Rocket software

Rocket Uniface Library 10.4

componentToStruct

Writes component data to a Struct.

```
componentToStruct {/mod} {/one} {/reconnecttags} {/firetriggers}   StructTarget  {, EntityName}
```

Example: `componentToStruct /mod /reconnecttags /firetriggers vStruct, EMPLOYEE.ORG`

Qualifiers

Table: Qualifiers

Qualifier	Description
`/mod`	Include only modified occurrences, and their ancestors. Ancestors are included to provide context for the modified Structs.
`/one`	Include only the current occurrence of the named entity. This qualifier only affects the named entity; for inner occurrences all occurrences are always included. If no qualifier is specified the switch has no effect.
`/reconnecttags`	Adds reconnect processing tags to occurrence members (the `u_type=occurrence` annotation must be present), and includes occurrences marked as deleted in the Struct. If omitted, these tags are not generated and occurrences marked as deleted are not included in the generated Struct.
`/firetriggers`	Causes the `preSerialize` and `postSerialize` triggers to be fired. These triggers can be used to provide additional processing, for example when preparing data to be loaded and reconnected into a component that contains data.

Parameters

Table: Parameters

Parameter	Data Type	Description
StructTarget	`struct` or `any`	Variable, parameter, or non-database field to hold the generated Struct.
EntityName	`String`	Name of the entity to convert. If specified, conversion starts at the specified entity, which is not necessarily a top level entity. When no *EntityName* is specified, conversion starts at component level; the top level struct has the name of the component, and it includes all top level entities of the component as members.

Return Values

Table: Values Commonly Returned in `$status` after `componentToStruct`

Value	Meaning
`0`	Struct successfully created.
`<0`	An error occurred. `$procerror` contains the exact error.
`-1102`	Entity not valid if a non existing entity is specified as the second parameter.

Use

Allowed in all component types.

Description

The `componentToStruct` statement writes occurrence data in the component instance to a Struct. If no qualifiers are used, the Struct is built from the complete hitlist, including occurrences currently marked for deletion.

> **Note:** Static fields and control fields are skipped when using `componentToStruct`.

In most cases, `componentToStruct` changes the active occurrence to the first occurrence. However, when `/one` is used, the active occurrence remains unchanged.

Conversion

During conversion `componentToStruct` converts Uniface objects to Struct nodes as indicated by the following table. Each Struct member has a `u_type` annotation, which defines the original object type.

Table: Component to Struct Conversion

Uniface Object	Struct	Value of `u_type` Annotation
Component	Named Struct, with the name of the component. This node is not created if *EntityName* is specified.	`component`
Entity	Named Struct, with name of fully qualified entity	`entity`
Occurrence	Named Struct, with name `OCC`	`occurrence`

Uniface Object	Struct	Value of `u_type` Annotation
Field	Named Struct, with name of field	`field`

Struct Annotations

By default, `componentToStruct` generates the `u_type` annotations. When the `/reconnecttags` switch is used, it adds annotations for reconnect attributes.

Annotations can be accessed using `$tags`. For example, in the following code, `vType` contains the object type of the first member of the Struct:

```
componentToStruct MyStruct
vType = MyStruct->*{1}->$tags->u_type
```

Table: Annotation Tags for Uniface Component-Struct Conversions

Tag	Allowed Values	Comments
u_type	`component` | `entity` | `occurrence` | `field`	Each node in a component Struct has a u_type annotation that indicates the object type.
For nodes that have the tag `u_type="occurrence"`, the following tags are also supported. These can be used if you are using the Struct to manipulate data prior to a reconnecting the data to its source. For more information, see Metadata for Reconnect.		
u_id	*OccID*	Uniface-generated occurrence identifier
u_crc	*CheckSum*	CRC checksum of the occurrence
u_status	`est` (exists in DB) `mod` (modified) `new` (new) `del` (delete)	Modification status of the occurrence.

Triggers Fired by `componentToStruct/firetriggers`

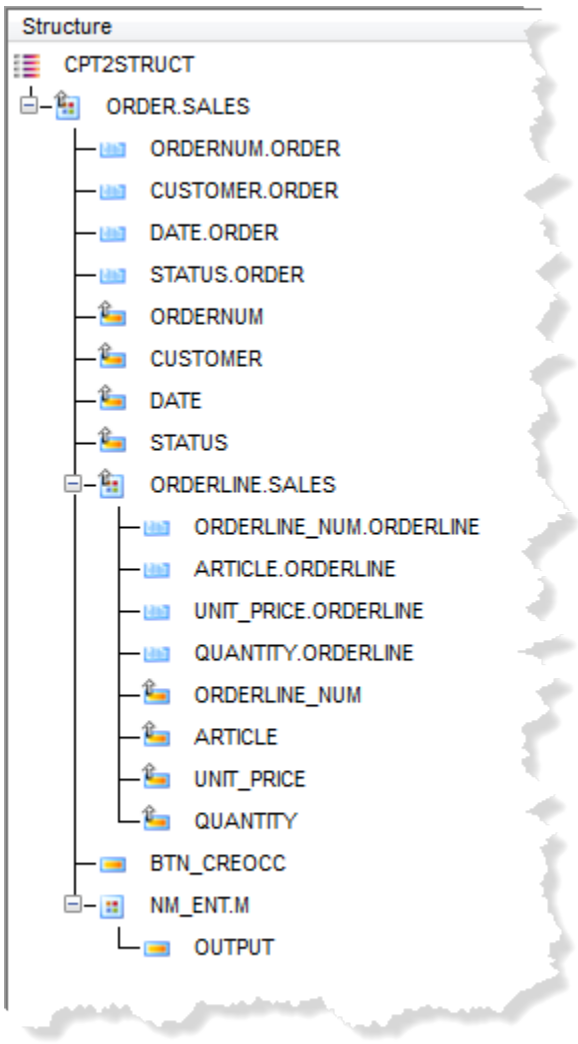
The `componentToStruct` statement only fires triggers if the `/firetriggers` switch is specified.

- trigger preSerialize—fired immediately before a Struct member is generated for an occurrence. For example, you could use this trigger to exclude an occurrence, or calculate the value for a derived field.
- trigger postSerialize—fired immediately after a Struct member is generated for an occurrence.

Example: Converting a Component Structure to a Struct

The component structure of the CMP2STRCT component includes an ORDER entity and its ORDERLINEs:

Figure: Component Structure



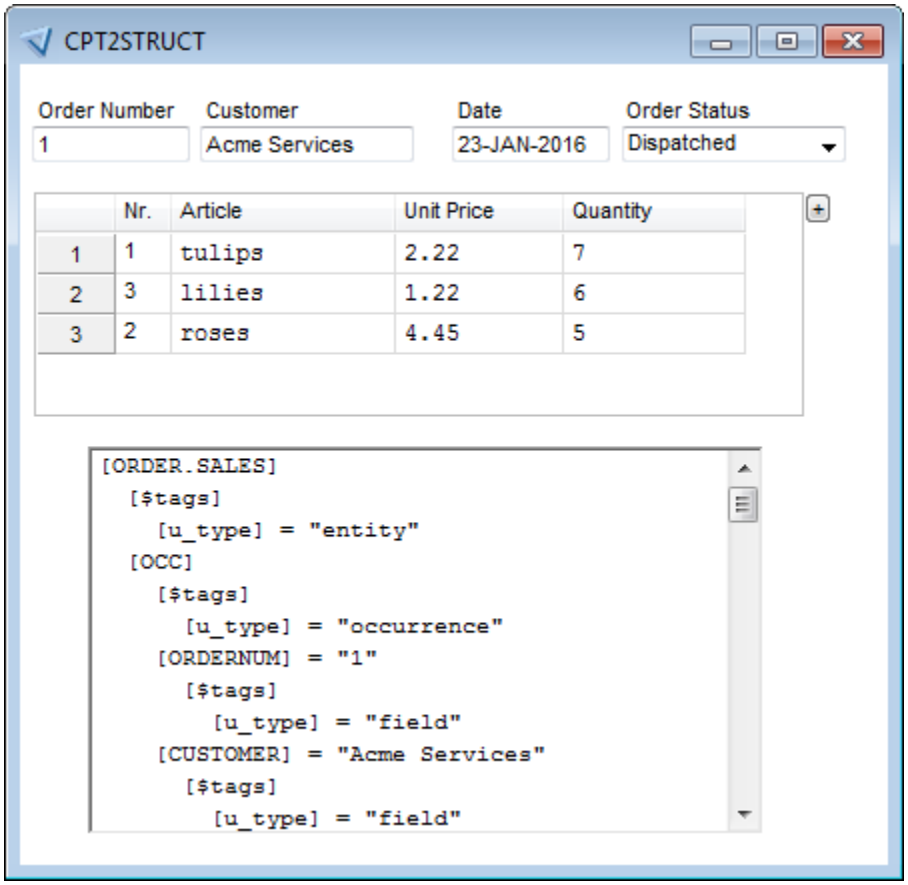
The following code in the component's exec operation, converts only the current occurrence of ORDER.SALES to a Struct

```
operation exec
  ; component variable $vStruct$ is struct
  retrieve
  componentToStruct/one $vStruct$, "ORDER.SALES" ;  Specifying the /one switch
                        and the specific entity results in a Struct whose top-level Struct repr
esents an entity, not the
                        component. Although only one ORDER entity occurrence is included in the
Struct, all ORDERLINE
                        occurrences for the ORDER occurrence are included.
  OUTPUT = $vStruct$->$dbgstring                ;  The $dbgString Struct
                        function returns a representation of the Struct, which is displayed in
```

```
the OUTPUT field.
  edit
end; exec
```

1. Specifying the `/one` switch and the specific entity results in a Struct whose top-level Struct represents an entity, not the component. Although only one ORDER entity occurrence is included in the Struct, all ORDERLINE occurrences for the ORDER occurrence are included.
2. The `$dbgString` Struct function returns a representation of the Struct, which is displayed in the OUTPUT field.
3. Notice that the ORDER_ID field, which is used for the foreign key, is included, although it is not explicitly present in the component structure.

Figure: CMP2STRUCT Form




```
[ORDER.SALES]
  [$tags]
    [u_type] = "entity"
  [OCC]
    [$tags]
      [u_type] = "occurrence"
    [ORDER_ID] = "23"
      [$tags]
        [u_type] = "field"
    [DATE] = "20101201"
      [$tags]
        [u_type] = "field"
```

```
    [STATUS] = "02"
      [$tags]
        [u_type] = "field"
    [ORDERLINE.SALES]
      [$tags]
        [u_type] = "entity"
      [OCC]
        [$tags]
          [u_type] = "occurrence"
        [LINE_ID] = "1"
          [$tags]
            [u_type] = "field"
        [ITEM_NAME] = "tulips"
          [$tags]
            [u_type] = "field"
        [UNIT_PRICE] = "2.22"
          [$tags]
            [u_type] = "field"
        [QUANTITY] = "7"
          [$tags]
            [u_type] = "field"
        [ORDER_ID] = "23"     ; Callout 3
          [$tags]
            [u_type] = "field"
      [OCC]
        [$tags]
          [u_type] = "occurrence"
        [LINE_ID] = "3"
          [$tags]
            [u_type] = "field"
        [ITEM_NAME] = "roses"
          [$tags]
            [u_type] = "field"
        [UNIT_PRICE] = "4.45"
          [$tags]
            [u_type] = "field"
        [QUANTITY] = "5"
          [$tags]
            [u_type] = "field"
        [ORDER_ID] = "23"      ; Callout 3
          [$tags]
            [u_type] = "field"
...
```

Related concepts

- Transforming Complex Data Using Structs
- structToComponent

Related tasks

- Structs for Uniface Component Data

Related reference

- trigger preSerialize
- trigger postSerialize