Data coercion

Data coercion

IMC-040B-UM.1
$Date: 2021/01/13 16:42:49 $
$RCSfile: Coercion.xml,v $

eRAD Inc

201 Brookfield Pkwy Suite 160
Greenville, SC 29607
www.erad.com

Revision History

Revision 1.0	Nov 17, 2010	BM
Submitted		
Revision 1.0.1	Nov 18, 2010	BM
Biblio ID		
Revision 1.0.2	Dec 10, 2010	TN
Copyright information has been moved to individual documents		
Revision 1.1	Nov 19, 2015	BM
New functions, Viewer rules added		
Revision 1.2	Jun 11, 2018	BM
mul, div, mod functions added (Scheduled for release in medsrv-7.3.4 and medsrv-8.0.7)		
Revision 1.3	Jan 13, 2021	BM
toUpper and toLower functions added (Scheduled for release in medsrv-8.0.22)		

Table of Contents

1. Data coercion in general

Data coercion rules insert, remove and modify values in DICOM attributes when the system first acquires the object. Rules are assigned globally or to a specific registered device. Coercion works for DICOM alpha-numeric attributes only. Multiple rules can exist for a single device, and are applied in the defined sequence.

Three sets of global coercion rules exist. The first rule set is the Preceding Global Coercion Rules and is applied before any device-specific rules are applied. The second set of rules is the Trailing Global Coercion Rules, applied after the device-specific rules. Global coercion rules use the same commands as the device-specific rules. They are configured in the Preceding Global Coercion Rules and Trailing Global Coercion Rules sections of the Devices page. To assure administrators know these rules exist, they appear in uneditable tables before and after device-specific coercion rules on the Edit Device page.

The 3rd set is the Viewer Global Data Coercion Rules. These rules apply in the Viewer. That means they will not be saved to the study data, but modfy the appearance of the study in the Viewer.

Device-specific coercion rules are available for registered DICOM devices. They get applied after preceding global coercion rules and before trailing global coercion rules.

To define coercion rules, perform the following steps.

1. From the Admin tab, click the Devices tab.

2. If defining global coercion rules,

2.1. Scroll to the Preceding Global Coercion Rules section or the Trailing Global Rules section or the Viewer Global Data Coercion Rules.

2.2. Define and save coercion rules as described below.

3. If defining device-specific coercion rules,

3.1. Find the device on the Devices table and click on the edit button, , to the left of the name.

3.2. Scroll to the Data Coercion Rules section.

3.3. Define and save coercion rules as described below.

To define coercion rules, perform the following steps.

1. To see a list of available coercion commands, click the Help button.

2. Type the rules into the text box. Use the pulldown list and Add Tag button to select the attribute tag, if needed.

3. Click Save. If there is a detectable Syntax error in the created coercion rule, it will be displayed after pressing Save (but the coercion with the bad syntax will not be saved).

Note: Data coercion rules do not propagate between servers. If they are required on multiple eRAD PACS servers, you must make the change on each one.

2. Data coercion syntax

General Syntax:

- comment lines

  Lines starting with '#' are treated as comments
- <lvalue>=<expression>

  Expression can be either a Value or a Function.

Data Types:

- String

  Group of ASCII-numeric and control characters. When non-contiguous, strings must be encapsulated within double quotes (""). Control characters must be preceded by a backslash, as in "\n", "\\" and "\"".
- Integer

  Numeric value used and returned in some expression functions, including arithmetic functions.
- gTag

  The first part of a DICOM tag representing the attribute group. Expressed as four hex digits.
- eTag

  The second part of a DICOM tag representing the attribute element. Expressed as four hex digits.

lvalues:

- (gTag,eTag)

  DICOM attribute tag. The tag value for DICOM defined attributes and eRAD PACS's pre-defined user fields, User1-10. This lvalue is in the form "(gggg,eeee)".
- SEQ(gTag1,eTag1, INo1,gTag2,eTag2 [,INo2,gTag3,eTag3...])

  A target attribute specified by Tag2 within the INo item in the sequence specified by Tag1, where 0 represents the first item in the sequence. If the sequence Tag1 does not exist in the object, the entire rule is ignored.
- USER(fieldName)

  User defined field. A custom configured user field, specifically, one that do not exist in the DICOM library. This lvalue is in the form "USER(fieldName)", where fieldName is the Field Name defined in the custom user fields configuration file.
- $(varName)

  Temporary Variable. A variable used to temporarily hold an evaluated expression. Temporary values are not stored in the object. This lvalue is in the form "$(varName)", where varName is the name of the variable.

Values:

- (gggg,eeee)

  Returns the value of a DICOM tag. If the DICOM tag does not exist, NULL() is returned.
- SEQ(gTag1,eTag1, INo1,gTag2,eTag2 [,INo2,gTag3,eTag3...])

  Returns the contents of the DICOM attribute specified by Tag2 within the sequence specified by Tag1. INo indicates the sequence item instance, where 0 represents the first item in the sequence. Returns NULL() if the attribute does not exist in the INo item of the sequence.

- USER(fieldName)

  Returns the value of a user defined field. If the field does not exist, NULL() is returned.

- $(varName)

  Returns the value of a temporary variable. If the variable is not set, NULL() is returned.

- Quoted string

  This is in the form, "`string`". Returns the literal string `string`. The string can contain escaped characters, including "\n", "\\" and '\"'.

- Unquoted string

  Non-quoted strings can be used when they contain only contiguous, alphanumeric characters.

- (gggg,eeee),"d",n (Retired)

  This returns the `n`th field in the DICOM tag `(gggg,eeee)` value as separated by the delimiter `d`.

Basic Functions:

- NULL()

  Deletes the target lvalue.

- empty string

  The empty string, "".

Logical Functions:

- and(a,b)

  Returns true if both `a` and `b` are non-NULL.

- equals(a,b)

  Returns true if `a` and `b` are equal, NULL if they are not equal.

- if(cond,a,b)

  Returns `a` if `cond` is not NULL, `b` if `cond` is NULL.

- not(a)

  Returns true if `a` is NULL, NULL if `a` is not NULL.

- or(a,b[,c...])

  Returns the first non-NULL value.

String Functions:

- concat(a,b[,c...])

  Concatenates the values `a` and `b` (and `c`, etc.) NULL values are treated as an empty string, "".

- contains(a,b)

  Returns `b` if `b` is found in `a`, NULL otherwise.

- indexof(a, ss)

  Returns the starting position of `ss` in `a`, or -1 if not contained.

- split(a,d,n)

  Returns the `n`th field in `a` using `d` as the field delimiter.

- strlen(s)

  Returns the length of `s`, or NULL if `s` is NULL.

- substr(s, p [,n])

  Returns the given section of `s` starting from the position `p` (0 based, 0 means the beginning of the string), in the length specified by `n`.

- translate(a,d,i1,o1 [,i2,o2...])

  Returns the output value, oN, if the input string, iN, matches the source string, a. If the source string matches no input string, the function returns the default value, d. Source string, a, and input strings, iN, must be String types or NULL(). The default value, d, and output values, oN, can be any type but must all be of the same type.

- toUpper(s)

  Returns the upper-case version of s.

- toLower(s)

  Returns the lower-case version of s.

Date Functions:

- dicomAge(d1, d2)

  Returns the value `d1 - d2` in calendar years, months or days in DICOM-compliant format: nnnY, nnnM, or nnnD. If either `d1` or `d2` is an invalid date value, or `d2` predates `d1`, NULL is returned.

Arithmetic Functions:

- add(n,m[,o...])

  Returns the sum of the parameter integer strings.

- sub(n,m)

  Returns `n - m`, where `n` and `m` must be integer strings.

- between(v,n,m)

  Returns true if `v` is greater than or equal to `n` and `v` is less than `m`, NULL otherwise. `v`, `n` and `m` are integer strings.

- mul(n1,n2[,n3...])

  Scheduled for release in medsrv-7.3.4 and medsrv-8.0.7.

Multiply all operands. Operands must be Number types.

- div(n1,n2)

  Scheduled for release in medsrv-7.3.4 and medsrv-8.0.7.

  Divide the operands using integer division, n1/n2. Operands must be Number types. Division by zero is checked, but zero-value denominators will return an error.

- mod(n1,n2)

  Scheduled for release in medsrv-7.3.4 and medsrv-8.0.7.

  Return the remainder of the integer division, n1/n2. Operands must be Number types. Division by zero is checked, but zero-value denominators will return an error.

Encoding Functions:

- codenumber(n)

  `n` must be a >= 0 integer string, returns a coded >= 0 integer string with the same number of digits.

- codestring(s[,exc])

  `s` is a string, returns a coded string with the same length. If `exc` is used, the result will not include these characters.

Misc Functions:

- rnd(n[,seed])

  Returns a random number (integer string) between 0 and `n`-1. If `seed` is used, the random number will be calculated from it.