



**opentext**<sup>TM</sup>

# Team Developer 7.1

Was ist Neu in TD7.1?

Source Code Management

Kunden - Themes

WPF Controls

“Nette Kleinigkeiten”

Roadshow April, 2018 | Helmut Reimann

# Agenda



- Team Developer 7.1 Source Code Management
- Anwender LaF
  - Kunden-Themes
  - .NET WPF Objekte in Win32/Win64
  - GUI Erweiterungen
- Produktivität
  - Erweiterungen im Coding-Assistenten (ACA)

# TD 7.1 Source Code und Versions-Verwaltungs-System

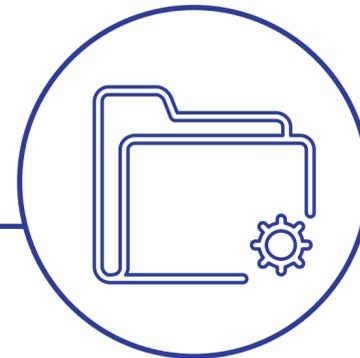
# Unterstützte Version-Control-Systeme



## Git

Access and use Git repositories right from within the Team Developer IDE

Git



## SubVersion

Source code version control using SVN from within the Team Developer IDE

SVN

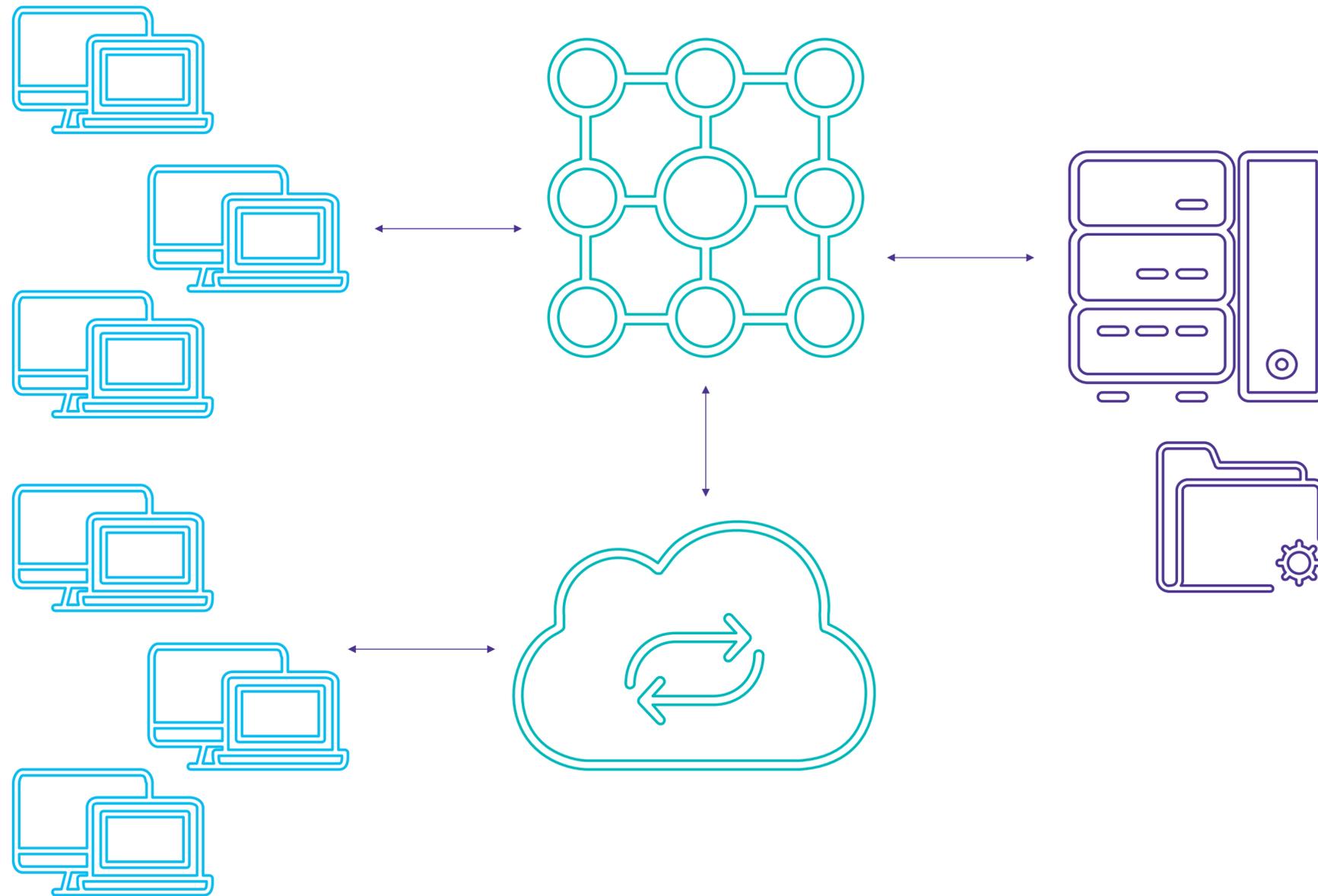


## Team Foundation Server

Use Microsoft Team Foundation Server for your source code version management via the TFS Git interface

TFS

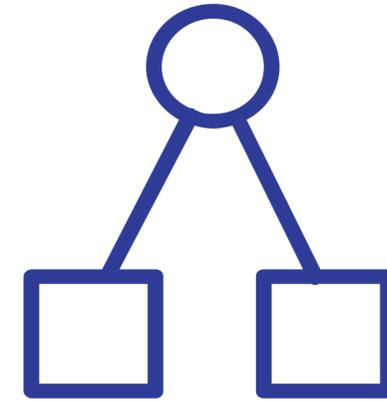
# Konfiguration



- LAN, Intranet und / oder Cloud Zugriff
- Zentraler Git, SVN oder Team Foundation Server
- Der Entwickler kann direkt im TD7.1 auf die entsprechenden Dateien zugreifen

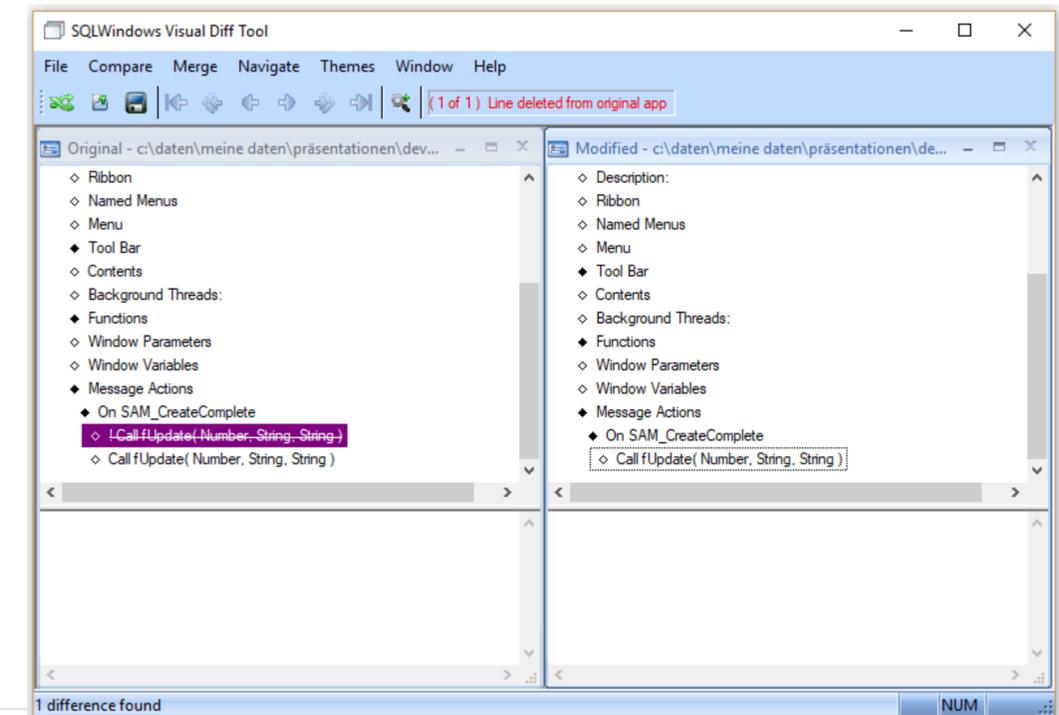
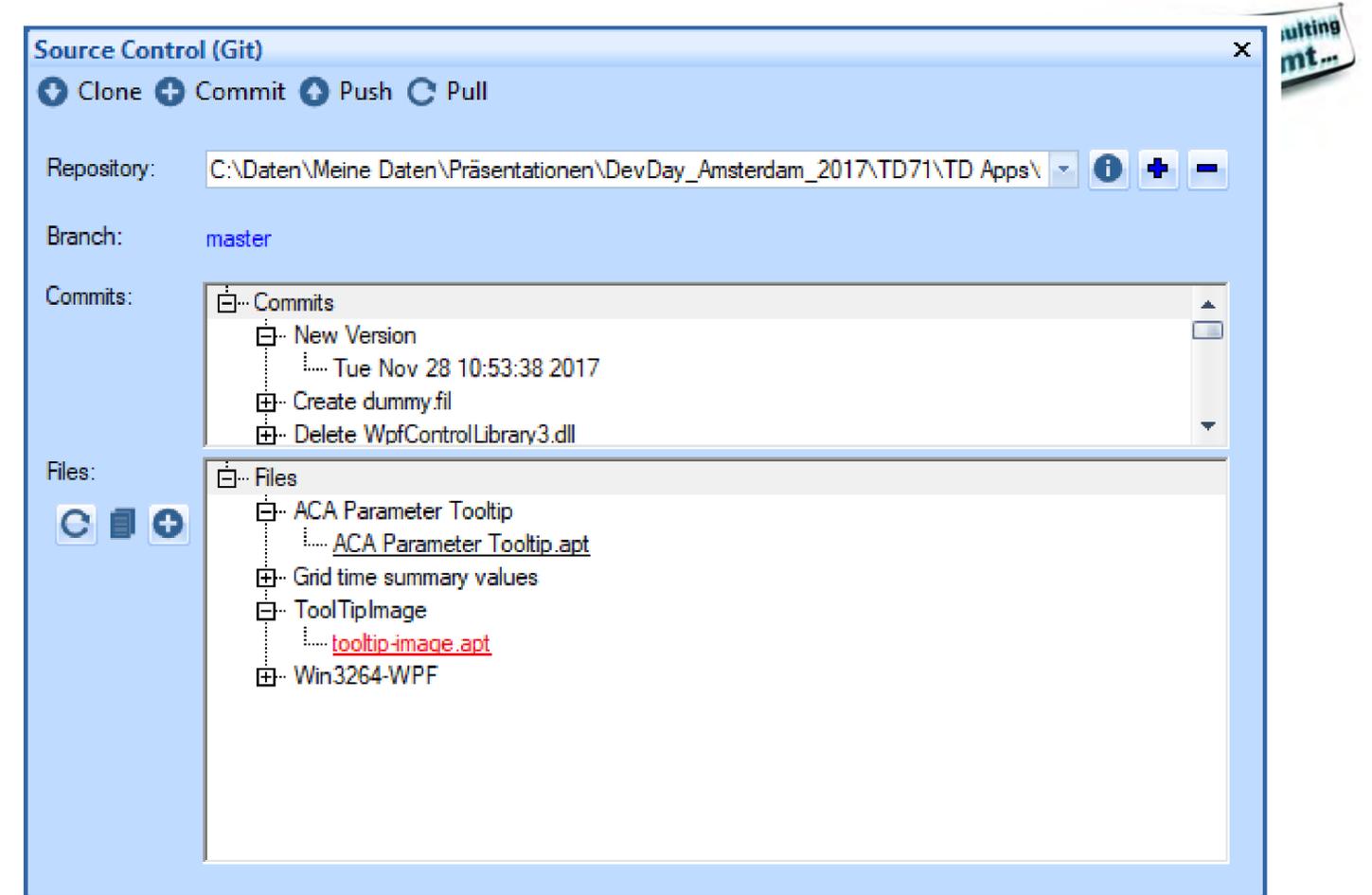
# Git und SVN Versionierung

- Ein Git Repository beinhaltet alle Details zu den Sources auch die verschiedenen Versionen
  - Informationen zu den Versionen werden im Git Fenster als Link angezeigt
- Mit SVN können mehrere “checkouts” getätigt werden, um ‘Trunks’ und ‘Branches’ zu erzeugen
  - Einfache Navigation innerhalb des SVN Fensters



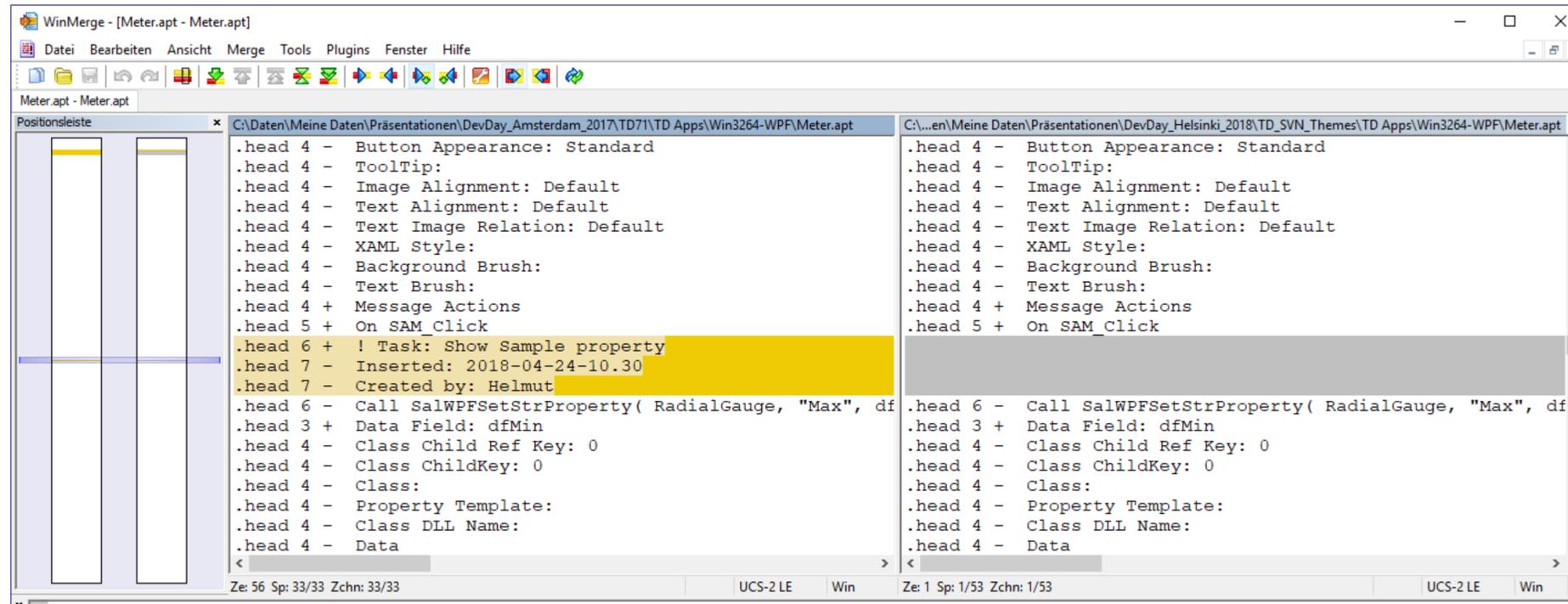
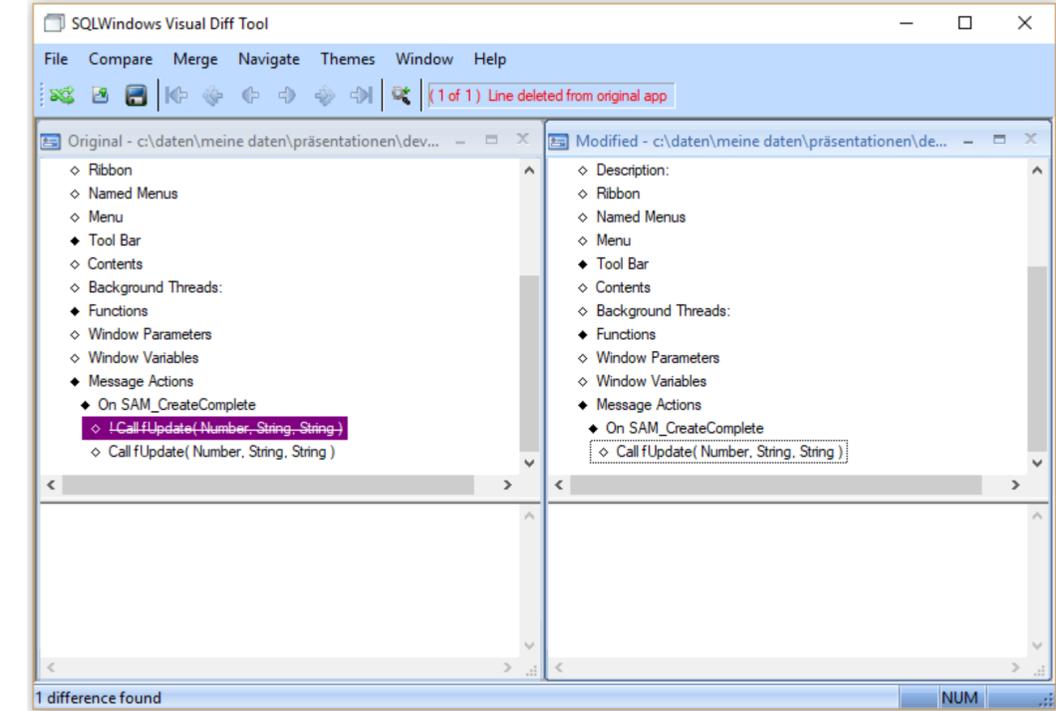
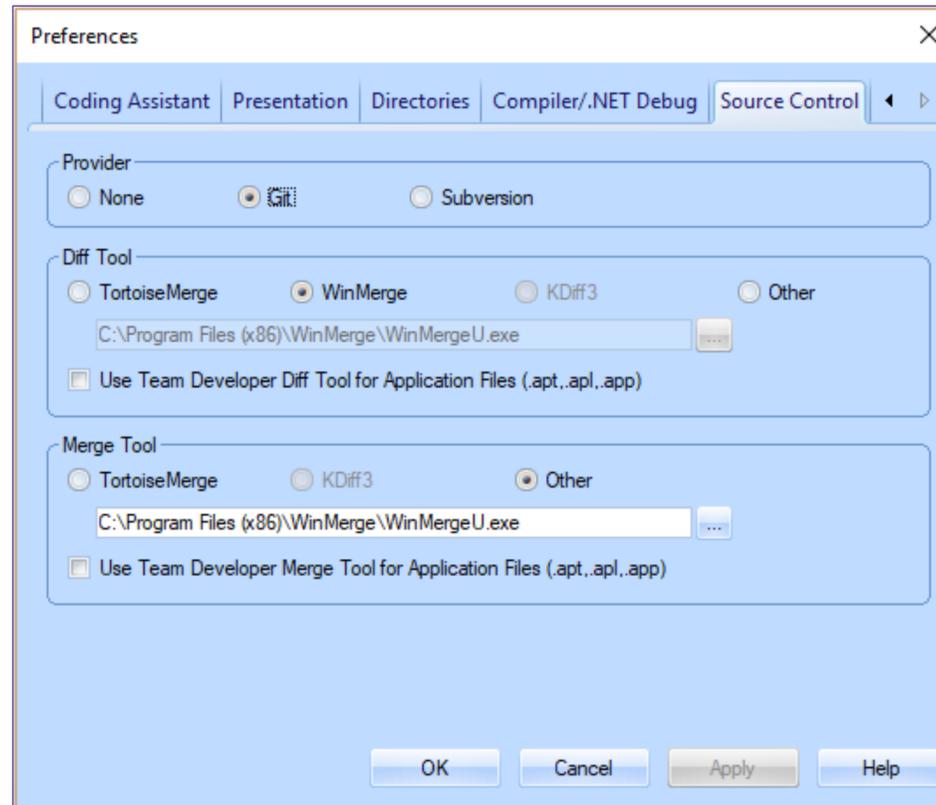
# Überblick Git Integration

- Arbeiten mit mehrfach geklonten Versionen innerhalb des Fensters
- Direktes öffnen der Sourcen innerhalb des Baums
- Ermitteln der Unterschiede zwischen Versionen von Files
- Schreiben 'Commit' der Änderung in das Repository
- 'Hochladen' einer Änderung in das Repository
- 'Download' eines Files aus dem Repository
  - ‚Merged‘ das lokale Repository
- Veränderte Dateien werden farblich angezeigt



# Diff Tools

- In 'Preferences'
- TD Diff Tool
- WinMerge



# Überblick Git Integration



- Beispiel:

HELLE12 / Demo1

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 1 Wiki Insights Settings

No description, website, or topics provided. [Edit](#)

[Add topics](#)

61 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit Message	Time
ACA Parameter Tooltip	Add files via upload	25 days ago
Grid time summary values	Add files via upload	25 days ago
SkinThemeXML	Create dummy.fil	25 days ago
ToolTipImage	Add files via upload	25 days ago
Win3264-WPF	Delete WpfControlLibrary3.dll	25 days ago
README.md	Create README.md	an hour ago

README.md

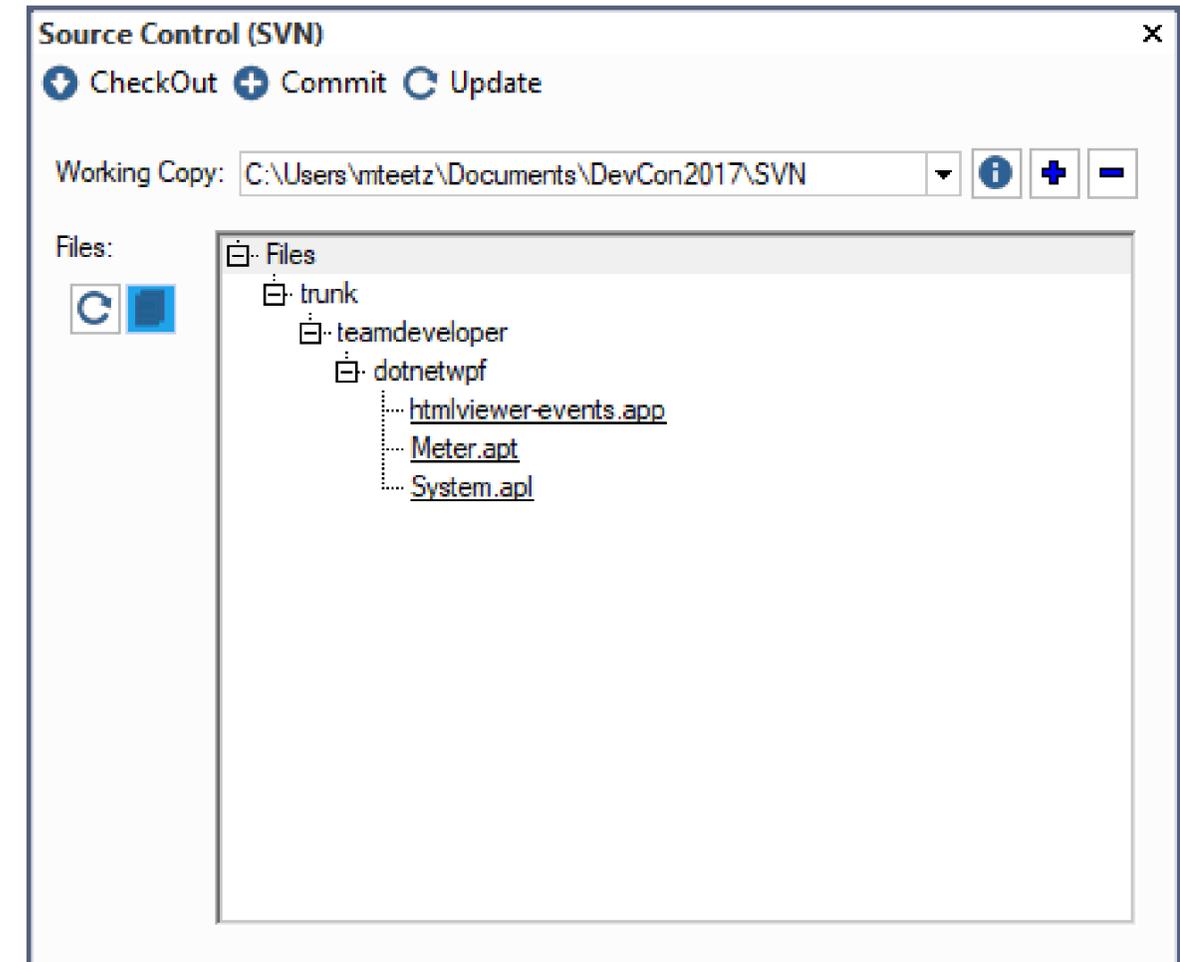
## Demo1

Only for HER Demos

# Überblick SVN Integration

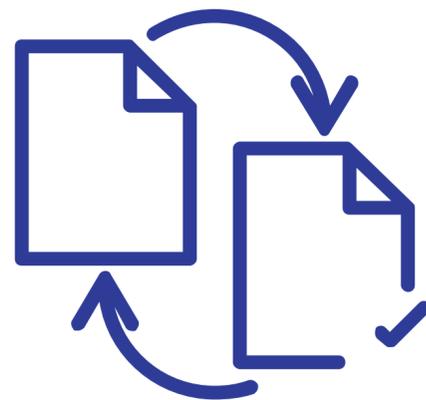


- Es kann mit mehreren Projekten und Versionen gearbeitet werden.
- Auschecken von Projekten innerhalb TD7.1
- Files können direkt im Baus ausgewählt und geöffnet werden
- Ermitteln der Unterschiede zwischen Versionen
- Änderungen werden im zentralen Repository 'Committed'
- Neue Files werden von Repository heruntergeladen
  - Client side merge using TD merge tool for TD code
- Visuelle Anzeige von geänderten Dateien



# Beispiel Projekt

- Beispiel Projekt gibt es auf SourceForge (SVN) und Github (Git)
- <https://svn.code.sf.net/p/opentextgupta-devday-2017/code/>
- <https://github.com/OpenTextGupta/OTGuptaDevDay2017>
- Es muss ein Account erstellt werden; voller Zugriff auf alle Beispiele



# TD 7.1 Erweiterung für das Look & Feel von Anwendungen

# Kunden Theme

- Kunden-Theme Definition
  - XML File
  - Ordner / Filesystem Struktur mit Images der einzelnen Controls
- Neue Funktion zur Pfadangabe des Ordner-Systems
  - Übergabe des Pfades und des Namens des XML File für das Kunden Theme
  - ***SalThemeSetXMLPathName()***
- Kunden Theme landen mit
  - ***SalThemeSet( THEME\_Skin )***

# Kunden Theme XML file

- Beinhaltet color, padding, border, font properties für jeden Control Type oder Control
- Beinhaltet den Image Datei Namen für jeden möglichen Status des Controls

```
<Glyph Name="Background"          DrawMethod="Hollow"  
  Lp ="3"  Tp ="3"  Rp ="3"  Bp ="3"  
  Clp="1"  Ctp="1"  Crp="1"  Cbp="1"
```



Beispiel: PNG ,Templates'

```
<Prof-UIS-Skin Name="OpenText">  
  <SkinItem Name="General">  
    <Color  
      Name="IconAlphaColor"  
      Color="0x706050"  
    />  
  </SkinItem>  
  <SkinItem Name="Label">  
    <Color  
      Name="Normal"  
      Color="0xF0F0F0"  
    />  
    <Color  
      Name="Disabled"  
      Color="0xC0C0C0"  
    />  
  </SkinItem>  
  <SkinItem Name="ComboBox">  
    <SkinItem Name="ComboBoxFrame">  
      <Glyph  
        Name="Normal"  
        Tp="2" Lp="2" Bp="2"  
        Image="OpenText\ComboBox\Fram\Fram.png"  
        DrawCenter="false"  
      />  
      <Glyph  
        Name="Disabled"  
        Tp="2" Lp="2" Bp="2"  
        Image="OpenText\ComboBox\Fram\Fram.png"  
        DrawCenter="false"  
      />  
      <Glyph  
        Name="Hover"  
        Tp="2" Lp="2" Bp="2"  
        Image="OpenText\ComboBox\Fram\Fram.png"  
        DrawCenter="false"  
      />  
      <Glyph  
        Name="Pressed"  
        Tp="2" Lp="2" Bp="2"  
        Image="OpenText\ComboBox\Fram\Fram.png"  
        DrawCenter="false"  
      />  
    </SkinItem>  
  </SkinItem>
```

# Eigenschaften

- Draw Method: Hollow, Stretch, Center, TileH, TileV, Tile
- Padding: Tp, Lp, Bp, Rp
- Content padding: Ctp, Clp, Cbp, Crp
- Border: Lo, To, Ro, Bo

# Kunden Theme Ordner Struktur

- Ein Ordner für jedes Control
- Ordner beinhaltet die Control Images
  - Normal state
  - Default state (has focus)
  - Hover over state
  - Pressed state
  - Disabled state



Beispiel: PNG ‚Templates‘

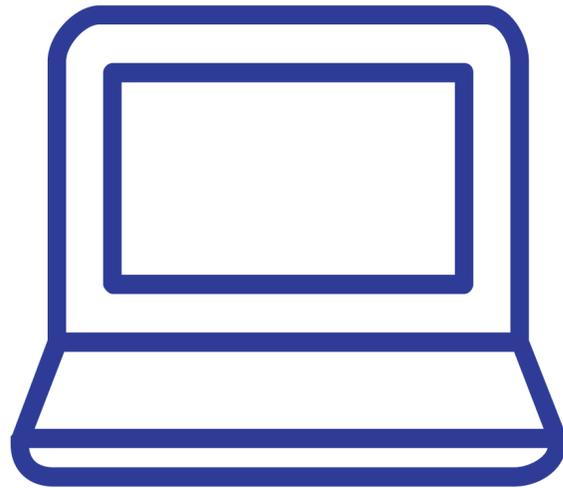
```
+---Background
+---Button
| +---CheckAndRadio
| | +---Check
| | | +---Disabled
| | | +---Hover
| | | +---Normal
| | | \---Pressed
| | \---Radio
| | +---Disabled
| | +---Hover
| | +---Normal
| | \---Pressed
| \---ControlDropDownButton
| | +---DropDownArrow
| | \---PushButton
| \---ControlPushButton
| \---ControlSeparatedDropDownButton
| | +---DropDownArrow
| | +---DropDownButton
| | \---PushButton
| \---FrameNcButton
| | +---Active
| | \---Inactive
| \---GridButton
| | +---Arrow
| | \---Background
| \---MenuBarPushButton
| \---RibbonButton
| | +---Large
| | | +---DropDownButton
| | | +---PushButton
| | | \---SplitButton
| | | | +---DropDownButton
| | | | \---PushButton
| | +---Small
| | | +---DropDownButton
| | | +---PushButton
| | | \---SplitButton
| | | | +---DropDownButton
| | | | \---PushButton
| | \---Tool
| | +---DropDownButton
| | +---PushButton
| | \---SplitButton
| | | +---DropDownButton
| | | \---PushButton
| \---RibbonGallery
```

```
| | +---Horizontal
| | | +---Left
| | | | +---Arrow
| | | | \---Button
| | | | \---Right
| | | | +---Arrow
| | | | \---Button
| | | \---Vertical
| | | +---Down
| | | | +---Arrow
| | | | \---Button
| | | \---Up
| | | +---Arrow
| | | \---Button
| \---ToolBarDropDownButton
| | +---DropDownArrow
| | \---PushButton
| \---ToolBarPushButton
| \---ToolBarSeparatedDropDownButton
| | +---DropDownArrow
| | +---DropDownButton
| | \---PushButton
+---Caption
+---ComboBox
| +---DropDownArrow
| +---DropDownButton
| \---Frame
+---Frame
| \---ControlFrame
| | +---Active
| | \---Inactive
+---Grid
+---ImageCollection
+---MenuItem
+---RibbonGallery
+---ScrollBar
| +---Classic
| | +---Horizontal
| | | \---Vertical
| | +---RibbonGalleryInplace
| | \---Zoom
| | +---Horizontal
| | \---Vertical
+---Tab
| \---Buttons
\---WindowNcArea
```

# Demo



- Kunden Theme



# .NET WPF Controls für Win32/Win64

- Nutzen von .NET WPF Controls in Win32 und Win64 Anwendungen
- Preview .NET WPF Controls im IDE Layout Designer

# WPF Controls



- WPF Controls bestehen aus
  - Properties
  - Methods
  - Events
  - Visual Class Definitionen und Eigenschaften in XAML
    - Text Farben, Hintergrund Farben, Control spezifische Eigenschaften

A screenshot of a window titled "XAML" with a close button in the top right corner. The window contains a text area with the following XAML code:

```
<WebBrowser  
Source="http://www.opentext.com/gupta"  
>
```

At the bottom of the window, there are two buttons: "Cancel" and "OK".

# WPF Control Funktionalität

- Welche Funktionalität hat das WPF Control?
- Beispiel: GoBack und GoForward Methoden der .NET Web Browser Klasse

	GoBack	Navigate back to the previous document, if there is one.
	GoForward	Navigate forward to the next HTML document, if there is one.
	GoForward	Navigate forward to the next HTML document, if there is one.

# WPF Control Events

- Events auf die der Entwickler reagieren möchte
- WPF Controls senden typischer Weise sehr viele Events
- TD kann auf alle diese Events reagieren
- ***SAM\_WPFEvent*** Message erlaubt das Reagieren auf alle Events, die das Control sendet
- SAL Keyword: ***wpfEventName***
  - Beinhaltet den Namen des .Net Events

# Beispiel eines Event Codes

- Beispiel Code um auf ein Event zu reagieren

```

* Journal
* Contents
  * WPF Custom: wpfPlayer
    * Message Actions
      * On SAM_Create
      * On SAM_WPFEvent
        * If wpfEventName = "MediaEnded"
          * Call SalMessageBox( "Playing has completed", "Media Player", MB_Ok )
    * Functions
    * Window Parameters
    * Window Variables
    * Message Actions

* Message Actions
* Window Variables
* Window Parameters
* Functions
  * Call SalMessageBox( "Playing has completed", "Media Player", MB_Ok )
  
```

# WPF Control Eigenschaften setzen / lesen

- Die Eigenschaften eines WPF Controls werden mit folgenden SAL Funktionen gelesen oder gesetzt:
  - *SalWPFGetStrProperty()*, *SalWPFSetStrProperty()*
  - *SalWPFGetNumericProperty()*, *SalWPFSetNumericProperty()*
  - *SalWPFGetDateProperty()*, *SalWPFSetDateProperty()*
  - *SalWPFGetBoolProperty()*, *SalWPFGetBoolProperty()*

# Ausführen von WPF Control Funktionen

- Funktionen der WPF Controls können wie folgt SAL aufgerufen werden:

## ***SaIWPFInvokeMethod( )***

- Window Handle: Name des Controls
- String: Name der aufzurufenden Funktion
- String[\*]: Array mit Argumenten
- Receive String: Zurückgegebener Wert als String

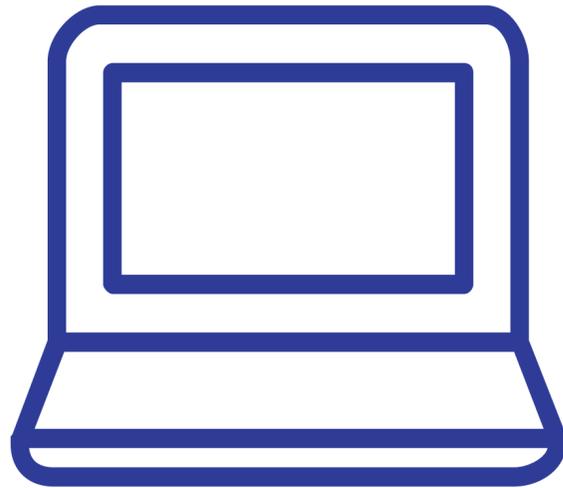
```

* Contents
  * WPF Custom: wpfPlayer
  * Pushbutton: pbPlay
    * Message Actions
      * On SAM_Click
        * Call SaIWPFInvokeMethod( wpfPlayer, "Play", sMethodArgs, sReturn )
  * Functions
  * Window Parameters
  * Window Variables
    * String: sMethodArgs[*]
    * String: sReturn
  * Window Variables
  * Window Parameters
  * Functions
    * Call SaIWPFInvokeMethod( wpfPlayer, "Play", sMethodArgs, sReturn )
  
```

# Demo



- Win32/Win64 WPF Controls



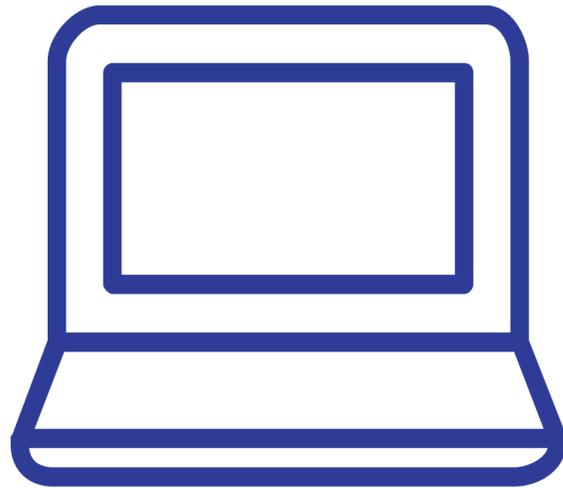
# Grid und Tooltip Erweiterung

- *Grid Group Totals* und *Summary Bar Totals* für Zeit und Datums Werte
  - Neues Event ***SAM\_DateColumnSum***
  - Neue Funktionen
    - BOOL ***SalGridColumnSummaryDateValues( WindowHandle hWndCol, DateTime dtArray )*** erzeugt ein Array mit Datumswerten
    - Number ***SalDateReturn( DateTime dtSum )*** berechnet eine DateTime Summe für eine DateTime Spalte
- Images in Tooltips
  - Resources und Image Files
  - Design Time und Laufzeit
  - ***SalSetToolTipImage()***

# Demo



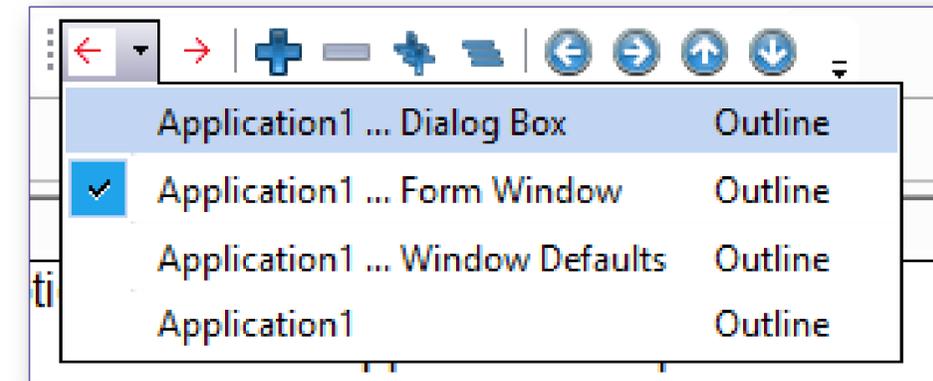
- Grid DateTime Summen
- Tooltip Images



# Programmers Productivity

# IDE Code Navigation

- Navigation zwischen bestehenden Code Segmenten
- Neue Buttons in der Outline Toolbar



# ACA Erweiterungen

- ACA Informationen über Funktion, Parameter, Tooltip am Mouse Cursor; (Wenn der Entwickler die Informationen eingepflegt hat!)
- Wenn keine Informationen zur Funktion verfügbar ist, werden die Datentypen eingeblendet
- *Microhelp* arbeitet auch für externe Funktionen, wenn *Commend-Subnode* entsprechend eingepflegt ist.
- Auch für TD Sal Funktionen

- ◆ Function: fUpdate
  - ◇ Description: This function updates customer records
  - ◆ Returns
    - ◇ Boolean: !whether update was successful
  - ◆ Parameters
    - ◇ Number: nID !Company ID of the row to be updated
    - ◇ String: sCompany !Company name of the row to be updated
    - ◇ String: sAddress !Address of the company to be updated

◇ Call `SqlPrepareAndExecute ( Sql__Handle, String )`

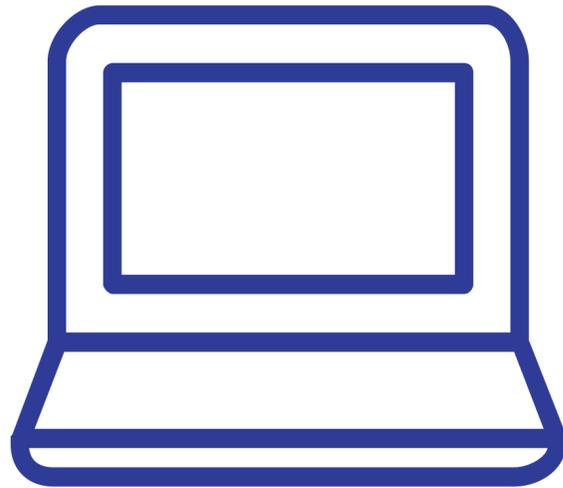
Boolean = `SqlPrepareAndExecute ( Sql__Handle: hSql, String: strSqlStatement )`  
Compiles and executes a SQL statement.  
hSql - Sql Handle. A handle that identifies a database connection.  
strSqlStatement - String. The SQL statement to compile and execute.

nID - Company ID of the row to be updated  
sCompany - Company name of the row to be updated  
sAddress - Address of the company to be updated

# Demo

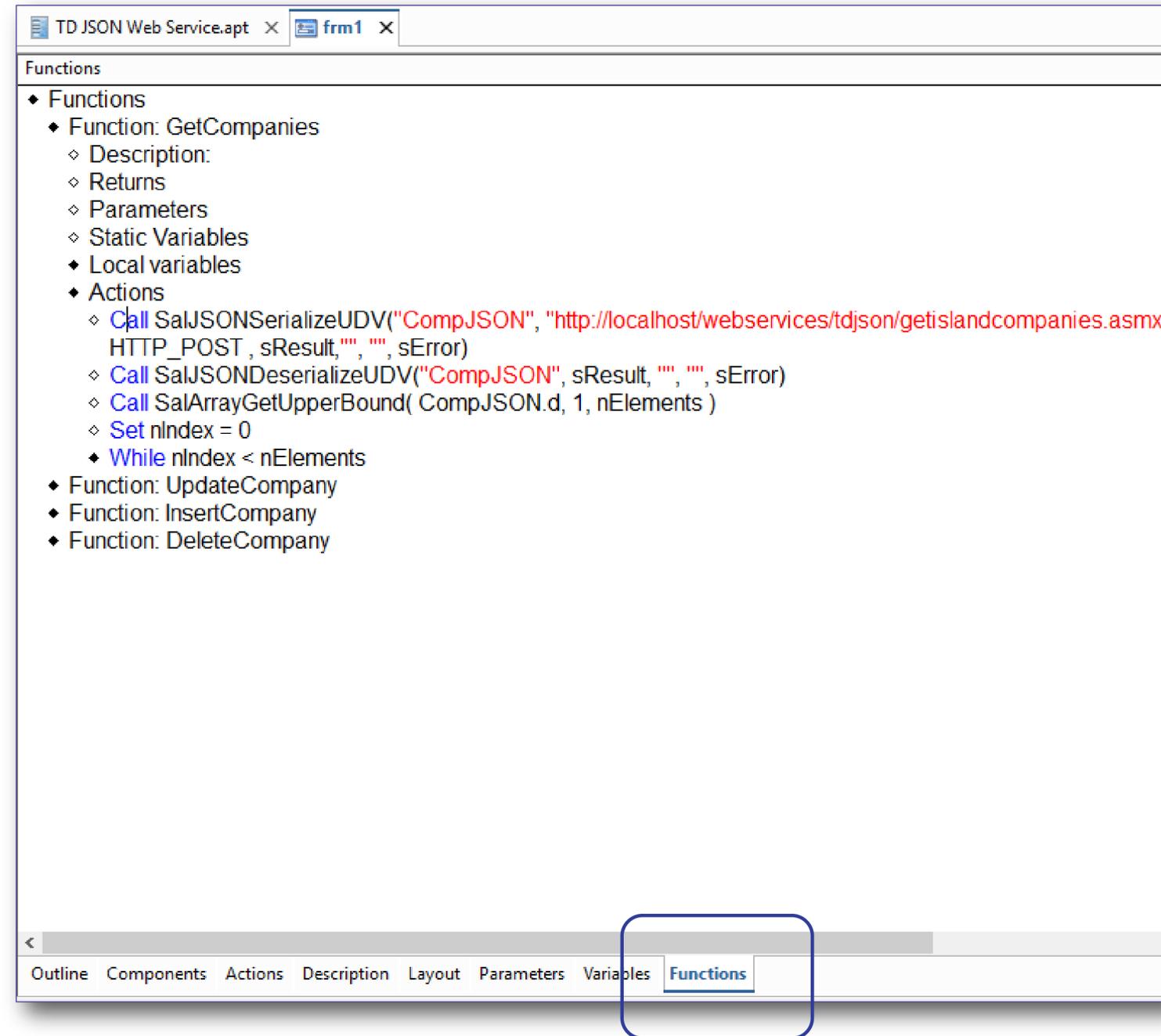


- ACA Erweiterungen



# Neue Window Functions Tab

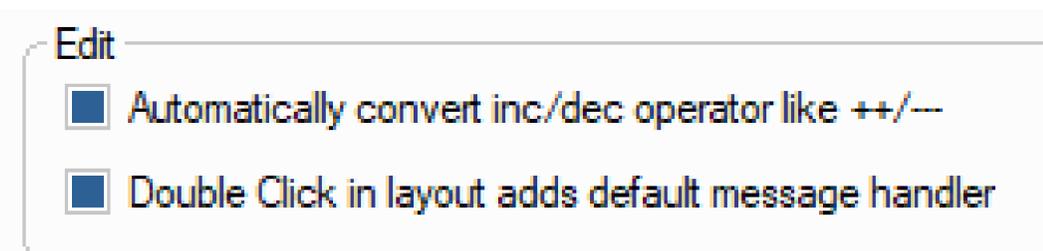
- Schneller Zugriff auf die Funktionen eines Windows



# Zur Info

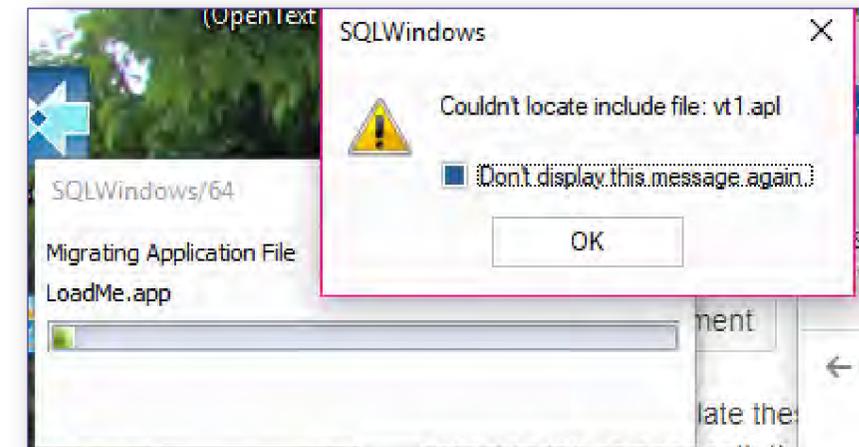


- Informix 64-bit Driver
- Child Objects können nun mit mehreren Tab Pages verknüpft sein
- 'Toggle Bookmark' Menüeintrag nun im 'Edit Menü' der IDE
- Double-Click auf ein Control im Design Mode fügt die 'Default Message' ein
  - Das Verhalten kann im 'Preferences-Dialog' eingestellt werden.



# Last Nice Little Things

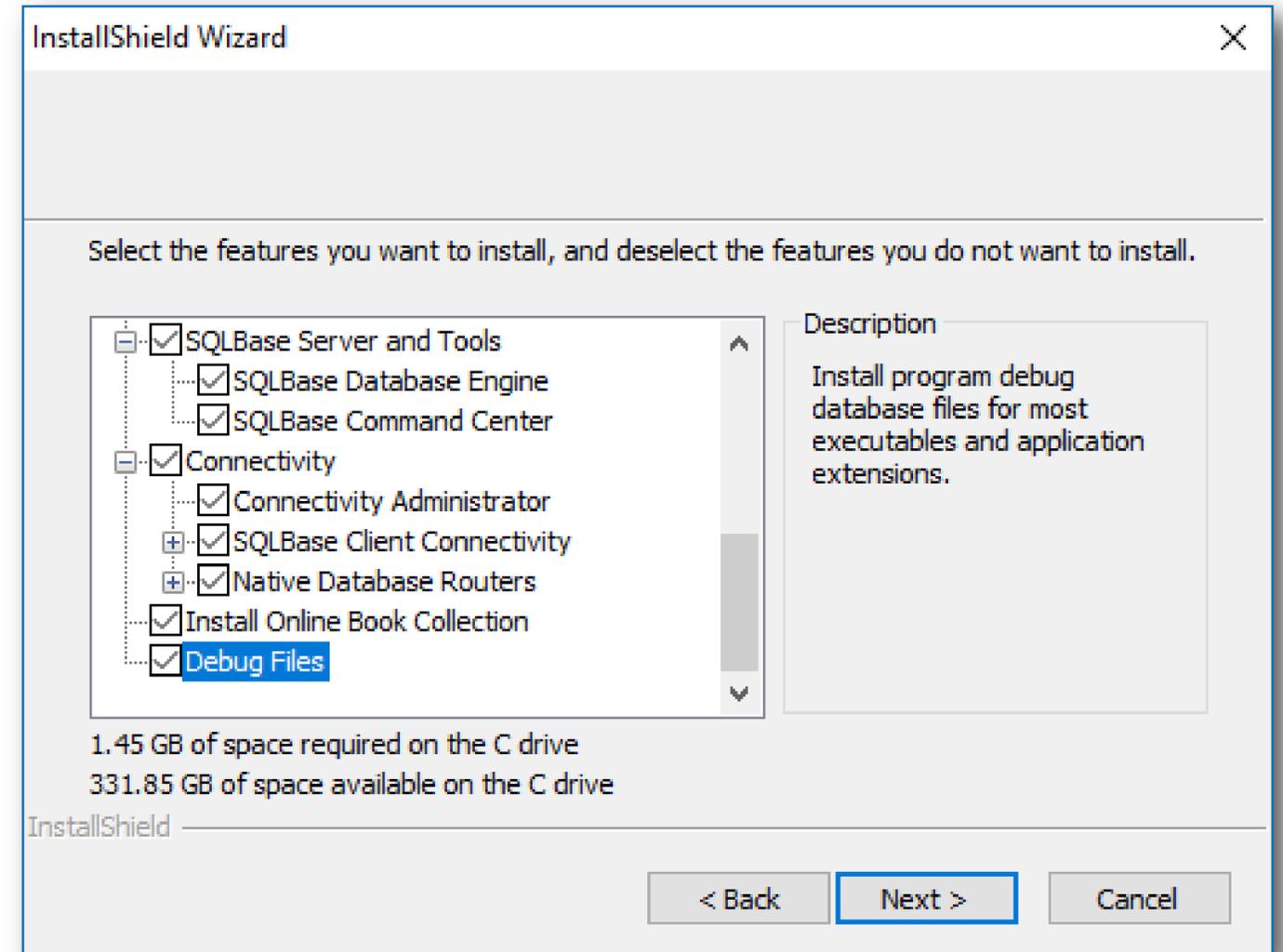
- Neue Funktion ***SalCenterWindowEx()***
  - Für Windows: setzt ‚resizeable=no‘
- ***SalPicSet()*** und ***SalPicSetFile()*** kann genutzt werden, um ein MDI Hintergrund-Bild darzustellen.
- ***SalPicSetFit()*** unterstützt ‘Scale’ und ‘Size’ der Bilder
- Fehlermeldungen beim Laden von Sourcecode Files können abgestellt werden:
  - Z.B. Fehlende Icons, fehlende APLs



# Last Nice Little Things



- Team Developer – Unterstützung beim ‘Issue-Tracking’
- Der TD Installer kann nun auch die \*.DBS Files installieren
  - Möglicherweise benötigt der OT Customer Support und das Engineering Informationen aus den \*.DBS files um das Problem einzugrenzen.
  - Sinnvoll bei Problemen die nur bei bestimmten Kunden, Umgebungen, ... auftreten.



**opentext**<sup>TM</sup>

Danke!



[twitter.com/ot\\_gupta](https://twitter.com/ot_gupta)



[facebook.com/opentextgupta](https://facebook.com/opentextgupta)



[linkedin.com/company/opentextgupta](https://linkedin.com/company/opentextgupta)

**[opentext.com/gupta](https://opentext.com/gupta)**