma (,) for grouping and/or period (.) as decimal separator. Some Format value examples:

- ###,###.### — where a source number 123456.789 is string formatted as 123,456.789
 - ###,## — where a source number 123456.789 is string formatted as 123456.79
 - 00000.000 — where a source number 123.78 is string formatted as 000123.780
- A statusNumeric “null” value changes the linked statusEnum target to “null”.
- A statusNumeric-to-absTime link adds that number of milliseconds to the Java “epoch” date/timestamp of December 31 1969 for another date-timestamp. For example, a statusNumeric value of 1296509138929 results in an absTime value of January 31 2011 4:25pm EST.
- A statusNumeric “null” value changes the absTime value to “null”.
- A statusNumeric-to-relTime link adds that number of milliseconds to 0ms. For example, a statusNumeric value of 4800000 results in an relTime value of 01h 20m.
- A statusNumeric “null” value does not change the current linked relTime value.

Links to other [“Conversion link "From" notes”](#) on page 7.

Links from statusEnum

- A statusEnum-to-string link results in the string value of the Enum’s tag (descriptor), for example “Off” or “Occupied”. A statusEnum “null” value does not change the target string value.
 - A statusEnum-to-boolean link has a “False Value” converter property, with a default value of 0, such that *any ordinal value other than 0* results in a boolean *true*. If needed, “False Value” can be edited to specify a different ordinal (integer) Enum value to associate with false.
- A statusEnum “null” value does not change the target boolean value.
- A statusEnum to “different simple number type” link results in the (integer) ordinal value of that Enum. A statusEnum “null” value does change/affect a linked long or integer value; however, a “null” changes a linked double or float value to “nan” (not a number).
 - A statusEnum-to-statusBoolean link has a “False Value” converter property, with a default 0 value, such that *any ordinal value other than 0* results in a boolean *true*. If needed, “False Value” can be edited to specify a different ordinal (integer) Enum value to associate with false.
- A statusEnum value “null” value changes the statusBoolean target to “null”.
- A statusEnum to statusNumeric link results in the (integer) ordinal value of that Enum. A statusEnum “null” value changes the statusEnum target to “null”.
 - A statusEnum-to-statusString link results in the string value of the Enum’s tag (descriptor), for example “Off” or “Occupied”. A statusEnum “null” value changes the statusString target to “null”.

Links to other [“Conversion link "From" notes”](#) on page 7.

Links from statusString

- A statusString-to-string link results in that string value in the statusString.
- A statusString-to-boolean link has a “False Value” converter property, with a default value of `false`. Any other source string value, instead of (or in addition to) this “False Value” string (case insensitive) results in the target boolean value to be *true*. Note that a *blank* statusString source value is also a boolean *true* (unless the “False Value” has been set to blank).
- A statusString-to-double or statusString-to-float link requires a source string using decimal numerals only, expressed as a decimal number or integer, either positive or negative. Any other source string characters (or blank string) results in the linked double or float to have a “nan” (not a number) value.
- A statusString-to-long or statusString-to-integer link requires a source string expressed as an integer only, either positive or negative. Any additional characters in the source string results in the linked long or integer slot to have a 0 value, or the last non-zero value.
- A statusString-to-statusNumeric link requires a source string using decimal numerals only, expressed as a decimal number or integer, either positive or negative. Any other source string characters (or blank string) results in the linked statusNumeric to have a “fault” condition, and no value change.
- A statusString-to-ord link provides no Niagara ord format/error checking—use sparingly.
- A statusString-to-absTime link requires an ISO-formatted string—otherwise, the linked absTime value remains null. The ISO format is “`yyyy-mm-ddThh:mm:ss.mmm[+/-]hh:mm`”, for example, “2011-01-31T13:23:53.772-05:00”, which an absTime slot could show as “31-Jan-2011 01:23 PM EST”.

Links to other [“Conversion link "From" notes”](#) on page 7.

Links from time

- A time-to-string link results in a string value that (by default) reflects hr:min:sec.ms, for example a time of 3:31 PM could result in a string value of 15:31:23.647. The link has “Format” converter property, with a default value of `HH:mm:ssZ`. You can edit this if needed—for example reducing Format to `HH:mm` for string output like 15:31, or to `HH:mm a` for string output 3:31 PM.
- A time-to-“simple number type” link (double, float, long, integer) results in the number of millisec-

onds in the current time since the base time of 12:00 AM midnight (00:00). For example, a time-to-integer link from 11:30 AM would have a value of 41400000.

- A time-to-absTime link results in the current date and time in the absTime value.

Links to other [“Conversion link “From” notes”](#) on page 7.

Links from absTime

- An absTime-to-string or absTime-to-statusString link results in a string value for a date-timestamp that (by default) reflects `yyyy-mo-dayThr:min:sec.ms-tzone offset hr`, for example a value of `2011-02-01T13:47:13.358-05:00`. The link has “Format” converter property, with a default value of `YYYY-MM-DDTHH:mm:ssZ`. You can edit this if needed—for example to change Format to `MM-DD-YYYY HH:mm a` to get a string output like `02-01-2011 13:51 PM`.
- An absTime-to-“simple number type” link (double, float, long, integer) results in the number of milliseconds for that date-timestamp since the Java “epoch” timestamp of December 31 1969.
- An absTime-to-statusNumeric link results in the number of milliseconds for that date-timestamp since the Java “epoch” timestamp of December 31 1969.
- An absTime-to-time link results in the time portion of the date-timestamp to be the time value.

Links to other [“Conversion link “From” notes”](#) on page 7.

Links from relTime

- A relTime-to-string link or relTime-to-statusString link results in a string describing the relative time. For example, a relTime of `25h 20m 02s` can result in a string value of `1day 1hour 20mins 2sec`.
- A relTime-to-“simple number type” link (double, float, long, integer) reflects the number of milliseconds in the relative time. For example, a relTime value of `01h 20m` results in a value of `4800000`.
- A relTime-to-statusNumeric link reflects the number of milliseconds in the relative time. For example, a relTime value of `01h 20m` results in a value of `4800000`.

Links to other [“Conversion link “From” notes”](#) on page 7.

For more information

Additional developer-level information about Baja converters can be found in `bajadoc`, with a listing of all the various converters at `local:|module://converters/doc/javax/baja/converters`

Document change log

Updates (changes/additions) to this *NiagaraAX Conversion Links* Engineering Notes document are listed below.

- Publication: April 4, 2011
Initial document.