Advanced Reports and Table Concepts v9.0



Verify that you have the most current version of this document from **www.hvacpartners.com**, the **Carrier Partner Community** website, or your local Carrier office.

Important changes are listed in **Document revision history** at the end of this document.

Carrier© 2024. All rights reserved.

The content of this guide is furnished for informational use only and is subject to change without notice. Carrier assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.



Contents

What are Advanced Reports and Tables?	1
Selecting the Primary Column	2
Adding Data Columns	6
About Aspects	6 7
Useful GQL Node Steps Useful Attributes	9 9
Data Column - Expression	11
Data Column - Trend Data	12
Data Column + / - Date Range	15
Data Column - Function	16
Variables	20
Where	22
Options	23
Output	26
Preview	27
Icons	28
Graphics	30
Miscellaneous	33
Document revision history	34

What are Advanced Reports and Tables?

Advanced Reports and Tables is designed to create a data table to use as tables and charts on a graphic or as a tabular report that you can save as a PDF or CSV file.

To create the data table

- **1** Define the table name and ID.
- 2 Decide if you want the report to appear in the **Reports** menu in the i-Vu® interface. If so, select an optional category for grouping it.
- **3** Select the rows to define the primary column.
- 4 Define the data you want for each column. Each of the following settings are optional:
 - Variables
 - Where the report can be run
 - Data Options
 - Customized Output

Selecting the Primary Column

The first step in defining a report is to select what determines the primary column. This results in the total number of rows and is the value or location that most of the other columns are evaluated relative to.

Listed below are the descriptions of each primary column type that you can select.

Primary column type	Purpose
Locations	To select specific locations you want in the report in an exact order. The list of locations and the content is the same, regardless of where you run it from.
	NOTE This type is simple and straight forward.
Control Programs	To select equipment at or below the desired location to include in the report based on their control program name.
	EXAMPLE Running a report about "all chillers" or "all RTUs"
	You can select the control program names from the list or use pattern matching. This is useful if the control programs all start with the same prefix name, or if you version the control programs and need all current and future types, regardless of file name changes.
	EXAMPLE If the chiller control program names are "chiller_rev1" and "chiller_rev2", use the pattern "chiller_*" to select both the existing names and also the future "chiller_rev3" name.
Reference Names	Location types whose reference names match a specific name or pattern. If chillers always start with "chiller," then entering a reference name pattern "chiller*" selects all of the chillers at or below the location where the report is run. You can enter as many names or patterns as necessary.

Primary column type	Purpose		
Tag Names	Geographic location types based on semantic tags. Tag Names are new in the i-Vu® v8.0 application and Semantic tags are more powerful than References Names and sometimes even Control Programs.		
	EXAMPLE To create a "meter" report - tag all of the meter equipment with the "meter" tag (Tag ID="meter") and the corresponding equipment is selected, regardless of its Reference Name or Control Program name.		
	Use multiple tags to more precisely select locations. EXAMPLES		
	• To create a "water meter" report, select both the "water" and "meter" tags. Then, add the Tag Names list.		
	Primary column: Tag Names		
	 Add multiple sets of tags to create more advanced selections. For example, to select the water meter and gas meter but not other meters, add two entries. The first is "meter & water" followed by "meter & gas". As the report runs, the tags at the location are examined to see if they match all of the tags in any 		

AND "gas" must be present.

Tag Names

Meter & Gas

Meter &

Primary column: Tag Names

Water

Any tag name entry can select a location as long as all of the tags in that entry are present.

row. In the example, both "meter" AND "water" must be present OR "meter"

v

The advantage of using Semantic tag names in a report is that as new locations are added to the system, they will be selected in reports. The reference name and control program name do not matter.

Primary column type	Purpose		
Date Range	Creates rows based on dates going backward from the current date or dates going forward from a specific date. This row type is typically used to select trend data.		
	The date range format can be used to customize the display of the starting date of each date range. If the interval is "months", instead of a default label such as "12/01/2020," use "MMM" for the 3 month prefix ("Dec") or "MMMM" (4 or more) to obtain the full month name ("December"). "YYYY MMM" produces "2020 Dec". You can also quote literal text using "single quotes"; 'Month of MMMM' produces "Month of December."		
Existing Report	Two main use cases:		
	• To use another report as a starting point.		
	EXAMPLE Create a report that selects all chillers in the system, have nothing else defined, and "Show in Reports Menu" is not selected. However, this "base report" would feed other chiller reports the list of selected chillers. As new chillers are added to the system, only the first "selecting" report would need to be modified to have all of the other reports updated. The better solution would be to use semantic tags.		
	• The existing report is provided by an add-on. An example is the "Aspect Tables" add-on. It provides several "base reports" that select microblocks based on an aspect (an internal API concept that defines a set of properties based on the type).		
	EXAMPLE The "aspect_table_lockable" report that selects all microblocks that are "lockable" can be used as a starting point to create a "locked point info" table or report. This permits singular tables or reports to be supplied by an add-on that are otherwise unable to be created using the Report Manager interface.		
Color Map	Used with the Color Map control on a graphic. The locations for the "table data" are the locations associated in the Color Map. The expectation is that the columns are rendered as a "Color" to product custom color maps using any desired color scheme or meaning.		

Primary column type	Purpose
Hide primary column in report	An option for all of the primary column types is to hide the default display of the primary column in the report. A useful application is to show the display path instead of only the display name.

Location	Usage
Big Chicken	0.0
City Hall	0.0

By hiding the default primary **Location** column and creating a new column using the ".displayPath", we can show the path to the location based on the system setting **Levels displayed in paths**.

Туре	Path	Path	
Display Name	Location	Usage	
ID	alt_location	usage	
Add Copy	Delete		
Display name:	Locatio	on	
D:	alt_loc	ation	
Render data as:	Value	✓ Value a	
Column format: Align: Left V Width: 0 Dig			
Column data is f	rom: Path	~	
Path:	.display	/Path	
Show value as te	ext? 🗹 👔		
Preview			
Hide Refresh	Sho	w all columns 📄 Show Column IE	
Locati	on	Usage	
	1.1.1	0.0	
Marietta / Big C	nicken	0.0	

Advanced Reports and Table Concepts v9.0

Adding Data Columns

After you have selected the primary column, the data for each row is created by adding columns. All columns are evaluated left to right. You cannot reference a column in an expression that is to the right.

For all columns, you decide how to render the value:

- Value displays the value
- Hidden does not show the column

NOTE Use when the data is intermediate and used in a calculation by a later column or by a filter.

- Color interprets the value as a color. Does not display the column but makes it available to the color map control
- Icon uses the value as the name of an image

Data Column - Path

The **Path** column data type selects a value relative to the location for that row based on a relative data path. A path can be most any GQL path string, like those used on the i-Vu® **Graphics** page.

In some cases, the value for a path will be a boolean or enumeration that has both a text value and a numeric value. For example a boolean can be **True/False** or **1/0**. Enumeration might be **Low/Medium/High** or **1/2/3**. When you select **show value as text**, the text is shown in the operator's language. Otherwise, the numeric value is used.

About paths

The path traversal process in data tables exclusively uses internal APIs, meaning it uses **Location**, **Node**, and **Aspect** and not **CoreNode** directly. This has some constraints when you can only use items in the navigation tree. You cannot access anything in the Schedule Groups tree.

The path also leverages an internal API that refers to **aspect** properties. The most useful ones are documented below.

- A path is a sequence of steps through locations or nodes and then an optional property. Present value, that is
 "value" property or the getValue() function, is the default property if none is specified. If the property string has
 dashes, like "some-thing", it will fall back and try a camel case variant of "someThing" as well.
- Most GQL paths in the navigation tree should work fine. ALT+click on a field and most display the **Global Modify** dialog that displays the path to the value with a **Copy** button.

About Aspects

When on a location, you can reference an aspect as the next step in the path. If valid, properties of the aspect, functions that start with **get**, can be referenced using the "dot-property" form. If the name of the aspect or the reference ends with "Aspect", it can be referenced with or without "Aspect". For example, "FooAspect" can be referenced using "foo" or "fooAspect". You can reference the **Foo** aspect with "foo" or "fooAspect". The aspect name is case insensitive, therefore, "FOOaspect" works as well.

EXAMPLE Equipment has the **AttachedEquipment** aspect with a **getColor** method, which returns an

EquipmentColor enumeration. The path to this from an equipment location would be:

 $\verb+attachedequipment.color.\/ (The full path might be$

/trees/geographic/area1/some_equipment/attachedequipment.color.) This might have the text value MAXIMUM_HEATING. Use the **Convert To Color** function to convert this to an RGB color to use with a color map.

Complex Aspect Properties

The AttachedEquipment aspect has a getDefiniton method that returns an EquipmentDefinition. This then has a getName method.

The path to this would be chained together using "dot-property" notation as: attachedequipment.definition.name

EXAMPLE attachedequipment.device.outOfService

Key Aspects and Their Properties

IMPORTANT The property names below are case sensitive. You must type "outOfService" and not "outofservice".

AttachedEquipment

Valid for equipment

- **device** returns the Device Aspect (see below)
- **color** returns an EquipmentColor
- prime returns prime value
- **definition** returns an EquipmentDefinition
 - **definition.name** returns the equipment type

NOTE EquipmentColor is an enumeration. It will have text values like **OCCUPIED** or **MODERATE_HEATING**. You can use attachedequipment.color.value and get the number value from 0 to 15, which is the same as "~color".

Device

Valid for device

- "outOfService" true/false if the device is out of service
- "modelName" the device type/model name
- "macAddressString" the MAC address of the controller

Driver

Valid for device

- **name** the name of the driver file
- version the version number of the driver

NOTE The **Driver** aspect cannot be accessed currently due to a precedence order where the driver child location will be found first. This has been corrected in v8.0 Cumulative 1, and it can now be referenced using **DriverAspect**.

ModuleStatus

Valid for device

"reportText" - returns the entire modstat text. This can be slow. The "regular expression" function can be used to extract parts of the text.

PhysicalPoint

Valid for a physical point microblock

- **Input** true/false if it is an input point
- enable true/false if it is enabled
- **IONumber** the input/output number (see getIONumber() method)
- expander the expander number

Useful GQL Node Steps

The following table lists some of the useful pseudo-node steps.

Path	Description	
~parent	Navigate up one node.	
~equipment	Navigate to the equipment (up) that the current node is associated with.	
~device	Navigate to the device (up or up and over). This can be at or under the equipment, or at or under the driver.	
~color	Navigate to the equipment first and then to the color node.	
~prime	Navigate to the equipment first and then to the prime value node.	
~network	Navigate to the device and then the parent.	
~site	Navigate to the network and then up to the site.	

Useful Attributes

The following table lists some of the useful attributes that are used after the "dot". For example, ".display-name".

Value	Where	Description
.alarm_count	*	The number of active alarms for the location (those waiting to return to normal)
.display-name	*	The display name of the location
.display-path	*	The display path of the location. Depending on system setting, it shows "N" parent display names as well.
.gqlpath	*	Shows the full GQL path to the location
.instance-number	Equipment	The control program number
.instance-number	Device	The controller's Device ID
.mac-address	Device	The controller's MAC address
.media-type	Network	The network media type (ex: "bacnet/ip", "arcnet")
.model-name	Device	The controller's model name
.network-number	Network	The BACnet network number
.notations	Equipment	The notations
.reference-name	*	The reference name of the location
.serial-number	Device	The controller's serial number
.sort	*	The numeric tree sort value

Advanced Reports and Table Concepts v9.0

Carrier Proprietary and Confidential

Order of traversal (advanced):

1. Locations - the Location "getChild(step)" method is tried first. This works down to microblocks / points.

2. Nodes - If using a Node or at a Location where the step doesn't work, the Location is converted to a Node. The Node "evalToNode(step)" method will then be evaluated.

3. Aspects - If "evalToNode" fails, then this step is used to determine if an aspect exists with the given name, which is case insensitive. The word "Aspect" does not have to be given. For example, if the aspect class is "FooAspect", just "foo" will work.

~node:

You can use "~node" as a step to force evaluation to move from Location to Node. This is needed in rare cases of "name step conflicts" and you want to force the node traversal user and not the Location child or aspect to be found.

~location:

You can use "~location" as a step for evaluation to move from a Node to a Location. This is needed when some paths like "~device" only work for Nodes but you want to get back to a Location in order to access an aspect.

NOTE "~location" support requires the i-Vu® v8.0 April 2021 cumulative update or later.

Data Column - Expression

The expression column type allows for robust math and logical expressions that involves numbers and text or strings. Basic operators for addition ('+'), subtraction ('-'), multiplication ('*') and division ('/') are supported. Grouping is done with parentheses. Multiplication and division take precedence over addition and subtraction.

The expression (1+2*3-4/2) is evaluated in this order:

- 1 1+ 2*3-4/2
- 2 1+ 6- 4/2
- **3 1+6**-2
- 4 7-2
- 5 5

It is recommended to use parenthesis to avoid confusion: 1 + (2*3) - (4/2).

See the product help for a complete list of math and text/string functions.

Substitution

You can substitute other values into an expression by enclosing it with '\$'. It can be a variable, the value from a previous column (to the left), or a path. It is important to understand that the substitution of the value is done before parsing the expression.

If you are substituting something that is a string, it is recommended to put it in quotes. Use double quotes if the substituted string might contain single quotes and vice versa.

For the following examples, assume we have a variable with id "limit" (maybe a number set to 72), a column "setpoint" and a column "zone_temp".

- "Difference is " + (\$zone_temp\$ \$setpoint\$).
- \$setpoint\$ > \$limit\$? "Setpoint above limit" : "Setpoint ok" .
- abs(\$zone_temp\$ \$setpoint\$) < 3 ? "Green" : "Red".
- "\$.parent.display-path\$" + " / " + "\$.display-name\$" (see note below)
- 1+1 # The rest of this is a comment.

NOTE You must include the quotes around 'display-path' because that will include a '/' which will look like division if not quoted.

Data Column - Trend Data

The "trend data" column type supports basic operations on a single series of trend data for a given time range. There is no capability of using another data series at the same time, such as showing the max zone temp while the zone is occupied using a "zone temp" and "occupancy" trend together.

The "trend path" is typically a path to a trend source relative to the location of the current row or the location of the report. It can be an absolute path that will be the same trend source regardless of where the report ran from.

There are several trend operations, most of which are straight forward.

The "Aggregate Consumption" is designed to work with meter-style trend sources, such as something that is counting or accumulating an always increasing value. An electric meter or water meter is a typical example. The aggregate consumption value is the sum of the value's increments between each sample.

If the samples were 10, 12, 15, 15, 19, 20, the value would be (12-10) + (15-12) + (15-15) + (19-15) + (20-10) or 2+3+0+4+1 = 10. Or simply, the last sample minus the first sample (20-10 = 10).

However if the value ever decreases, it assumes that the meter was reset to zero and will adjust the accumulated value accordingly. If the samples were 10, 12, 15, 3, 5 the value would be (12-10) + (15-12) + (3-0 - RESET!) + (5-3) or 2 + 3 + 3 + 2 = 10.

The **"% Time in Range"** function allows you to compute the percentage of time that the trend samples are equal to or within a range of values. For each sample, if the value of the sample matches the range, it will then include the time between the PREVIOUS sample and the current sample as being in range. The first sample at the start (or the data series or after a gap) is only used as the starting time for the following sample. Its value is ignored.

The argument can include multiple values or ranges separated by comma. For example, "0:10,20:30" would match values from 0 to 10 and from 20 to 30. You can invert the match by starting the arguments with "!". So "!0:10,20:30" would match all values NOT from 0 to 10 and also NOT from 20 to 30.

You can use "Infinity" and "-Infinity". To select all non-negative values would be "0:Infinity". The word "Infinity" must be capitalized. "-Infinity:0" would match all non positive values.

Advanced Reports and Table Concepts v9.0

Interval Sample

The "Interval Sample" is used to decide whether to include or exclude the first or last sample within a time range. If the interval is daily, this will determine whether the "00:00" or the "24:00" sample is included in the current day. This option only applies to some functions.

Database trends only

It is very important to set this option correctly. When collecting data for a trend function, the most recent data may not be in the database yet. If the requested time range is after the latest sample in the database, then the system will attempt to read the most recent data from the controller. It will take time to read data from a controller. If it is acceptable for the report to not be using the most up to date data, then this option should be selected to only query from the database, which will result in a much faster report.

Ignore specific samples

This option lets you specify a list of values or ranges to ignore. This is intended to be used when there are one or more invalid values in the trend data. Often network points have a default value such as "-1" or "-999" that indicate it is not reading a valid value. To prevent a value from affecting the trend calculation, it can be ignored. See **% Time in Range** above for example of ranges.

Time Range

From primary column: The date range should be set using the primary column type of "Date Range" which will create a hidden column with the ID "daterange". Select "show all columns" in the preview section to see this in detail.

From column: The data range will come from a previous column with the specified ID. The report manager cannot create a custom "date range" column. This is intended to be used as an extension mechanism by another add-on that would supply a "base report" of ranges that would be referenced using primary column type of "existing table". A use case would be for billing analysis where the date ranges are based on a custom billing cycle that do not correspond to the first day of the month.

Value: A specific start and end date are specified.

Past: A count of past days/weeks/months/quarters/years from the current date.

Advanced Reports and Table Concepts v9.0

Custom trend date formatting

Some of the trend functions will show the time of the sample. By default, the system date/time format is used. This may look like "71.2 @ 12/25/20 6:30 AM".

You can override the trend format in a single report by setting a variable name 'trend_time_format' with the date/time format desired. In the example below, the value would show as "71.2 on Dec 25". Since date/time formats use letters to represent the various substitution parts, include literal text in single quotes.

Туре	Columns	Variables	Where	Options	Output
ID	Туре	Value			
trend_time_format	String	'on' MMM dd			
Add Delete					
ID: trend_time_f	ormat	Type: Strin	ig 🗸 Value: 🤇	'on' MMM dd	

For more information on formatting options, search the Internet for "customizing date time formats with java" or see https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html for details and examples.

Data Column + / - Date Range

The **+/- Date Range** column type is used to create a date range based on a date range in a previous column. This is used when the primary column is a date range.

EXAMPLE Show the past twelve months in the primary column, then adjust the date range by one year to have a second column that selects trend values for months of the prior year. This allows for a table or chart of monthly energy consumption of this year vs prior year.

It can adjust by any number of days, weeks, months, quarters, or years forward or backwards. Typically, dates are adjusted back in time but a negative value adjusts forward in time.

By default, it modifies the primary date range column, but you can specify a column ID with another date range to adjust.

Adjusting by **Month** doesn't always work for the end of the month when the date range is a single day or a week. Because February 2020 doesn't have the 30th or 31st, the adjusted date range in the example below would be a **no value** cell.

Date Range	Adj Date
Mar 25	Feb 25
Mar 26	Feb 26
Mar 27	Feb 27
Mar 28	Feb 28
Mar 29	Feb 29
Mar 30	?
Mar 31	?

When adjusting by month, quarter, or year, if the input range is a whole month (1st to last day or quarter), then the new range adjusts to the last day of the respective month. So "Feb 1-28" adjusted by 1 month will be "Jan 1-31" and not "Jan 1-28".

NOTE This requires the i-Vu® v8.0 August 2021 cumulative update or later.

Data Column - Function

The **Function** column type is used to perform a function on a prior column on the same row. Each function is designed for a specific purpose that helps with creating useful tables and reports. Each function is configured with an input column (ID) of a prior column on the same row. Columns are evaluated left to right, and so it cannot reference a column to the right as it will not have a value yet.

If a function takes an argument, you can reference a variable by using "\$var_name\$" in the argument string to replace with the variable's value.

The following list describes the specific functions:

- **Valid Column -** returns value "true" if the input column has a valid value. Otherwise it will be "false". Examples of invalid values that this was designed to detect are invalid paths for the current location, no communication, or an error in the expression.
- **Default Value** returns the value of the input column if it is valid, otherwise returns the "argument" given. This is similar to the Valid Column function, but instead of true or false, it provides a useful value. This is often used to ensure a value is available to use in an expression, in case field communications fail. Otherwise the cell will be invalid, and that causes the expression to be invalid as well.



• You can use a variable name substitution ("\$varname\$") in the argument for the default value.

Format - provides the ability for advanced formatting of values, typically used with numbers. Search the
Internet for "string format with java 8" or see
https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html. Assume we have a cell with ID "input"
that has the numeric value 12345.67 (a floating point number). The format conversion "f" would be used.

Format	Result	Notes
%.1f	12345.7	Format to 1 digit with rounding.
%.5f	12345.67000	Format to 5 digits.
%+.2f	+12345.67	Format always with a leading "+" or "-" .
\\$%,.2f	\$12,345.67	Format to 2 digits. Show a dollar sign (need to escape with "\") and show "comma" every 3 digits.
\$format_var\$?	Use the format string set in a variable.

• Format Duration - The "Format Duration" specifically works with the "% Time In Range" trend calculation. It provides customized formatting of the duration. There are two ways to format it. The first is where the duration is split into days, hours (0-23), minutes (0-59) and seconds (0-59). The lower case format letters are used for this time style. The second style is total count of days, hours, minutes or seconds. Note that for the second style, the counts are rounded down. Thus, 59 minutes will be 0 hours.

EXAMPLE The format "%hh%:%mm% or %MM% total minutes" would show "02:45 or 165 total minutes"

Advanced Reports and Table Concepts v9.0

• Format Trend Sample Time - works specifically with the first, last, min, or max value "with time" trend operations. It provides the ability to format the date/time of the trend sample using the formats already described.

If you want to show both the sample value and a custom formatting time in one cell, use 3 columns. First, get the sample with a trend operation in a hidden column. Use **Format Trend Sample Time** in a second hidden column. Use an "expression" column to append the two values together.

• **Convert Values to Text** - designed to convert a number range into something more meaningful. The arguments are a list of "number=output value" pairs where the numbers must increase with each pair. The first number that is equal to or below the input value will be matched. The first entry can just be a value (no equal sign) and that will be used as a default minimum value.

Input	Arguments	Output	Notes
85	0=F,60=D,70=C,80=B,90=A,100=A+	В	> 80= B
-15	0=F,60=D,70=C,80=B,90=A,100=A+	-1	Not greater than any pair, -1 is error output.
-15	F,60=D,70=C,80=B,90=A,100=A+	F	First item is default for low values.
64	Cold,68=Perfect,75=Warm	Cold	Uses default.
68	Cold,68=Perfect,75=Warm	Perfect	Matches 68=Perfect.
74.99	Cold,68=Perfect,75=Warm	Perfect	Below 75 - numbers are not rounded.
71	too_cold.png,68=perfect.png,75=war m.png	perfect.p ng	Selects an image file to render as "Icon."

- **Convert Integer to Text** This function is like the "Convert Values to Text" but it is more precise and requires an exact integer match. Every pair separated by a comma must be "number=text". If there is no match, the value is empty. This is typically used to convert enumerations into labels. If there is a precise set of possible values, this function should be used; otherwise, the "Convert Values to Text" function should be used.
- **Convert Text to Integer** The "Convert Integer to Text" function. It converts a text string into a number. It takes a list of "text=number" pairs, is case insensitive, and can use "*" to match any number (zero or more) of letters.

Sometimes a value will be "text," and it needs to be transformed for use in a numeric calculation or color gradient.

• **Convert to Color** - takes an "ALC Color" number in the range of 0-15 and converts it to the equivalent RGB value. For example, the number "7" (Moderate Cooling) will result in "#FFFF00". The function is designed to turn a color number into an RGB value to render data as "Color" for use with color map controls. It is important to remember to select render data as "Color" for this column.

• **Color Gradient** - used to convert a number into a color for use with the color map control. It takes 4 arguments: min, max, color1, color2. The arguments "min" and "max" are numbers that will map from 0 to 100 percent. Anything below "min" will be 0 percent and anything above will be 100%. That percentage is then used to select a color value using an HSV gradient from "color1" (0 percent) to "color 2" (100 percent).

This function is useful for creating interesting custom color maps that can indicate energy usage or temperature ranges. It is important to remember to select render data as "Color" for this column.

- Date Range Start used to create custom formatting for the starting date of a date range when the primary column type is date range. The input column ID is normally "daterange". See "Custom trend date formatting" above for options.
- **Date Range End** used to create custom formatting for the ending date of a date range when the primary column type is date range. The input column ID is normally "daterange". See "Custom trend date formatting" above for options.
- Ordinal Value used in rare cases where the "value" or a column is a Java enumeration. In some cases using Aspects, the value may be an enumeration. A simple example of this would be to use the path ".type" which will return the value "Equipment". That is an internal API Java enumeration of the location type. If the "Ordinal Value" function were applied to that column, it would get a value of "6".

Use the "attachedEquipment.color" expression and then the "Ordinal Value" function to convert the label into a number.

EXAMPLE For example, "MODERATE_COOLING" converts to 7. Follow that with a "Convert To Color" function and you get the RGB value "#FFFF00" for MODERATE_COOLING.

- Location Tags shows all of the semantic tags and their values if they are a value tag. This can be used to
 create a report of the tags applied to a location. It was created so one could create their own "semantic tags
 report" specific to their needs.
- **Regular Expression** used to extract part of the text from a string value. The text matching the regular expression will be the output value of the function. You can also use a capture group, which simply surrounds part of the expression with opening and closing parentheses. The best example of this function is to work with the module status text. The "path" column type with a path of "moduleStatus.reportText" will leverage the internal API "ModuleStatus" as expected to return the module status report text as a single value. This will create a value in a column that is all of the module status text. Typically this column will be hidden.

Search for "regular expression patterns with java 8" on the internet for information about regular expressions or see https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html .

The following table shows examples of extracting key information.

Expression	Output	Notes
Downloaded by: .*	Downloaded by: i-Vu Pro Server 12/25/20 06:30 John Doe	Grabs the entire line that starts with "Downloaded by:". The ".*" will stop at the end of the line.
Downloaded by: .*Server (.*)	12/25/20 06:30 John Doe	Uses the capture group (parenthesis) to just return the date, time, and user.
(\d+) Power Failures	2	Uses the capture group to get 1 one more digits (\d+) before "Power Failures."
Flash Archive Status: (.*)	Valid on 12/25/20 06:30:00	Capture all of the text after "Flash Archive Status."
Core board hardware.* $r\n.*S/N$	2002A2179P	Looking for:
(. ^)		
		Core board hardware:
		Type=126, board=36, manufactured on 12/17/2012, S/N 2002A2179P
		Finds the line with "Core board hardware".
		Matches the rest of the line (.*) followed by the CR/LF (\r\n).
		Then, matches all of the next line up to "S/N".
		Finally, captures the serial number value which is the remainder of that line.

Another example of the regular expression is to extract the driver version from the driver name. If the name was "drv_melgr_vanilla_6-00-082", then the expression "\d+-\d+" ("\d+" is one or more digits) would find "6-00-082".

Advanced Reports and Table Concepts v9.0

Carrier Proprietary and Confidential

Variables

You use variables to set parameters in your reports. A variable can be edited by the user when the report is run, or set when used as a table or chart on a graphic.

The types of variables are:

- String
- Number
- Date
- Time

A variable has an ID, default value, and if editable, a display name used as a label.

Variables can be used in the following places:

- In the "Data Range" as the number of days/weeks/months/etc or as a starting dat.
- In the "Date Range" in the TREND DATA column type
- As a substitution in a **PATH** column type (surround with "\$")
- As a substitution in an **EXPRESSION** column type (surround with "\$")
- As a substitution in a FUNCTION column type argument (surround with "\$")

input column (ID):	oops		
Function:	Default Value	~	
Arguments: (<pre>\$default_value_var</pre>	\$	

EXAMPLE Create two number variables named "high_limit" and "low_limit". Given a column with an ID of "temp", use the following expression to test if the temperature is out of range:

\$temp\$ < \$low_limit\$ || \$temp\$ > \$high_limit\$? 'out of range' : 'ok'

This concept can be used to hunt for values that are not in the range, including:

- status values
- parameters
- other configurations

Creating generic tables or reports and then customizing them for a graphic can be done using variables while designing the table in ViewBuilder. For example, a trend table can be created and the number of rows/weeks in the table can be specified.

2			
	Data Table		I
F	Report ID: zone_temp_history		
	Size Width: 300	Height: 200	
	Variables	Value 📕	
•	num_weeks	10	

Advanced Reports and Table Concepts v9.0

Carrier Proprietary and Confidential

Where

When creating a report, it can be run anywhere by default. This is typically not desirable. The **Where** option allows control of what location the report is accessible to users when the **Show in Reports menu**... option is enabled.

NOTE: When editing a report, it is always previewed from the current location regardless of its **Where** options. It just determines when it shows up in the **Report** menu.

The Where options are:

- **Anywhere** This is the default and the report will appear everywhere.
- Control Programs Select the control program types or patterns where the report will appear.
- Location Types Select the type(s) of locations where the report will appear.
- Locations Select or enter the specific locations where the report will appear.

Options

You can set other report options on the Options tab.

Show Min/Max/Avg/Total

This option appends rows to the table that calculates the selected statistic. The label for the statistics can go into any column desired. By default, the statistics are applied to all numeric columns, but only specific columns can be designated by ID as desired.

Show First ___ Rows

This option reduces the report, or table, to the number of rows specified. This is useful to show the top "N" values and to keep a table smaller in size for **Graphics** pages. The count can be a number or a variable name. If you enter a variable, it must be defined on the **Variables** tab.

Sort Column

This option sorts the report or table based on the selecting columns. The column ID can be specified more than once. If the first columns are equal, then the next column is used to sort, and so on. Values are sorted based on their natural type. Numbers sort numerically, and text sorts alphabetically. Check **Reverse Sort** to sort Z to A, ... to 1.

Filter Rows

This option filters out rows (include/exclude) based on matching a value. If a complex filter needs to be used, create a hidden expression column that results in a simple true or false value. Then, apply the filter to that hidden expression's result. Select **Include row when** or **Exclude row when** a specific column (ID) equals a specified value.

NOTE Filters are applied in the order listed. Therefore, it is typical to "sort" first and then "show first __ rows" after.

Table Calculation

The table calculation option allows custom expressions to be run after all columns have been generated.

EXAMPLE The percent of active zones based on a table where each zone has an "active" zone value.

Location	Count	Active Zone
<u>E1</u>	1	0
<u>E2</u>	1	1
<u>E3</u>	1	0
<u>E4</u>	1	1
<u>E5</u>	1	1
Total	5	3
		60% active

The last line is appended using the **Table Calculation**.

Type Show Max/Min/Avg/Total Table Calculation		
Table Calculation	✓ Add	
Variable ID:	percent_active	Holds value of Expression calculation (Example: total_variable)
Expression:	round(100 * \$activ	re_zone\$ / \$count\$) + "% active"
	Example: (\$total_var	<pre>iable\$ + \$quantity_column\$) - sum each rows quantity column</pre>
For Each Row:		
Include Appended Rows:		
Hide Variable:		
Column:	active_zone	Column ID to display calculated value (appears at bottom of table)

The calculation above wants to use the values from the **Total** row, which is an "appended" row. So **Include Appended Rows** must be selected. If not, the values in the row for "E5" would have been used.

Field	When selected
Variable ID	Each calculation must be assigned to a unique variable ID
Expression	Computes the variable value. An existing variable or column value can be referenced by surrounding the ID with "\$". The value for a column is either the value for the current row when For Each Row: is selected or it will be the last value in the column.
For Each Row	The expression is run once for each row. You must specify an Initial Value for the variable. Typically, the variable will be used in the expression and then re-assigned to itself. This is useful for custom statistical calculations. All of the Show Max/Min/Avg/Total operations can be done using this.
	EXAMPLE To manually compute the sum of column "x" into variable "sum_x", use the expression " sum_x + x " with an initial value of "0". For each row, the new value of the expression would be re-assigned to "sum_x".
Include Appended Rows	The value in rows appended by other Options are considered. Otherwise, those values are ignored.
Hide variable	The result of the expression is NOT shown in the output. Its value will be hidden from display and is typically used for intermediate calculations.
	When not selected, the value of the expression is added to the bottom of the table in the specified column. A new row is created if there is already a value in the last row for that column.

Table Calculation details

Advanced Reports and Table Concepts v9.0

Simple text can easily be added to the bottom in a similar manner. The following **Table Calculation** adds the text **Active** to the first column (ID 'location_name') of the last row:

Variable ID:	label
Expression:	'Active'
	Example: (\$total_variable
For Each Row:	
Include Appended Rows	: 🖸
Hide Variable:	
Column:	[location_name

Location	Count	Active Zone
<u>E1</u>	1	0
<u>E2</u>	1	1
<u>E3</u>	1	0
<u>E4</u>	1	1
<u>E5</u>	1	1
Total	5	3
Active		60% active

Reorder Columns

Set the order in which columns appear by listing the column IDs separated by commas.

Add appended rows to chart

Displays data from appended rows in charts, such as Maximum and Minimum.

Advanced Reports and Table Concepts v9.0

Output

Most of the output options are self-explanatory.

Ignore Page Width

Normally, a report will fill the page width of the selected page's size. Sometimes, the report will have a lot of columns that may not fit on the page and be truncated. Select this option to create a PDF file that is as wide as it needs to be. If it needs to be printed, then the printer options can be selected to print the oversized pages.

Custom Refresh Rate

Some reports or tables can take a long time to run. Others may not meaningfully change over the time period that a user will typically work with a graphic's page. The default refresh interval for table data is 30 seconds. When the data will be used on a graphics page and the data will never change (such as in a yearly trend report), set the refresh interval to zero. If the data is expensive to fetch and compute (such as current trend data), it is recommended that the refresh interval be increased. If the trend data is on a 5 minute interval, then set the refresh rate to 5 minutes (300 seconds).

Preview

You can preview at any time, and do not need to "Accept/Apply" change to preview. Use it to experiment without making permanent changes. When previewing the output, you can select "show all columns" to see the data in all of the columns including "hidden" and "color".

The "show column ID" will show the column ID below the display name. This is useful to see all of the column names including those when the type is "from existing report" or when it is on another tab and you need to find a column ID.

In order to make the preview load faster, limit the maximum number of rows to a small number. In rare occasions, the "show debug information" option may be used to show the raw JSON data that rendered the report or table. This includes all error messages and configuration information, and can be used to collect data for support.

Icons

Reports can contain icons as well as text. To show a column as an icon, the value of the column must be in icon file name.

Standard report icons:

There are currently 24 standard report icons. There are 8 color variants of a light on, off, and blinking (alarm). The names of the icons are:

- light_alarm_X.gif (animated blinking light alternates between "on" and "off")
- light_on_X.png
- light_off_X.png

NOTE X is one of "blue", "green", "Itblue", "magenta", "orange", "red", "white", and "yellow". The blinking image is a GIF and not a PNG file type.

An example of showing a green light that turns on or off based upon a value can be done easily using this expression:

\$value\$ > 50 ? "light_on_green.png" : "light_off_green.png"

Location	Value	Icon
<u>E1</u>	42.2	
<u>E2</u>	77.2	

Custom icons

Any icon image file can be placed in the "tables" directory for the current system and can be referenced by the file name. This can include sub-directories under "tables" for better organization.

Examples:

- \$value\$ > 50 ? "custom_on.png" : "custom_off.png"
- \$value\$ > 50 ? "icons/custom_on.png" : "icons/custom_off.png"



NOTE If you export a report, the images will not be copied.

System icons

Any system icon available on the web server can be referenced using its full path relative to the "webroot" directory:

 \$value\$ > 50 ? "/_common/lvl5/graphics/event_categories/general_alarm_closed.png" : "/_common/lvl5/graphics/event_categories/general_alarm_critical.png"

Location	Value	lcon
<u>E1</u>	11.2	
<u>E2</u>	96.9	

Graphics

A primary feature of reports or tables is using the report manager to create a data table for a graphic.

Tables

In the simplest case,

- 1 Add a "Data Table" control.
- 2 Enter the ID of the report or table as shown in the report manager.
- **3** Size the table as desired.

Horizontal and vertical scroll bars will appear as needed. The table will not stretch, and so it is suggested that you make the table control as tall and wide as it might need to be.

Table variable values can be set to specific values for that instance of the table in the graphic.

If you set the variable name "location" to the GQL path of a location in the system, then that table will be run from that location instead of the current location. This can be useful for tables on summary graphics. The "location" variable is automatic and does not have to be predefined when creating the table.

The table data on a page is based on the table ID and any variables set. If the same ID is used for two tables on the same page and no variables are set, or if they are all the same, then there will be only one set of data backing both tables. However, if the values in the variables differ, then there will be separates sets of data collected for each table.

Charts

Configuring charts works the same as tables.

- 1 Add a chart.
- 2 Enter the ID of the report or table.
- 3 Select a chart type.
- 4 Size the chart.
- 5 Set any variables.

You can use a custom chart type by entering the name in the "Type" field.

Colormaps

A data table can also be used to customize the colors used in the color map control. This is a powerful new feature that can be used to provide new visualizations.

To create a report or table to use with a color map control, you must do the following:

- 1 Type must be "Color Map".
- 2 There must be one or more columns that render as "Color".
- 3 The values in the "color" columns must be "web colors". A color can be an RGB hex value in the form "#RGB" or "#RRGGBB" or a standard web "color name". See https://www.w3schools.com/colors/colors_names.asp

To use it with an image, open the "Associations" dialog and then open the "Report Properties" dialog. The "Report ID" must be the ID of the table/report. By default, the first "Color" column found will be shown. A specific column can be set using the "Default Column ID" field. If there is more than one color column, a local variable of type "Report Column" can be created to select which color column will be shown. By assigning that local variable to a drop list control, you can dynamically switch between colors on the image on the graphic.

	뒄 Droplist		
Column 1 🔻	General Advanced		
	Microblock Path: (\$\$column_var\$\$	
	Property:	~ ·	
	Editable:		
	Preview Text:	Column 1 v	

That variable can then be used in "show/hide" expressions. This can be used to change labels or color legends around the image.

5 Show/Hide Condition	×
Expression:	
\$\$column_var\$\$ == 'column1'	\sim
Show When	
True	
⊖ False	
OK Cancel Help	

Advanced Reports and Table Concepts v9.0

Using table data in other controls

It is possible to reference values of a cell in a table in another control. This is useful when the unique capabilities of the report manager allows for values to be computed that would otherwise not be accessible. A common technique is to create a small table with the desired values and hide the table on the graphic. This can be easily done with a show/hide expression of "false".

To reference the value of a cell in a data table, use one of these expressions:

- CELL::table ID,column ID,column ID=value
- CELL::table ID,column ID,numerical position in the column

NOTE The numerical position in the column can be positive if counting for the top or negative if starting from the bottom.

To reference the value of 17.02 in the table below called "sample_table", below are a few ways you can reference the value:

- CELL::sample_table,c1,location=#e8
- CELL::sample_table,c1,ref=#e8
- CELL::sample_table,c1,3
- CELL::sample_table,c1,-5

Location Path	Location	RefName	Col1	Col2	C1 > C2*10
location	location_name	ref	C1	c2	c1_v_c2
<u>#e6</u>	<u>E6</u>	#e6	20.24	4.06	0
<u>#e7</u>	<u>E7</u>	#e7	43.96	0.25	1
<u>#e8</u>	<u>E8</u>	#e8	17.02	7.15	0
<u>#e9</u>	<u>E9</u>	#e9	60.78	6.16	0
<u>#e10</u>	<u>E10</u>	#e10	80.66	4.20	1
	Average		44.53	4.36	
	Total		222.67	21.82	

The negative "count from bottom" is typically used to reference the statistic values appended to the data. When locating the row using "column ID=value", the column can be hidden.

Clicking on a column to sort does not affect the references to the original data.

Miscellaneous

Tips

Use "hidden" columns for intermediate calculations. Do not try to do too many calculations in one expression.

When experimenting and learning what can be done, use **random()** in expression to generate a random number between 0 and 1. An expression like "68 + (random() * 7)" produces a number that resembles room temperature.

Maximum Number of Rows

The maximum number of rows permitted in a table is 1000. If this limit needs to be exceeded, override the limit for a single table by setting a variable name "maxrows" to the desired limit. It can be set in the **Report Editor** or as a variable for a **Data Table** or **Chart** control in ViewBuilder.

The limit for all tables can be changed by setting "table.maxrows=###" in system.properties.

The maximum number of columns is 50. It can be overridden by setting "table.maxcolumns=###" in system.properties..

NOTE "maxrows" variable support requires the i-Vu® v8.0 August 2021 cumulative update or later.

Document revision history

Important changes to this document are listed below. Minor changes such as typographical or formatting errors are not listed.

Date	Торіс	Change description	Code*
		No changes yet	

* For internal use only



Carrier ©2024 · Catalog No. 11-808-1068-01 · 5/9/2024