



GUPTA

Roadshow 2015

Team Developer 6.3

Helmut Reimann

Was ist neu?

- **Überarbeitete IDE**
- **Debugger Erweiterungen**
- **Farbunterstützung: Tasks & Bookmarks**
- **File Handling und Libraries**
- **Produktivitätserweiterungen**
- **Constructor & Destructor**
- **Erweiterte Ribbon Bar**

Was ist neu?

- **Neues Gallery Control**
- **Neue Ribbon Bar API**
- **Grid Erweiterungen**
- **Date/Time Picker Erweiterungen**
- **Neues Theme**
- **DBPipe32**
- **Report Designer**

“Preferences” Dialog

- Dialog wurde vergrößert und aufgeräumt:

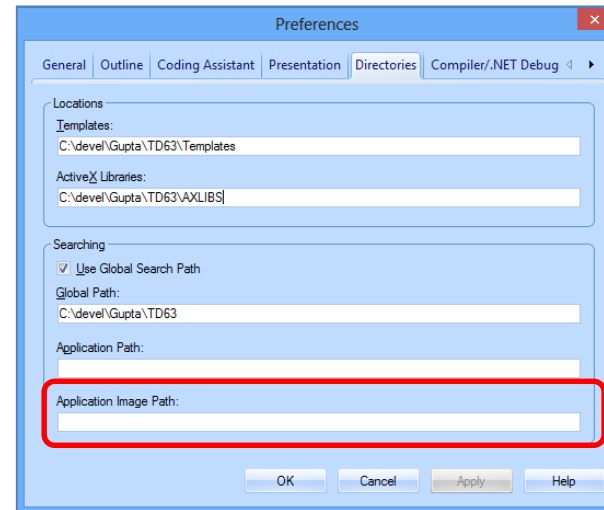
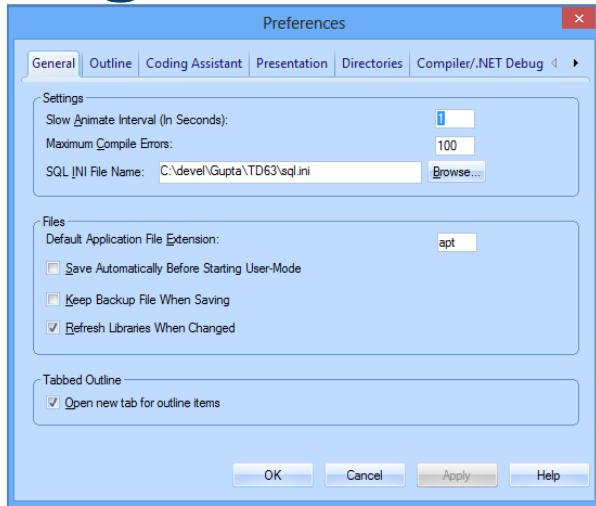
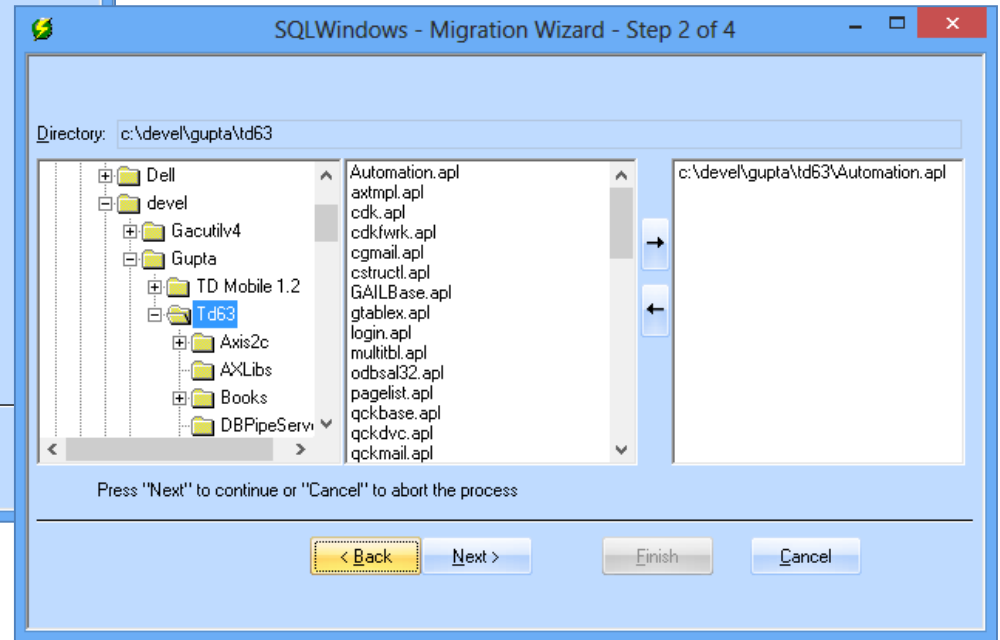
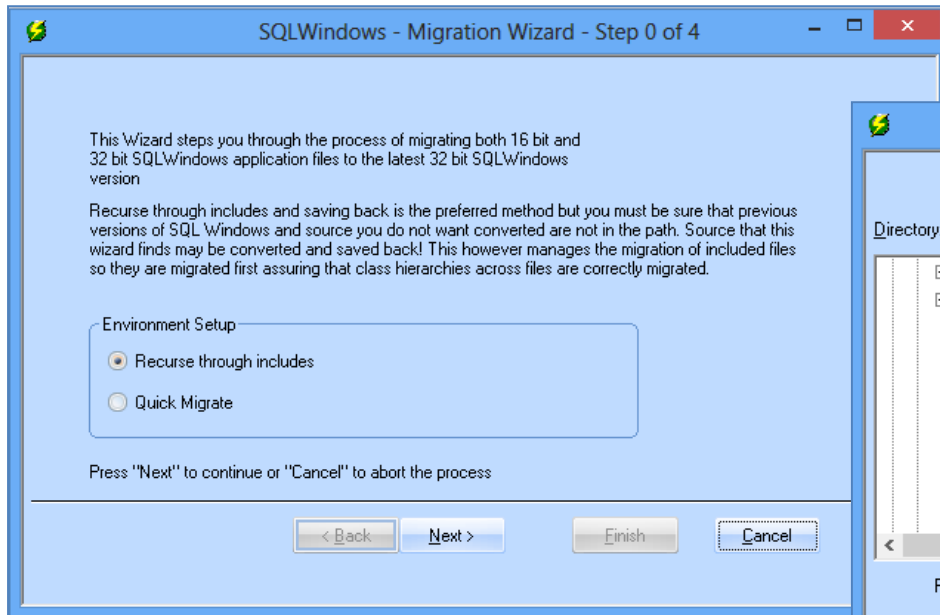


Image Path für Anwendung

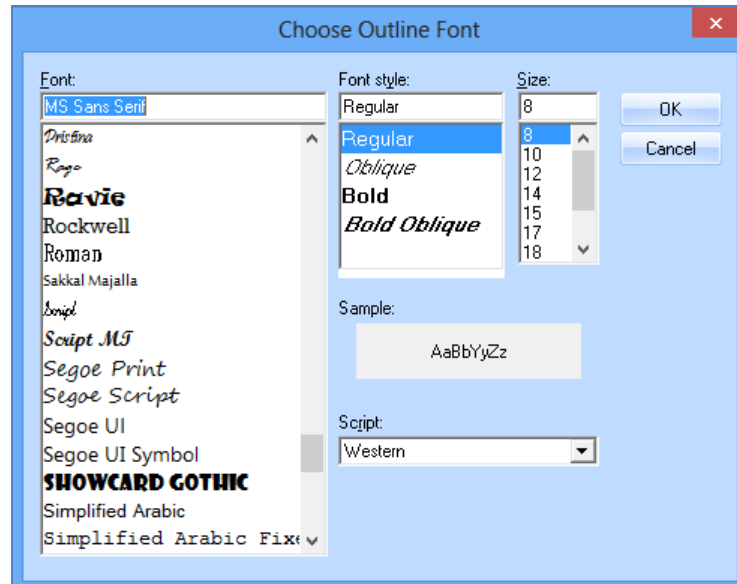
- **Absoluter oder relativer Pfad**
- **Grundeinstellung: Pfad ist leer**
 - Gleiches Verhalten wie bisher
- **Wenn der Pfad gesetzt ist**
 - **Open File Dialog für die Auswahl eines Images für ein Control:**
zeigt auf den Image Pfad
 - **Wenn der Pfad eines Images nicht gesetzt ist, dann wird TD zuerst im Image Pfad nach dem Image suchen**
 - **Wenn ein Image aus dem Image Directory gewählt wird, so wird der Pfad entfernt**

Neuer Migrations Wizard



„Themed“ Dialoge

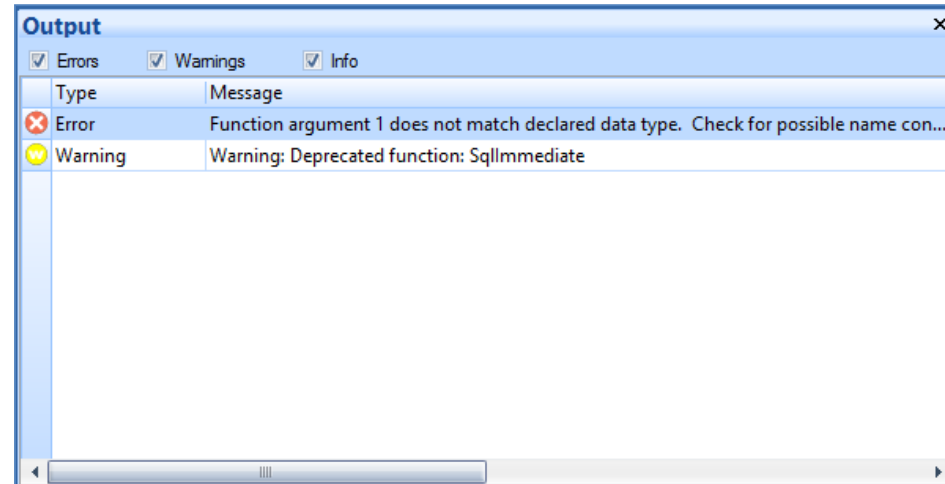
- Einheitliches Look & Feel:



Compiler

- **Compiler Ausgabe:**

- Grid statt Liste
- Drei Typen von Informationen: Errors, Warnings & Info
- Toggle Buttons, um die Anzeige entsprechend der Typen umzuschalten
- Sortieren der Meldungen



Show Item Information

- **"Show Item Information" Funktion ist nun auch während des Debug Modes verfügbar**
- **Filename wird während des Debuggings auf Wunsch angezeigt**

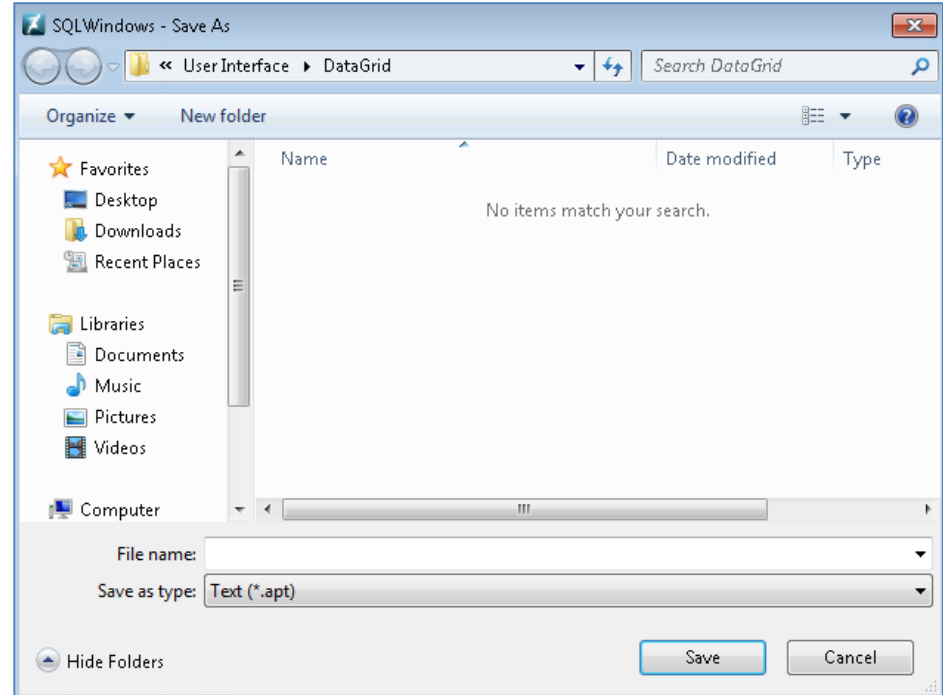
Show Item Information

The screenshot shows the 'Component' menu in Gupta Team Developer 6.2. The 'Show Item Information' option is highlighted with a green rectangle. The menu also includes options like 'New', 'Wizards...', 'QuickObject Editor...', 'Menu Editor...', 'Go To Item', 'Refresh Libraries', 'Merge Libraries', 'Large Icons', 'Small Icons', 'List', and 'Details'. Below the menu, a list of file includes is visible, such as 'File Include: vtrsw.apl', 'File Include: vtmenu.apl', etc.

The screenshot shows the 'Component' menu in Gupta Team Developer 6.3. The 'Show Item Information' option is highlighted in yellow. A tooltip is displayed over the option, stating: 'Show information about the library of the selected item'. The background shows the 'Outline' window with a tree view of the application structure, including 'Symbol Import: vtlbx' and 'Class Definitions'. A callout box at the bottom right contains the text: 'Call SalMessageBox('Show Item Information now enabled during debug', 'test', MB_Ok)'. The title bar of the window reads 'Gupta Team Developer 6.3 [Break] - Show Item Information enabled during debug'.

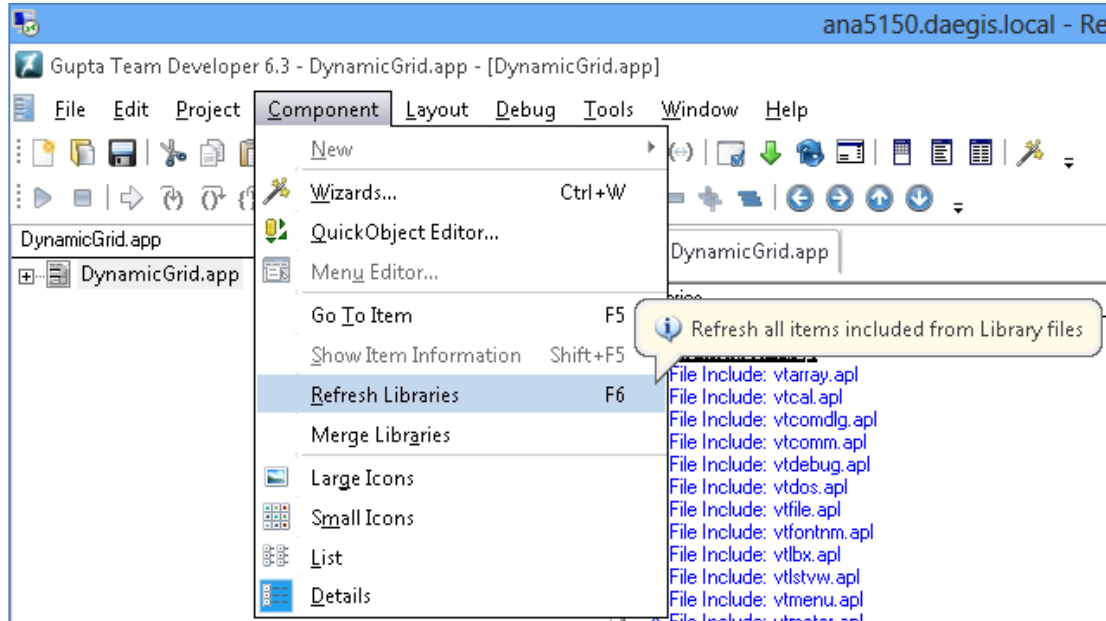
File Handling

- **“Save As” Dialog**
 - Sourcecode wird als Text abgespeichert
 - Text Format wird bei allen neuen Anwendungen als Standard genommen
 - Existierende Anwendungen sind nicht betroffen



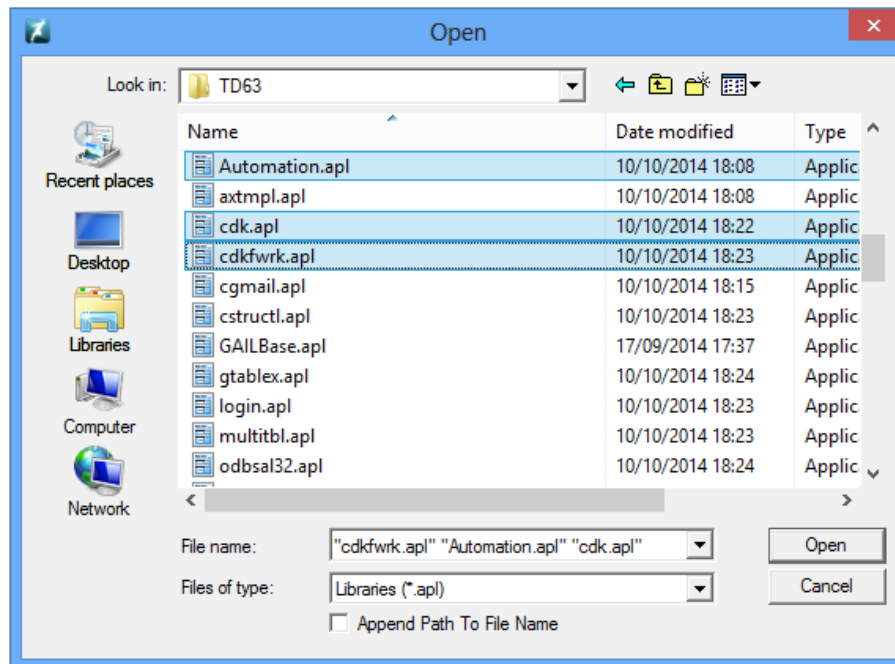
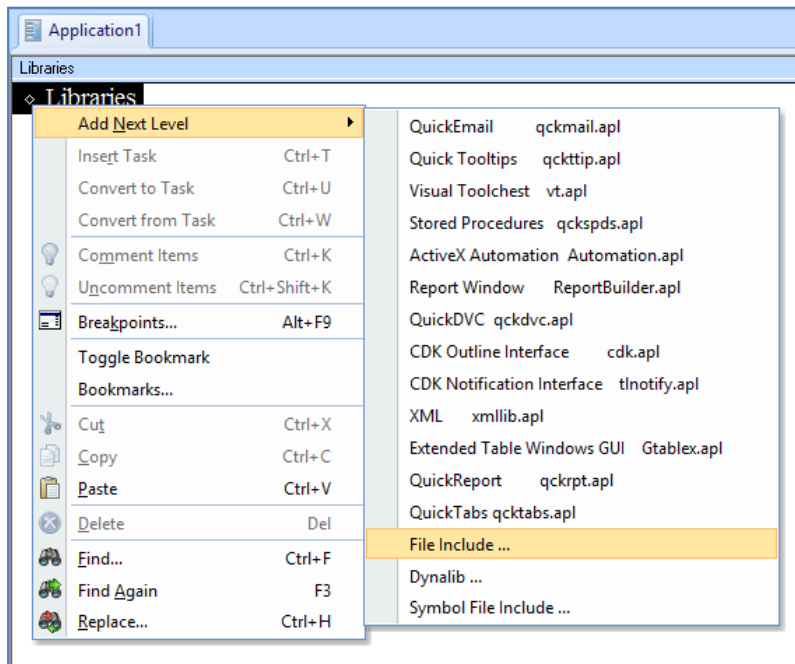
Libraries

- Neue Funktionstaste für “Refresh Libraries”



Libraries

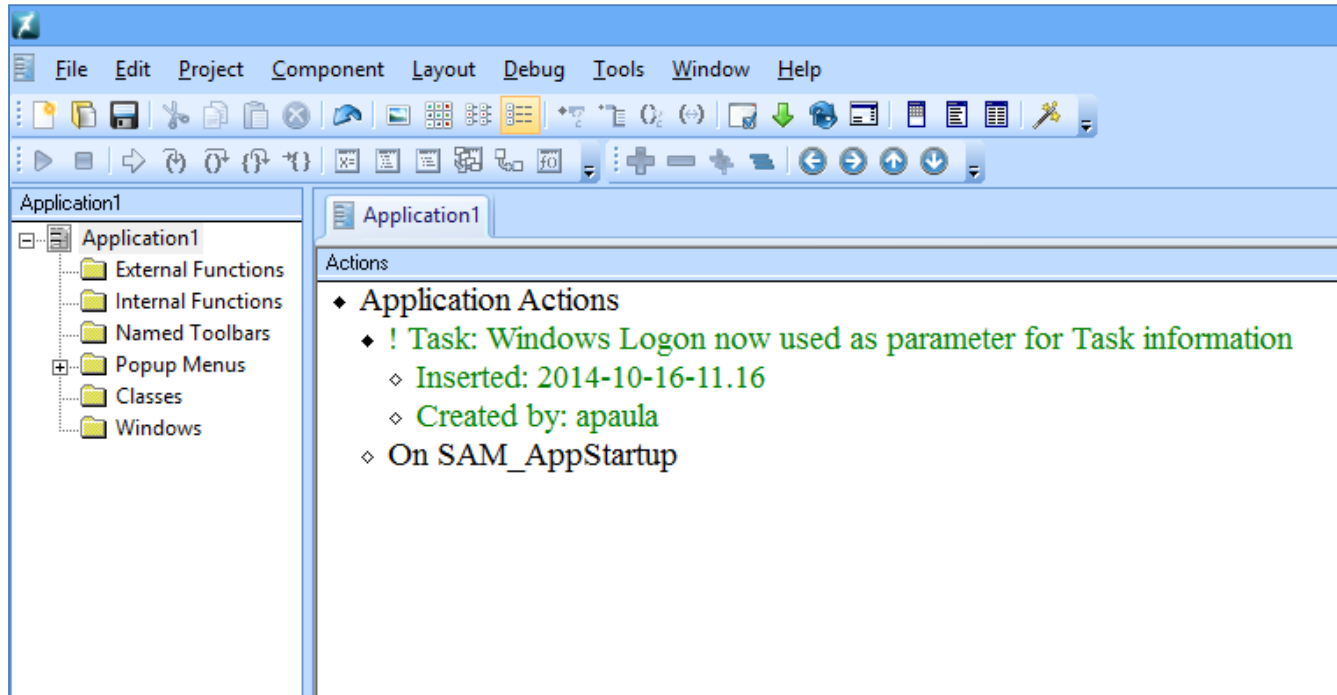
- Mehrere APLs auf einmal einbinden:



Tasks

- **“Insert Task” Funktionen**
 - **Bisher wurde nur das Datum im Task eingetragen**
 - **TD6.3 ermittelt auch den aktuellen Anwender und fügt den Login Namen in den Task ein**
 - **Sinnvoll, wenn mehrere Entwickler an einem Projekt arbeiten**

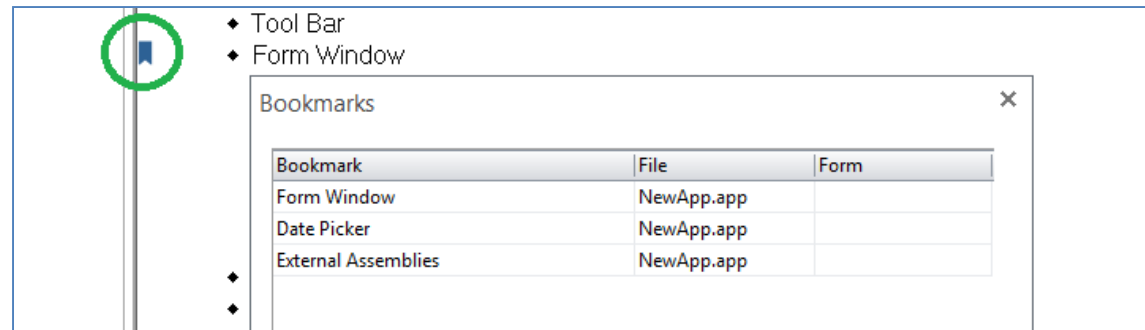
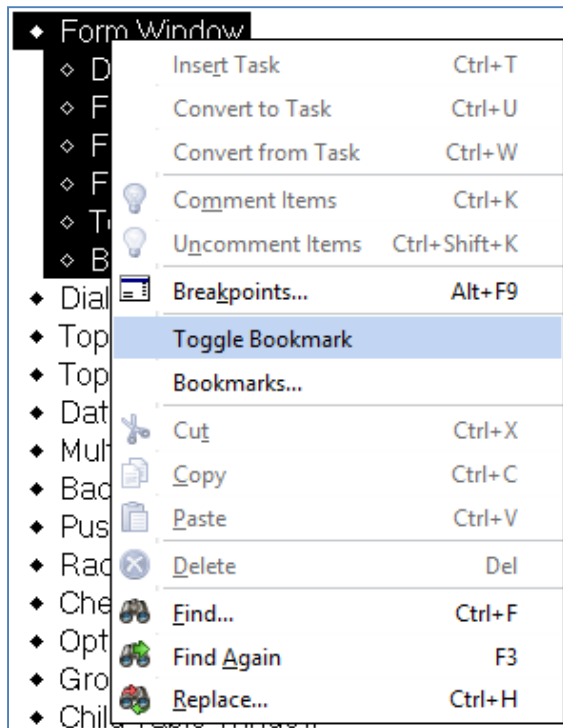
Tasks



Bookmarks

- **Möglichkeit zur Erzeugung von Bookmarks für die schnelle Navigation innerhalb des Sourcecodes**
- **Anwender kann mit “Toggle Bookmark” einen Bookmark ein-/ausschalten**
- **Ein Bookmarks Dialog erlaubt es, die Bookmarks wiederzufinden**
 - **Automatisches Refresh, wenn Bookmarks ein-/ausgeschaltet werden**
 - **Double-click Bookmark, um in den entsprechenden Code zu wechseln**
- **Sortierung mit Hilfe des Spaltentitels**

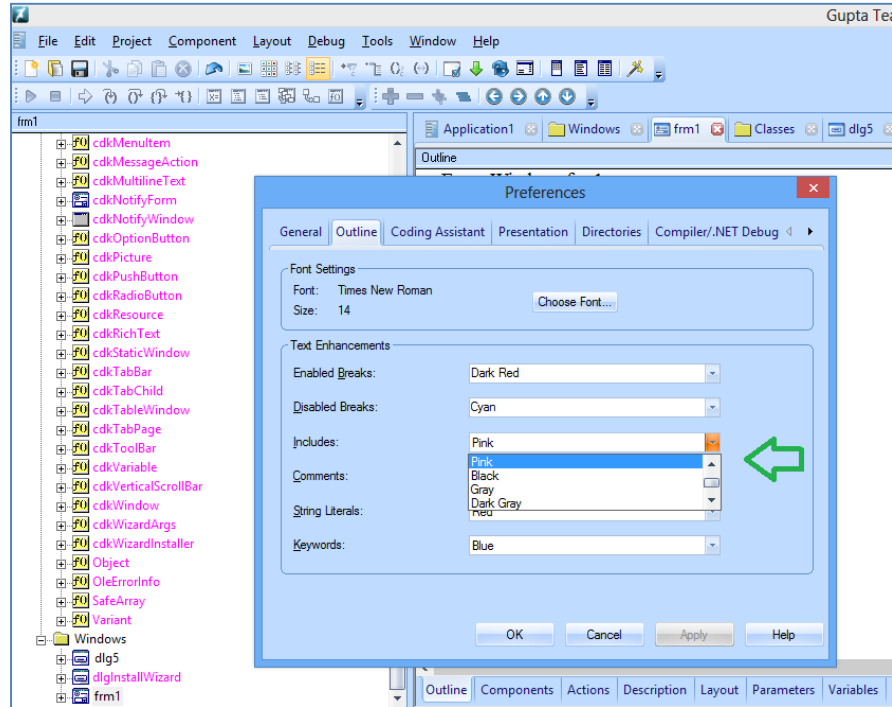
Bookmarks



Colored-Coding

- **Farbige Einträge im Navigationsbaum, wenn ein Node aus einer Bibliothek stammt**
- **Blau ist der Standard, kann aber im Dialog Tools->Preferences->Outline geändert werden**

Color-coded Objekte im Sourcode Baum



Produktivität

- Unterstützung von mehreren “Undo” Änderungen werden schrittweise zurückgenommen
- Unterstützung der “Redo” Funktion

Neues Theme

- Neues, modernes Theme
 - MS Office 2013 look & feel



Team Developer OOP

Team Developer OOP war seiner Zeit weit voraus:

- **Multiple-inheritance**
- **Overriding functions**
- **Late-bound calls**
- **Windows classes**
- **GUI Property Inheritance**

★ Aber: Etwas fehlt! ★
Constructor / Destructor

Hintergrund

Was ist ein Constructor?

Ein Constructor ist ein Block von Code (oder eine Funktion) der automatisch ausgeführt wird, wenn die Instanz eines Objektes erzeugt wird.

- * Application Actions
- * On SAM_AppStartup
 - ◊ Set obj1 = New MyClass()



1. Erzeugen einer Instanz der Klasse "MyClass" im Speicher
2. Aufruf des neuen Konstruktors der Klasse ★
3. Erzeugen der Variablen "obj1" für diese Klasse

Hintergrund

Wozu werden Constructors benötigt?

Constructors sind notwendig, um die Klasse beim Initialisieren vorzubereiten. Zum Beispiel können Werte vorbelegt werden oder Funktionen zur Initialisierung ausgeführt werden.

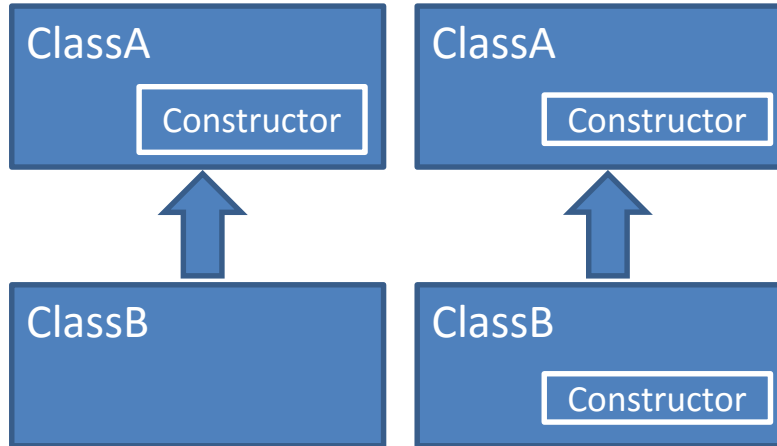
- * Actions
 - ◇ Set nTimeout = 100
 - ◇ Set nProtocol = FTP
 - ◇ Call GetMachineName()

Hintergrund

Constructors und Vererbung

Constructors können von den Eltern vererbt werden und Kinder der Klassen können den Constructor überschreiben:

ClassA's constructor will run when you create an object of ClassB



ClassB's constructor will run when you create an object of ClassB

Hintergrund

Constructors können Parameter haben:

- ◆ Functional Class: clsMyClass
 - ◇ Description:
 - ◇ Derived From
 - ◇ Class Variables
 - ◇ Instance Variables
 - ◆ Constructor/Destructor
 - ◆ Constructor: clsMyClass
 - ◇ Description: Set defaults
 - ◆ Parameters
 - ◇ String: sCurrentUser
 - ◇ Number: nCurrentUserID
 - ◇ Local variables
 - ◇ Actions
 - ◆ Destructor: clsMyClass
 - ◇ Description: Reset defaults
 - ◇ Local variables
 - ◆ Actions
 - ◇ [Set](#) nCurrentUserID = NUMBER_Null
 - ◇ Functions

Hintergrund

Was ist ein Destructor?

- Ähnlich zum Constructor; wird ausgeführt, wenn ein Objekt gelöscht wird
- Wird oftmals zum “Aufräumen” benötigt (Speicher freigeben, Variablen zurücksetzen, ...)
- Es sind keine Parameter erlaubt und es kann nur einen Destructor je Klasse geben

Implementierung

Welche Arten von Klassen können Constructores haben?

- Funktionale Klassen und Windows Klassen
- Aber nicht:
 - Background Text Klassen
 - CoClass Klassen
 - Frame Klassen
 - GroupBox Klassen
 - Line Klassen
 - Separator Klassen
 - Web Service Klassen
 - WPF Custom Klassen

Implementierung

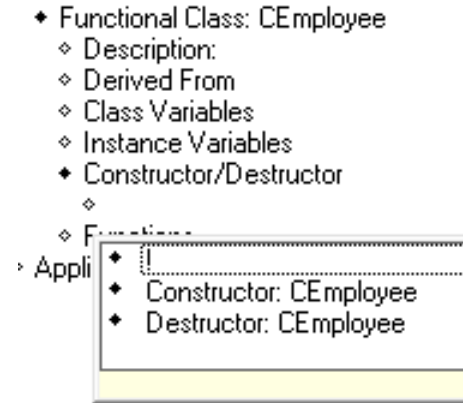
Neuer Node im Outline Baum: “Constructor/Destructor”

- ◆ Class Definitions
 - ◆ Functional Class: CEmployee
 - ◆ Description:
 - ◆ Derived From
 - ◆ Class Variables
 - ◆ Instance Variables
 - ◆ Constructor/Destructor
 - ◆ Functions

Implementierung

Ein Kontext Menü zeigt die Auswahlmöglichkeit an:

- Constructor: <class_name>
- Destructor: <class_name>



Implementierung

Die Auswahl “Constructor” erzeugt einen neuen Constructor Node:

- Der Name des Constructor ist mit dem Klassennamen vorbelegt. Dies ist bei den meisten Sprachen der Standard. Dieser Name kann überschrieben werden:
 - ✦ Constructor/Destructor
 - ✦ Constructor: CEmployee
 - ◇ Description:
 - ◇ Parameters
 - ◇ Local variables
 - ◇ Actions

Implementation

Der Constructor ist eine Art von Funktion der Klasse, somit werden weitere Nodes eingefügt:

- * Constructor/Destructor
 - * Constructor: CEmployee
 - ◇ Description: ← Parameter
 - ◇ Parameters ← Locale Variablen
 - ◇ Local variables ← SAL Code
 - ◇ Actions

Was fehlt:

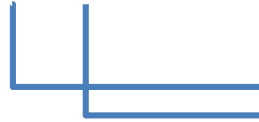
- **Der Constructor hat keinen Return Wert, da er nicht wie eine Methode der Klasse aufgerufen wird**
- **Keine “static” Variablen; diese machen keinen Sinn.**

Implementation

TD's Constructor mit Parameterliste:

- ◇ Parameters
- ◇ Static Variables
- ◇ Local variables
- ◆ Actions

◇ Set objEmployee = New CEmployee("John", 30)



- ◆ Constructor: CEmployee
- ◇ Description:
- ◆ Parameters
 - ◇ String: psArg1
 - ◇ Number: pnArg2
- ◇ Local variables
- ◇ Actions

Destructors

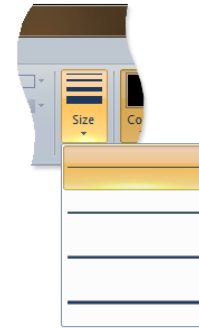
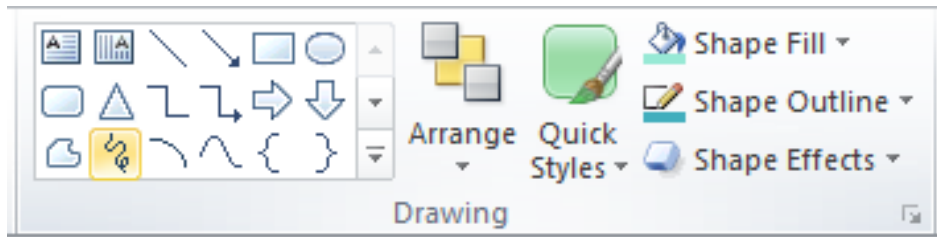
Team Developer's Destructor Implementation

Was sollte nicht bei in einem Desctructor gemacht werden:

- Keine Datenbank Interaktionen
- Keine Exceptions auslösen
- Keine zeitaufwendigen Aktionen
- Keine Logik, die weitere Objekte erzeugt

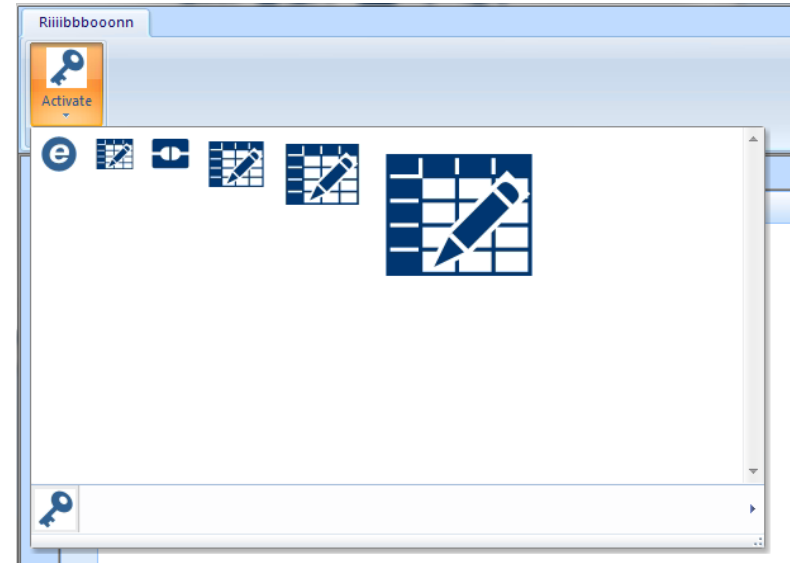
Neues Gallery Control

- **Was ist ein Gallery Control?**
 - **Visuelle Zusammenstellung von Objekten gleicher Art:**



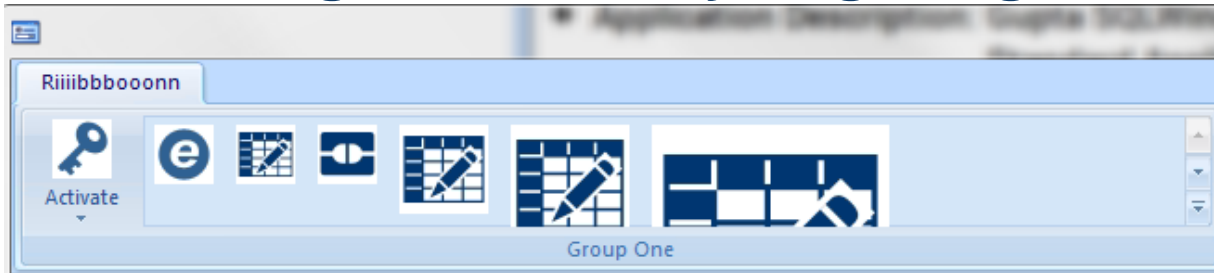
Gallery Control

- **Dropdown Gallery**
 - **Angezeigt als Button**
 - **Wenn es geöffnet wird, dann wird der entsprechende Inhalt angezeigt**
 - **Kann als kleiner oder großer Button angezeigt werden**



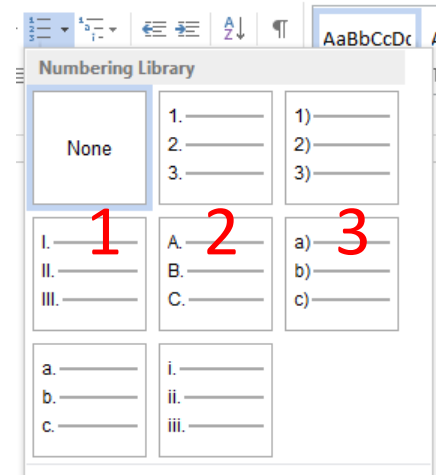
Gallery Control

- **Gallery**
 - Zeigt die erste Reihe einer Gallery im Ribbon Bar Menü
 - Benötigt die gesamte Höhe innerhalb der Ribbon Bar
 - Wenn die Gallery geöffnet wird, dann werden alle Einträge der Gallery angezeigt



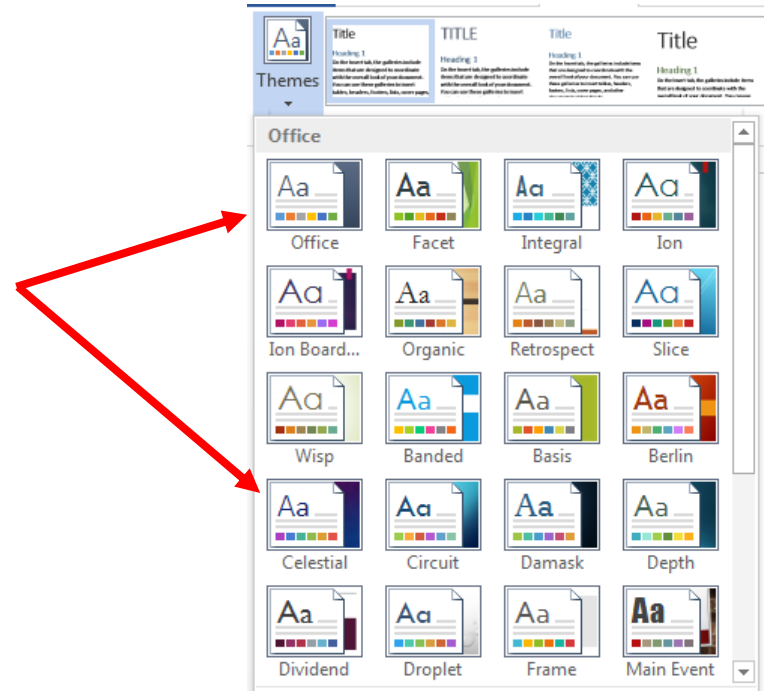
Neues Gallery Control

- **Columns Eigenschaft**
 - Als Standard werden 4 Spalten innerhalb einer Gallery angezeigt
 - Dies kann im Attribute Inspektor überschrieben werden



Neues Gallery Control

- **Gallery items**
 - Der Name jedes Gallery Items kann angezeigt werden:



Neues Gallery Control

- **Gallery Items**

- **Image Größe: Aktuelle Größe des Bitmaps**

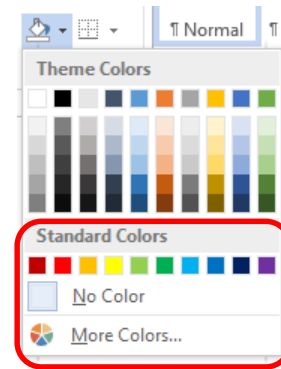
- **Ribbon Actions**

- **Programmcode der ausgeführt werden soll, wenn ein Item ausgewählt wurde**

- Gallery Item: gitemStyle3
- Gallery Item: gitemStyle4
- Ribbon Actions
 - ◊ Set nAction = 4

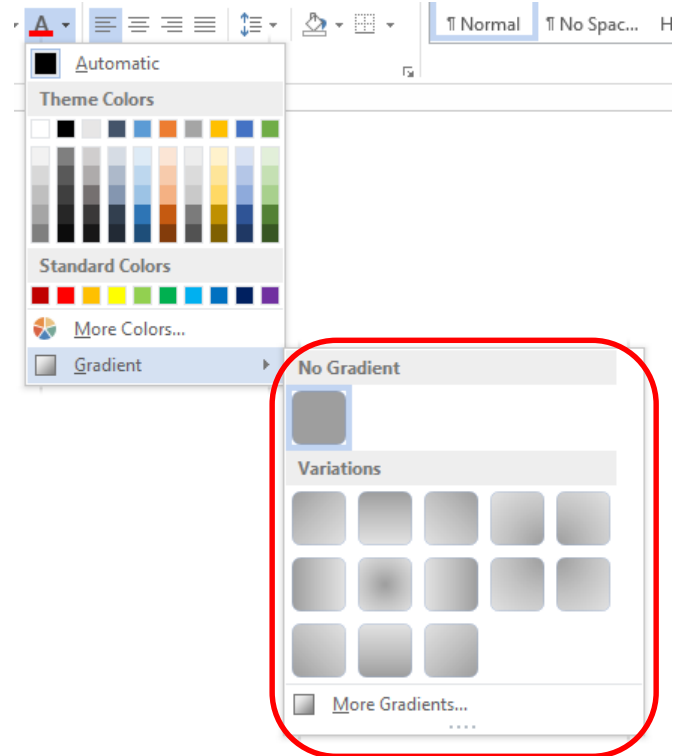
Neues Gallery Control

- **Buttons**
 - **Ein Gallery Control kann optional weitere Buttons beinhalten**



Neues Gallery Control

- Oder weitere Drop Down Gallerys



Neues Gallery Control

- **Schachteln von Gallerys**
 - **Gallerys können nur innerhalb von Ribbon Bar Gruppen definiert werden**
 - **Dropdown Gallerys können innerhalb von Groups, Trays, und anderen Gallerys definiert werden**

Dynamic Ribbon Creation

- **Laufzeit API zum dynamischen erzeugen von Ribbon Bar Objekten:**
 - **SalRibbonAddItem()**
 - Erzeugt ein neues Ribbon Bar Item
 - **SalRibbonDeleteItem()**
 - Löscht ein Ribbon Bar Item

SalRibbonAddItem()

- **Argumente:**
 - **hWndOwner**
 - **String itemName**
 - **String parentItemName**
 - **String itemInsertBefore** – empty “” = am Ende einfügen
 - **Number RibbonItemType**
 - **Bool bDisplayLarge**
 - **String caption**
 - **Template itemResource**

SalRibbonAddItem()

- **Neue Konstanten für Ribbon Bar Item Typen:**

RIBBON_ITEM_TYPE_TAB = 1

RIBBON_ITEM_TYPE_GROUP = 2

RIBBON_ITEM_TYPE_BUTTON = 3

RIBBON_ITEM_TYPE_CHECKBOX = 4

RIBBON_ITEM_TYPE_COMBO = 5

RIBBON_ITEM_TYPE_DATAFIELD = 6

RIBBON_ITEM_TYPE_DROPDOWN = 7

RIBBON_ITEM_TYPE_RADIO = 8

RIBBON_ITEM_TYPE_SEPARATOR = 9

RIBBON_ITEM_TYPE_TRAY = 10

SalRibbonDeleteItem()

- **Parameter:**
 - **hWndOwner**
 - **String itemName** – Name des zu löschenden Items
- **Hinweis: Löscht das Item mit all seinen Abhängigkeiten**

Dynamische Ribbon Erzeugung

- **Wie wird auf ein dynamisch generiertes Item zur Laufzeit reagiert?**
 - **Neu: SAM_RibbonItemClick Message wird ausgelöst, wenn ein dynamisch generiertes Items ausgewählt wird**
 - **Nut für Items die mit SalRibbonAddItem() erzeugt wurden**
 - **IParam – Liefert den Item Namen**
 - **Nutze SalNumberToHString() zum entschlüsseln des Item Namens:**

Set str = SalNumberToHString(IParam)

Ändern von existierenden Ribbons Menüs

- **Ändern von Ribbon Item Größen zur Laufzeit:**
 - **SalRibbonSetItemSize()**
 - **hWndOwner**
 - **String itemName**
 - **Bool bDisplayLarge**

Ändern von existierenden Ribbons Menüs

- **Hide/show Ribbon Items zur Laufzeit:**
 - **SalRibbonSetItemVisible()**
 - **hWndOwner**
 - **String itemName**
 - **Bool bVisible**
 - **Nur für Ribbon Tab, Ribbon Group Und Drop-Down Items**

Ändern von existierenden Ribbons Menüs

- **Ändern und Zuweisen von Images zur Laufzeit:**
 - **SalRibbonSetItemImage()**
 - **hWndOwner**
 - **String itemName**
 - **Template image resource**

SalRibbonPopulate()

- **Füllt ein Ribbon Combo**
 - **hWndOwner**
 - **String itemName – ribbon combo name**
 - **sqlHandle**
 - **String – select statement**

Reagieren auf Ribbon Änderungen

- **Ribbon Bar anzeigen:**
 - **Neue SAM_RibbonShow Message, wenn die Ribbon Bar minimiert, maximiert wird:**
wParam – neuer Status
 - **0 = minimized**
 - **1 = maximized**

Grid Erweiterungen

- **Erweiterte Summen Zeile:**
 - **Sub Totals in Gruppen**
 - Wenn Summary Bar und das Gruppieren eingeschaltet ist, dann werden die Summen in den entsprechenden Gruppen berechnet und angezeigt
- **Neue SAL Funktion, um die entsprechenden Summen der Summenzeile auszulesen**
- **API zum Ausrichten der Label und Werte innerhalb der Summenzeile**

Grid Erweiterungen

- **Neue Eigenschaft einer Zelle: Read only (Yes/No)**
 - SAL Funktion, um eine Zelle auf Read only (Yes/No) zu setzen
- **Anzeigen der PopupEdit Zelle, auch wenn Editable= No gesetzt ist**
 - Z. Z. wird das PopupEdit Fenster nicht geöffnet
- **Möglichkeit zum Zuweisen von Ressourcen mit SalGridSetCellPicture() und SalTreeSetItemImage()**
 - Akzeptiert Dateinamen oder Ressourcen Name als Parameter

Native Menüs

- **Neue native Sal* Funktionen der VT Menü Funktionen und Implementation der fehlenden API**

- VisMenuCheck
- VisMenuDelete
- VisMenuDisable
- VisMenuEnable
- VisMenuGetCount
- VisMenuGetHandle
- VisMenuGetPopupHandle
- VisMenuGetSystemHandle
- VisMenuGetText
- VisMenuInsert
- VisMenuInsertFont
- VisMenuInsertPicture
- VisMenuIsChecked
- VisMenuIsEnabled
- VisMenuUncheck
- VisMenuSetFont
- VisMenuSetPicture
- VisMenuSetText

Native Menüs

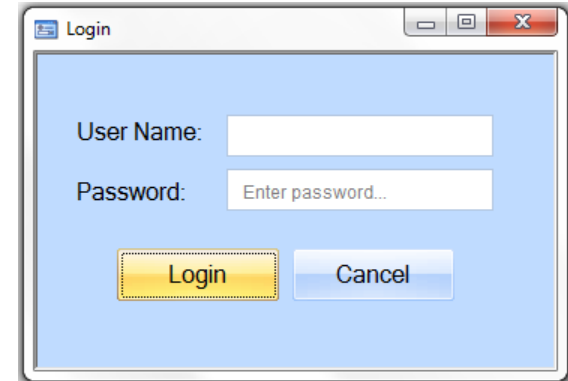
- **Neue Message, wenn ein Menü dynamisch erzeugt/verändert wird**
 - **SAM_MenuChanged**

DateTimePicker

- Möglichkeit zur Anzeige eines beliebigen Datums in Bold
- Mehrfachauswahl von Datumswerten (Array)
- Funktion zum Markieren von Datumswerten (z. B. Feiertage)
- Neue SAL Funktion “Zeige Kalenderwochen“
 - ◆ Check Box: cbWeekNumbers
 - ◆ Message Actions
 - ◆ On SAM_Click
 - ◆ If cbWeekNumbers
 - ◇ Call SalCalShowWeeks(dPicker, TRUE)
 - ◇ Call SalCalShowWeeks(dtPicker, TRUE)
 - ◆ Else
 - ◇ Call SalCalShowWeeks(dPicker, FALSE)
 - ◇ Call SalCalShowWeeks(dtPicker, FALSE)

DataField

- Neu "Hint Text"
- Ein Textfeld kann zusätzlich zum Tooltip einen Hinweistext anzeigen:



Toolbar

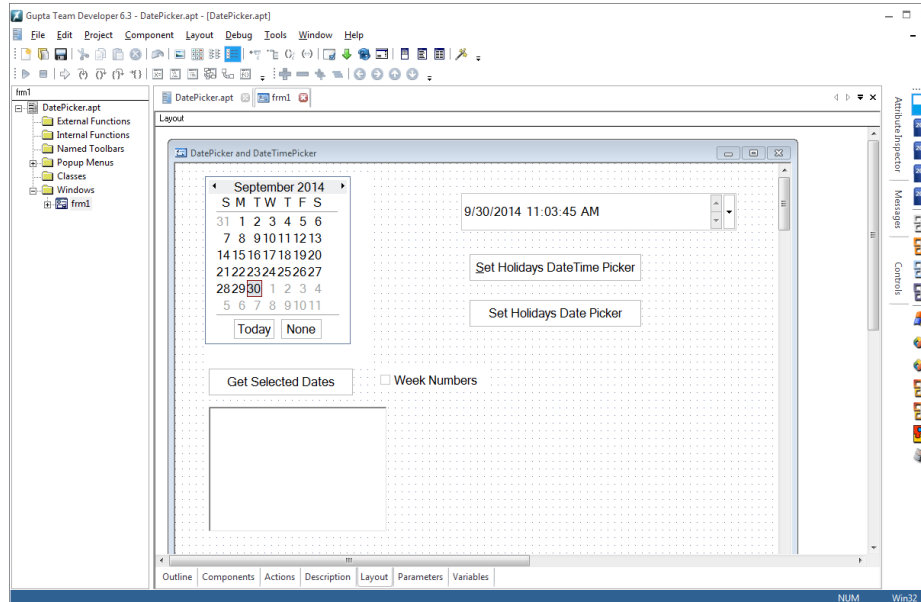
- **Neuer Dialog Toolbar Display Style: “Borderless”**
 - **Zur Zeit implementiert: Standard, Etched, und Shaded**
 - **Neuer Toolbar Style hat keine border/edge Linien**
 - **Kompatibel zu TD V4.2 und früher**

Rich Text Control

- **Word wrapping für Rich Text im Attribute Inspector**
- **Neue SAL Funktionen zum unsichtbar schalten der RTF Control Toolbar Buttons**
 - **SalRTFShowToolBarButton()**

Neues Theme

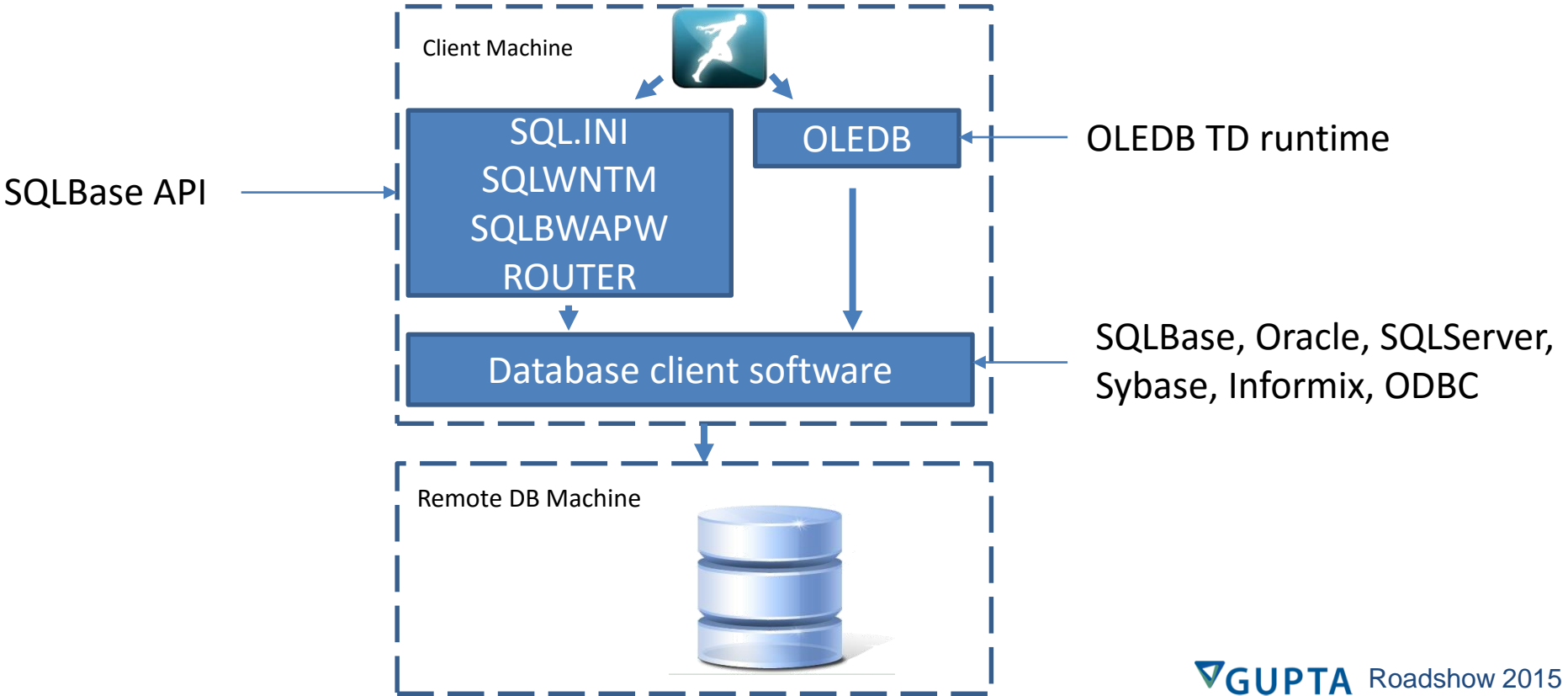
- Modernes “metro-like” Theme für TD



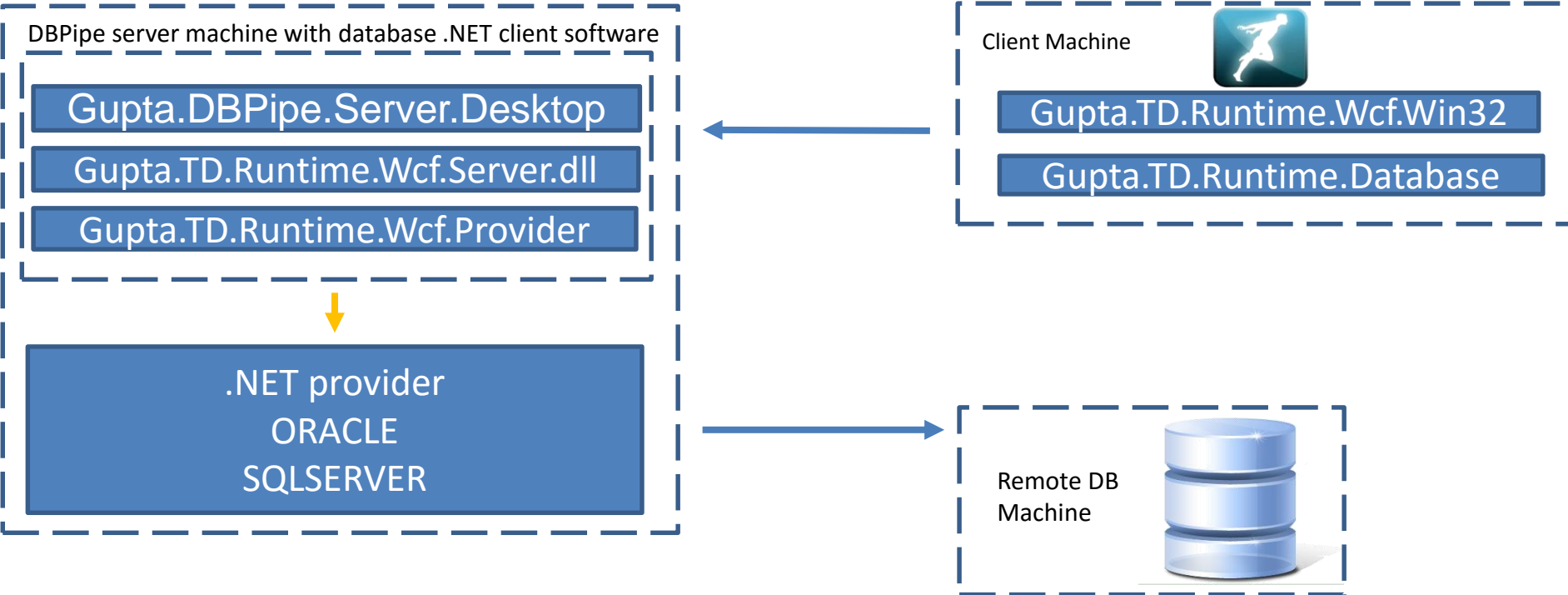
Warum DBPipe für Win32 ?

- **Kundenwunsch!**
- **Abhängigkeiten im Deployment reduzieren**
 - **DBPipe ist unabhängig von Client Routern der unterschiedlichen Datenbanken**

WIN32 DB Connectivity Architektur



DBPipe Architektur



DBPipe WIN32 API

- **bOk = SqlConnectionDbPipe (hSql, sConnectionString, nProviderType)**
 - hSql : Receive Sql Handle. A handle that identifies a database connection
 - sConnectionString - String. A .NET connection string that contains all the information needed to connect to the database, including the username and password.
 - nProviderType - Number. DBP_PROVIDER constant that identifies the database provider you want to use. This should be one of the following values.
 - 0 for ORACLE
 - 1 for SQLSERVER

```
♦ On SAM_AppStartup
♦ When SqlError
♦ Call SqlConnectionDbPipe( hSql, "Data Source=ORCLWIN1252;User ID=support;Password=support", 0)
♦ Set nTestType = 1
♦ If (nArgCount = 2)
♦ If (nTestType = 1)
♦ Else If (nTestType = 2)
♦ Else If (nTestType = 3)
♦ Call SqlPrepareAndExecute(hSql, "COMMIT")
♦ Call SqlDisconnect(hSql)
```

DBPipe Win32 API

- **bOk = SqlDbPipeGetErrorInfo(hSql, nError)**

- Gets the detailed DbPipe error message.
This function is only valid inside a When SqlError or On SAM_SqlError error handler, and the error is 19006 (R00 DBP).
- hSql : Sql Handle. The SQL Handle of the of the cursor that triggered the error.
- nError : Number. Received error number
- sErrorText : String. Received error message

- ◆ Application Actions

- ◆ On SAM_SqlError

- ◆ Call `SqlExtractArgs(wParam, lParam, hSqlError, nError, nPos)`

- ◆ If `nError = 19006`

- ◆ Call `SqlDbPipeGetErrorInfo(hSqlError, nError, sError)`

- ◆ Call `SalMessageBox("SqlError:" || SalNumberToStrX(nError, 0) || ":" || sError, "", 0)`

- ◆ Return `FALSE`

Report Designer TD & TDM

The screenshot displays the Report Designer TD & TDM interface. The main window shows a report layout for 'Invoicegrp_pic'. The layout includes a report header, a page header, a main body with a table, and a footer. The report header features the logo for 'The Island Company' and the word 'INVOICE'. The page header contains the company name and address. The main body contains a table with columns for ITEM NO., DESCRIPTION, QTY., UNIT PRICE, and TOTAL. The footer contains summary totals.

File Preview Language

Report1 Invoicegrp_pic


Report Objects

- Data Items
 - nInvoice_No
 - sCompany_N
 - sInvoice_Date
 - sStatus
 - nItem_No
 - sStyle_Desc
 - nQuantity
 - nItem_Price
 - nTotalLn
- Report Variables
- Formulas
- Totals

Layout

Report Header

Page Header

 **The Island Company** **INVOICE**
Escape | Travel | Live lblStatus

1420 Rocky Ridge Dr. Suite 3
Roseville, CA 95678
P.B.X. +1 (916) 218-47
Fax +1 (916) 218-43

Header: nInvoice_No

lblBillTo

sAddress	
sCityStateZIP	sCountry
sPhone	sFax
sCorporate_URL	

ITEM NO.	DESCRIPTION	QTY.	UNIT PRICE	TOT.
----------	-------------	------	------------	------

Detail Block

nItem_No	nItem_No	nQuantity	nItem_Pri	fLineTotal
----------	----------	-----------	-----------	------------

Footer: nInvoice_No

SUB-TOTAL: fSubTotal
TAX (7%): fTax
GRAND TOTAL: fGrandTotal

Properties



Fragen?