Rocket Uniface Library 10.4

$dbgString

Get a string that represents the Struct or Struct collection, including annotations.

Struct `->` `$dbgString`

Return Values

Returns a string that can be used for debugging.

Table: Values of $procerror Commonly Returned Following Struct Functions

Value	Error Constant	Meaning
`-84`	`UACTERR_NO_OBJECT`	*Struct* refers to zero Structs
`-1151`	`USTRUCTERR_NO_COMMON_CHARACTERISTICS`	Collection of Structs that do not share a common parent or the specified characteristic
`-1157`	`USTRUCTERR_ILLEGAL_MEMBER_TYPE`	Not a valid Struct member type

Description

Use `$dbgString` when you need to verify the structure and annotations of a Struct.

> **Note:** If you do not need to see the annotations, use the Struct function `$dbgStringPlain`.

`$dbgString` is intended for use during development, to visually represent and format the contents of a `struct` variable. It can be used to display the Struct as string, for example, in the message frame.

> **Note:** It is not intended to be a serialized form of the Struct. It is not possible to recreate a Struct from a string that was created this way.

The returned string shows a nested structure that includes both the normal Struct and the `$tags` Structs:

- The name of the Struct is printed on the first line between square brackets, with proper indentation.

  A Struct leaf is followed by an equal sign (=) and its value.

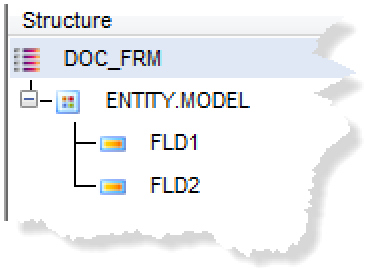
- String values are in double quotes.
- All other data types (numeric/float, date, raw, and so on) are displayed without quotes

  A node with no members and no value is considered to hold an empty string

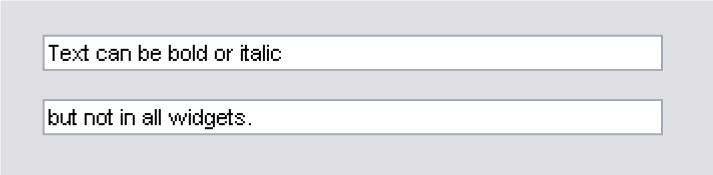
- If annotations exist, the `$tags` Struct is displayed as the first Struct under a node; it precedes the member list.

Uniface Component Struct

For example, given the following component structure:



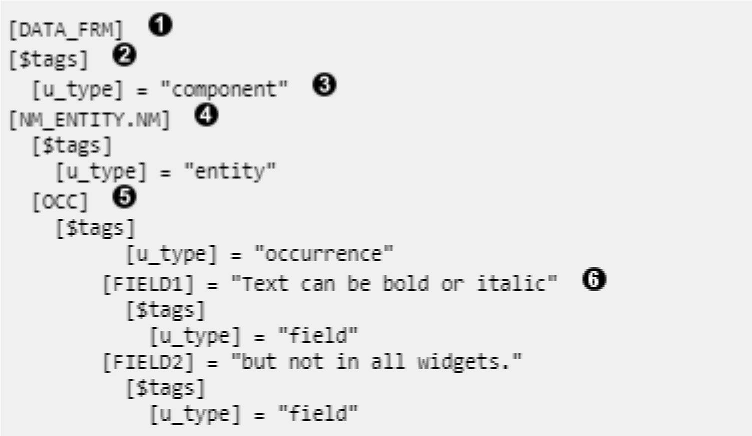
And the following runtime data:



Text can be bold or italic

but not in all widgets.

The Struct function `$dbgstring` returns a formatted string that represents the Struct:



```
[DATA_FRM] ❶
[$tags] ❷
  [u_type] = "component" ❸
[NM_ENTITY.NM] ❹
  [$tags]
    [u_type] = "entity"
  [OCC] ❺
    [$tags]
          [u_type] = "occurrence"
        [FIELD1] = "Text can be bold or italic" ❻
          [$tags]
            [u_type] = "field"
        [FIELD2] = "but not in all widgets."
          [$tags]
            [u_type] = "field"
```

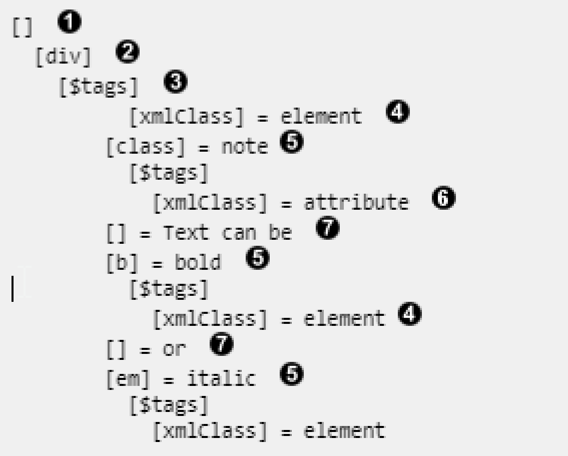
1. Named top-level Struct with name of component
2. `$tags` Struct containing annotations for the member
3. `u_type` annotation indicates the component object that the Struct member represents
4. Named Struct node for entity
5. Struct node for occurrence. The name is fixed to `OCC`
6. Struct leaf for field

String Returned for an XML Struct

For example, consider the following XML code:

```
<div class="note">Text can be <b>bold</b> or <em>italic</em></div>
```

The Struct function `$dbgstring` returns a formatted string that represents the Struct:



```
[] ❶
  [div] ❷
    [$tags] ❸
          [xmlClass] = element ❹
        [class] = note ❺
          [$tags]
            [xmlClass] = attribute ❻
        [] = Text can be ❼
        [b] = bold ❺
          [$tags]
            [xmlClass] = element ❹
        [] = or ❼
        [em] = italic ❺
          [$tags]
            [xmlClass] = element
```

1. Top-level Struct.
2. Struct node.
3. `$tags` Struct containing annotations for the member.
4. `xmlClass` shows that the member represents an XML element.
5. Struct leaf, with its value being the only member.
6. `xmlClass` shows that the member represents an XML attribute.

Related concepts

$dbgStringPlain