

**UNIFACE**

Advanced Development Technology

**Uniface 10.3.01**

What's new?

Henk van der Veer

# Uniface 10.3

- Maturing and Expanding functionality
- IDE performance
  - Large repositories
  - Overall IDE developer experience
- Final version of repository - Structure, optimization
- Create table utility
- Umeta.xml
- Graphical Form Painter Integration

# Topics

- Final version of the repository
- Embedded Graphical Form Painter
- SQL generation (Create Tables)
- Test mode in the IDE

# Final version of the repository

(1/6)

Objective:

- umeta.xml (DICT model) available for customers
  - Create reports from the repository
  - Develop custom tools for repository
- enable conversion to 'new style' editors without need for migration.
- Preparation for Work Area Support
- New documentation

# Final version of the repository

(2/6)

Main changes:

- Dropped tables
  - UCSCCH - application model
  - UXREGS - Component variables (moved to component Declarations container)
- Changed tables: New fields, removed fields
  - UCGROUP/UXGROUP – modeled entities, component entities
  - UCFIELD/UXFIELD – modeled fields, component fields
  - UFORM – components

# Final version of the repository

(3/6)

Main changes (continued):

- New tables
  - Typed libraries for global objects
  - Different tables for each global object type
- New object types: Palettes and Templates (used in resource Browsers)
- Field level templates for Interface, Syntax and Layout contained in a library (10.3: hard coded library name 'LIB')
- Modification timestamp moved from individual global object to its parent library

# Final version of the repository

(4/6)

Main changes (continued):

- Reduced number of code containers for Entities, Fields, Components and Application Shells

Object	Uniface 9.7	Uniface 10
Application Shell	8	1
Component	20 (Service: 9, Server Page: 7)	2
Entity	27	3
Field	18	2

# Final version of the repository

(5/6)

Entity code containers

Uniface 9.7

- Triggers
- Write
- Validate Key
- Read
- Lock
- Leave Modified Key
- Delete
- Write Up
- Validate Occurrence
- Pre Save Occurrence
- Pre Load Occurrence
- Post Save Occurrence
- Post Load Occurrence
- On Error
- Occurrence Operations
- Occurrence Gets Focus
- Local Proc Modules
- Leave Modified Occurrence
- Extended Triggers
- Delete up
- Defines
- Collection Operations
- <Remove Occurrence>
- <Menu>
- <Help>
- <Detail>
- <Add/Insert Occurrence>

Uniface 10

```
1  ┌─ Declarations
2  └─
3  ┌─ Collection Script
4  └─
5  ┌─ Occurrence Script
6  └─ trigger read
7      ; This trigger is fired as part of the retrieve process
8      ; Make sure not to leave the context of this occurrence
9
10     read
11
12     end
13
14 ┌─ trigger lock
15     ; This trigger is fired as part of a database lock req
16     ; Make sure not to leave the context of this occurrence
17
18     lock
19     if ($status = -10) reload
20
21     end
22
23 ┌─ trigger validateKey
24     ; This trigger is fired as part of the validate/store
25     ; Make sure not to leave the context of this occurrence
```



# Final version of the repository

(6/6)

Component code containers

Uniface 9.7

- Triggers
  - ▶ Operations
  - ▶ <Store>
  - ▶ <Retrieve>
  - ▶ <Retrieve Sequential>
  - ▶ <Erase>
  - ▶ <Clear>
  - ▶ <Accept>
  - ▶ Set State
  - ▶ Local Proc Modules
  - ▶ Get State
  - ▶ Form Loses Focus
  - ▶ Form Gets Focus
  - ▶ Execute
  - ▶ Defines
  - ▶ Asynchronous Interrupt
  - ▶ <UserKey>
  - ▶ <Quit>
  - ▶ <Pulldown>
  - ▶ <Print>
  - ▶ <Menu>

Uniface 10

```
1  ┌─ Declarations
2  │ ; Your component defines here (optional)...
3  │
4  │ ; Your component VARIABLES block here (optional)
5  └─
6
7  ┌─ Script
8  │ ┌─ operation exec
9  │ │
10 │ │ ; Your pre-edit code here...
11 │ │
12 │ │ ; Shows the form component and allow data entr
13 │ │ edit
14 │ │
15 │ │ ; At this point, $status is:
16 │ │ ;   9 when the trigger accept performend a clo
17 │ │ ;  10 when the trigger quit performed a close
18 │ │
19 │ │ ; Your post-edit code here...
20 │ │
21 │ │ ; Return execution status:
22 │ │ ;   return 0   ; return >= 0 typically indica
23 │ │ ;   return -1 ; return < 0 typically indicat
24 │ │ return 0
25 │ └─
26 └─ end
```

# Embedded Graphical Form Painter

## Uniface 10.2

- support for simultaneously opening multiple forms in development environment
- Single GFP
- GFP in stand alone window: focus problems
- worksheets *Define Structure* and *Define Frames* partially overlapping

# Embedded Graphical Form Painter

## Uniface 10.3

- 1 Worksheet for Graphical Form Painter *and* properties of all elements on the paint
- Each Form editor has its own Graphical Form Painter

# SQL generation (Create Tables)

- Command line mode
- SQL scripts for Uniface Repository
- SQL scripts for user defined entities

# SQL generation Command line

```
/gensql {/meta} createTable entity.model connector {fileName}
```

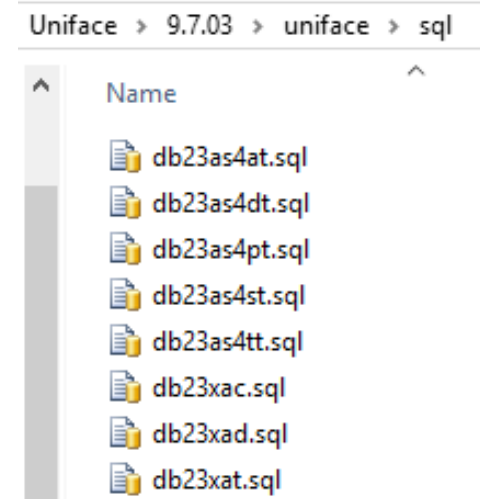
Examples:

```
/gensql /meta createTable *.dict MSS
```

```
/gensql createTable *.mymodel ORA
```

# SQL generation for Uniface repository

- No more sql files in product installation set



- Generation takes database driver parameters (USYS\$XXX\_PARAMS) into account
- Still to come: generate Referential Integrity scripts

# Test mode in the IDE

- Invoked via *Actions* menu
- Controlled via asn file
- Configurable:
  - executable
  - .ini file
  - .asn file
  - For USP/DSP: tomcat port#

# Test mode in the IDE – assignment file

```
TEST_EXECUTABLE_FRM = C:\Uniface\common\bin\ide.exe /ini=test.ini /asn=test.asn /tst %CPTNAME
```

```
TEST_EXECUTABLE_SVC = C:\Uniface\common\bin\ide.exe /ini=test.ini /asn=test.asn /tst %CPTNAME
```

```
TEST_EXECUTABLE_DSP = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"  
http://localhost:8228/uniface/wrd/%CPTNAME
```



# THANK YOU & QUESTIONS



[uniface.com](http://uniface.com)

