



opentextTM

Team Developer 7.2

Neue Funktionen im Team Developer 7.2

Roadshow Mai 2019 | Helmut Reimann

Agenda

- Neues Farb-Schema „Office 2016“
- Profiling Support
- Windows Nachrichten
- Background Threads – Datenübergabe „Best Practice“
- „REST-Importer“
- Report Designer in TD7.2
- Sonstige Neuerungen
 - Neue APIs

Neues Farb-Schema „Office 2016“

THEME_Office2016_R2_Metro

SB12 - TD71

Manuell Starten Gesamtzeit **0,0640** Sekunden Dashboard Beenden

Combo 1 Acapulco Ropa Tropical

Gl. Array Anne's Attire **0,0320** Sekunden

Combo 2 Lanai Lingerie

Invoice No	Comp ID	Company Name	Paid
40	120	Acapulco Ropa Tropical	1.012,25
45	120	Acapulco Ropa Tropical	0,00
41	119	Amazon Apparel	1.225,00
42	119	Amazon Apparel	0,00
38	110	Anne's Attire	0,00
39	110	Anne's Attire	0,00
43	110	Anne's Attire	0,00
44	110	Anne's Attire	0,00
46	118	Australia Clothing Outlet	0,00
47	111	Betsy Ross American Fashions	0,00
54	111	Betsy Ross American Fashions	0,00
6	101	Clothing Connoisseurs	1.536,75
Count	54	Total	47.786,95

Thread Liste:

- Combo Company Start
- Combo Product Start
- Fill Grid Start
- Combo Company Array Start
- Combo Company Array Filled
- Fill Grid Finished
- Combo Products Filled
- Combo Company Filled

Profile An / Aus

Einzelzeiten

Combo 1: **0,1080** Sekunden

Combo 2: **0,0840** Sekunden

Grid: **0,0600** Sekunden

Summe: **0,2520** Sekunden

Reduzierte

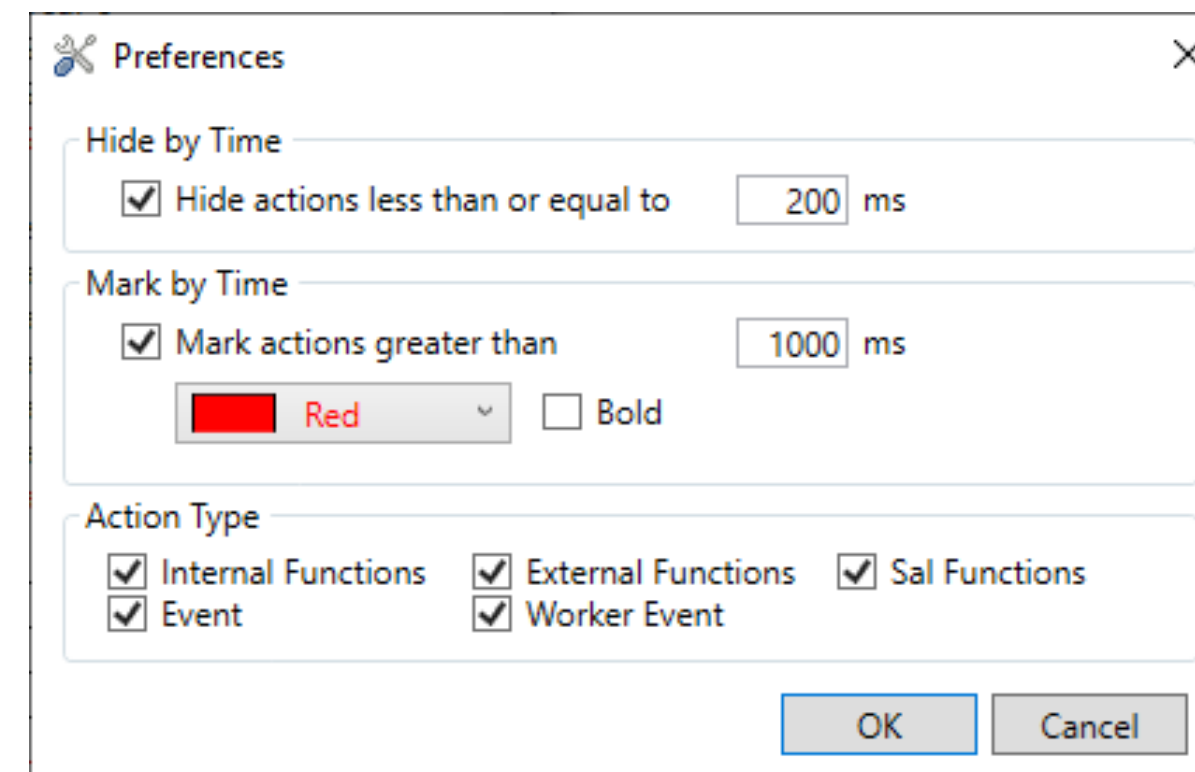
Wartezeit **-0,1880** Sekunden

Profiling Support

- Analyse zur „Laufzeit“ von Anwendungs-Code
 - „Wo ist Optimierungsbedarf? Machen Background-Threads Sinn?“
- SAL – API:
 - **SalProfileStart(„*FileName.trc*“, nLevel)**
 - nLevel: 1 = ‚ON‘ Messages und Interne Funktionen; 2 = Plus SAL Funktionen
 - **SalProfileStop()**
 - **SalProfileSuspend()**
 - **SalProfileContinue()**

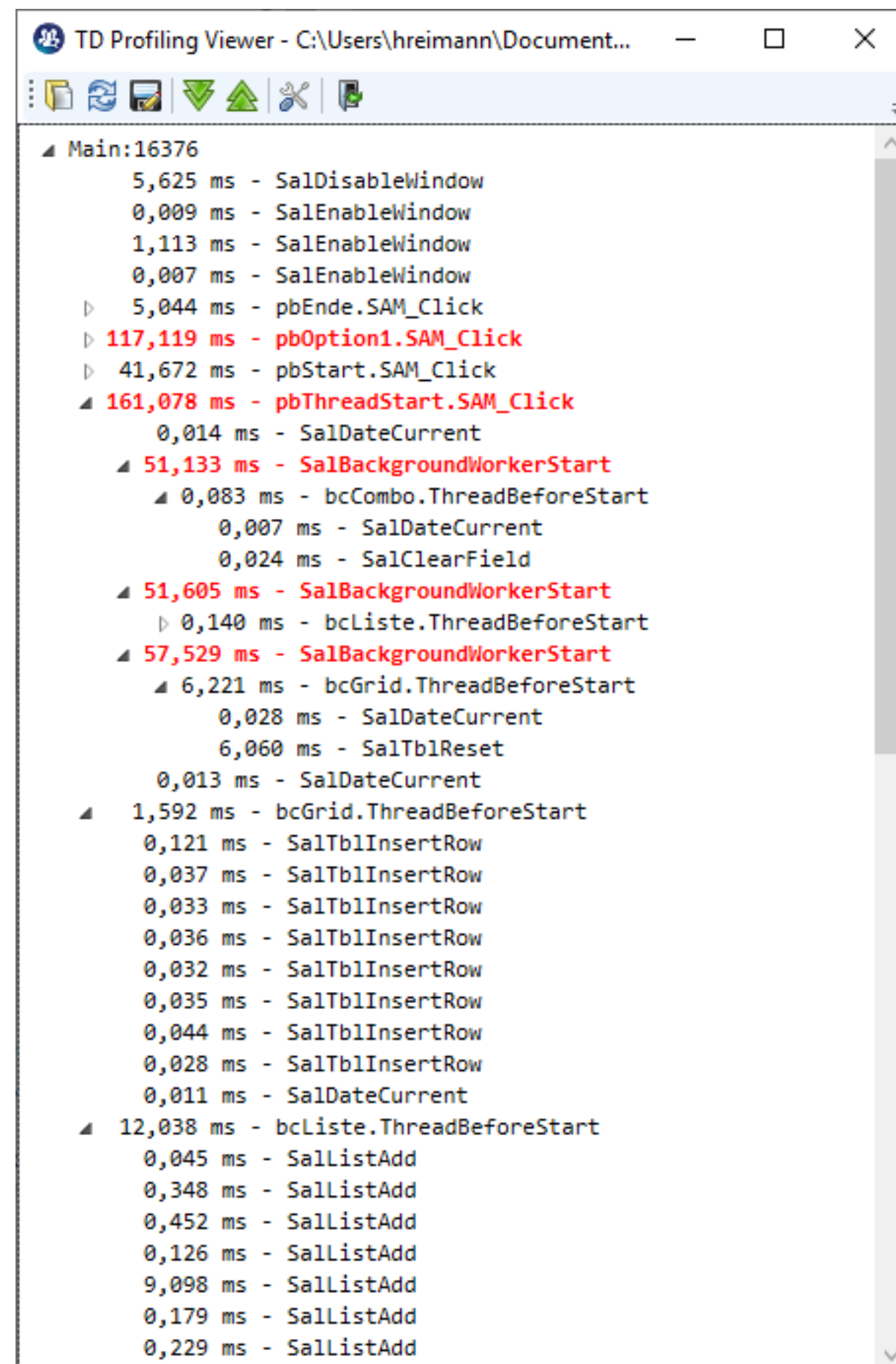
Profiling Support

- Informationen zum Programmablauf
 - Messages
 - Externe Funktionen
 - Informationen zu benötigten Ausführungszeiten einzelner Funktionen
 - SAL – API
 - Umfangreiche Auswertung:

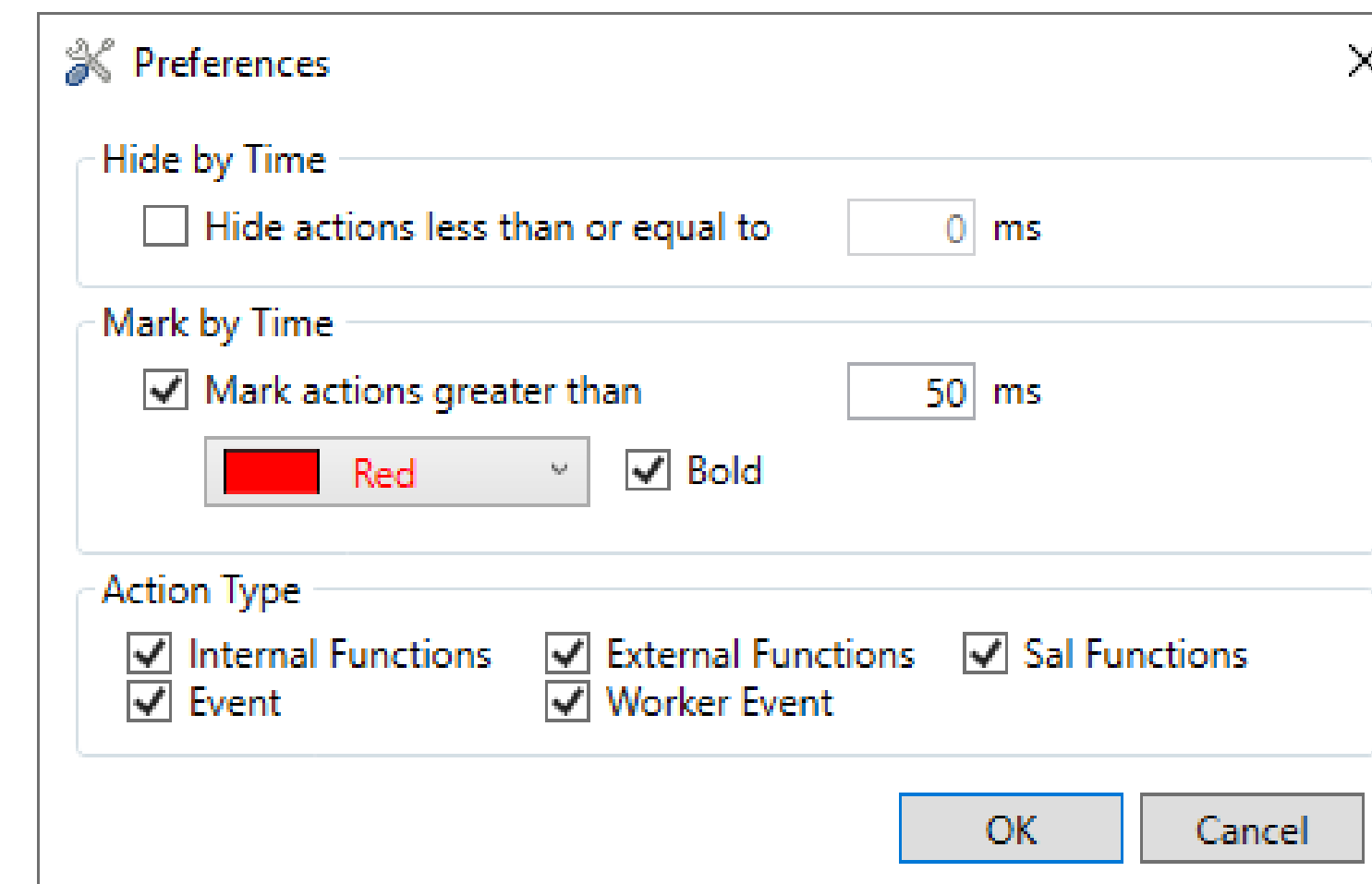


Profiling Support

- Informationen zum Programmablauf:



```
TD Profiling Viewer - C:\Users\hreimann\Document...
Main:16376
  5,625 ms - SalDisableWindow
  0,009 ms - SalEnableWindow
  1,113 ms - SalEnableWindow
  0,007 ms - SalEnableWindow
  5,044 ms - pbEnde.SAM_Click
  117,119 ms - pbOption1.SAM_Click
  41,672 ms - pbStart.SAM_Click
  161,078 ms - pbThreadStart.SAM_Click
    0,014 ms - SalDateCurrent
    51,133 ms - SalBackgroundWorkerStart
      0,083 ms - bcCombo.ThreadBeforeStart
        0,007 ms - SalDateCurrent
        0,024 ms - SalClearField
      51,605 ms - SalBackgroundWorkerStart
        0,140 ms - bcListe.ThreadBeforeStart
      57,529 ms - SalBackgroundWorkerStart
        6,221 ms - bcGrid.ThreadBeforeStart
          0,028 ms - SalDateCurrent
          6,060 ms - SalTblReset
          0,013 ms - SalDateCurrent
        1,592 ms - bcGrid.ThreadBeforeStart
          0,121 ms - SalTblInsertRow
          0,037 ms - SalTblInsertRow
          0,033 ms - SalTblInsertRow
          0,036 ms - SalTblInsertRow
          0,032 ms - SalTblInsertRow
          0,035 ms - SalTblInsertRow
          0,044 ms - SalTblInsertRow
          0,028 ms - SalTblInsertRow
          0,011 ms - SalDateCurrent
        12,038 ms - bcListe.ThreadBeforeStart
          0,045 ms - SallistAdd
          0,348 ms - SallistAdd
          0,452 ms - SallistAdd
          0,126 ms - SallistAdd
          9,098 ms - SallistAdd
          0,179 ms - SallistAdd
          0,229 ms - SallistAdd
```



Preferences

Hide by Time

Hide actions less than or equal to ms

Mark by Time

Mark actions greater than ms

Red Bold

Action Type

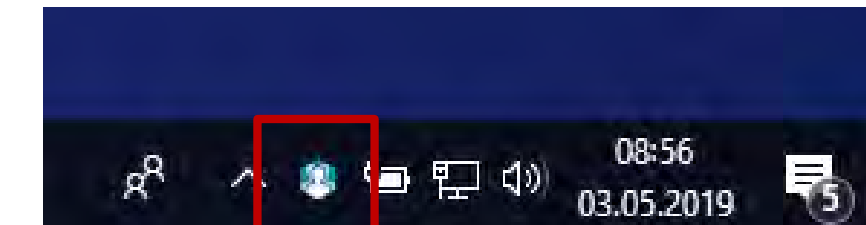
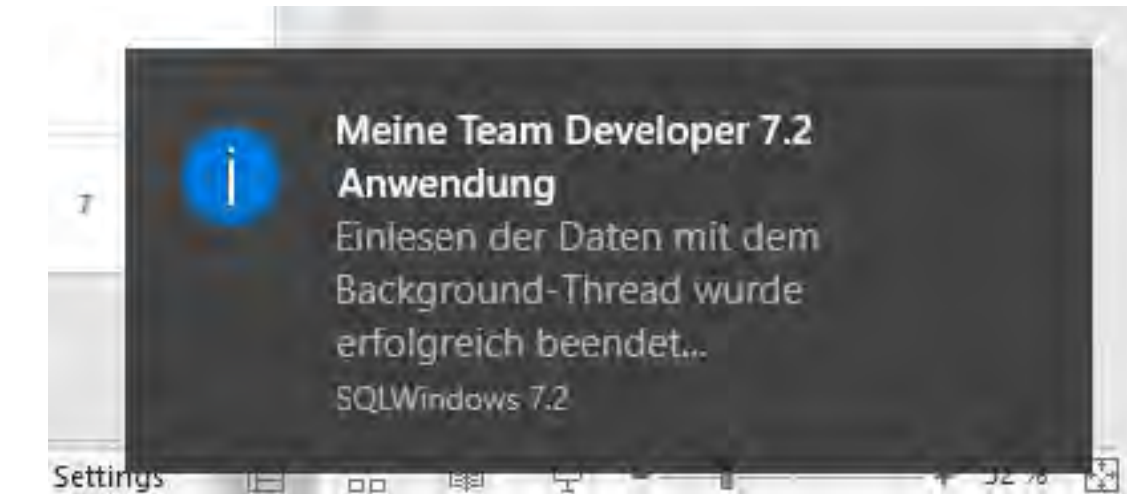
Internal Functions External Functions Sal Functions

Event Worker Event

OK Cancel

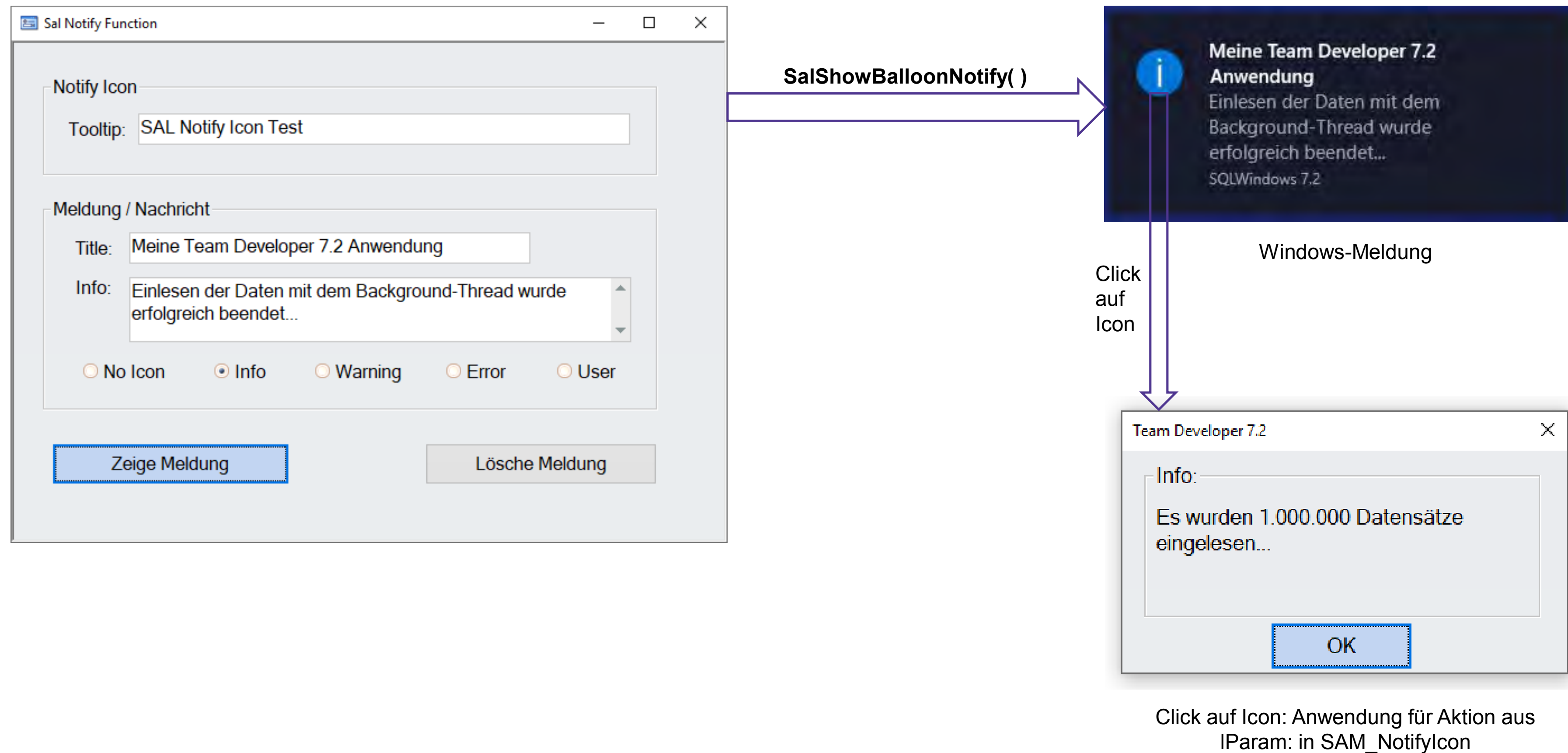
Windows Nachrichten „Notifications“

- Windows Benachrichtigungen
- SAL – API
 - **SalShowNotifyIcon(hWndForm, 'iconGupta', dfTooltip)**
 - **SalShowBalloonNotify(hWndForm, nBalloonType, iconBalloon, dfTitle, mlInfo)**
 - **SalShowNotify(hWndForm, 'iconGupta', dfTooltip, nBalloonType, iconBalloon, dfTitle, mlInfo)**
 - **SalDeleteNotify()**
- SAM Message: **SAM_NotifyIcon**



Windows Nachrichten „Notifications“

- Beispiel:



Windows Nachrichten „Notifications“

- Auslösen der Meldung:

- ◆ Pushbutton: pbShowBalloon
 - ◆ Message Actions
 - ◆ On SAM_Click
 - ◇ Call SalShowBalloonNotify(frm1, nBalloonType, iconBalloon, dfTitle, mlInfo)

- Klicken eines Icons:

- ◆ Message Actions
 - ◆ On SAM_NotifyIcon
 - ◆ Select Case IParam
 - ◆ Case WM_RBUTTONDOWN
 - ◇ Call SalTrackPopupMenu(hWndForm, 'popup1', TPM_CursorX | TPM_CursorY, 0, 0)
 - ◆ Case WM_LBUTTONDOWN
 - ◇ Call SalModalDialog (dlgZeigeInfo, hWndForm)
 - ◆ Case 1029
 - ◇ Call SalModalDialog (dlgInfo, hWndForm)
 - ◆ On SAM_Create
 - ◇ Call SalCenterWindow(frm1)
 - ◇ Call SalDeleteNotify()

Team Developer Background Threads

- Zur Erinnerung....

SB12 - TD71
Manuell Starten
Gesamtzeit 0,0720 Sekunden
Dashboard Beenden

Combo 1

Gl. Array 0,0320 Sekunden

Combo 2

Thread Liste:

Combo Company Start
 Combo Product Start
 Fill Grid Start
 Combo Company Array Start
 Combo Company Array Filled
 Fill Grid Finished
 Combo Products Filled
 Combo Company Filled

Profile An / Aus

Invoice No	Comp ID	Company Name	Paid
40	120	Acapulco Ropa Tropical	1.012,25
45	120	Acapulco Ropa Tropical	0,00
41	119	Amazon Apparel	1.225,00
42	119	Amazon Apparel	0,00
38	110	Anne's Attire	0,00
39	110	Anne's Attire	0,00
43	110	Anne's Attire	0,00
44	110	Anne's Attire	0,00
46	118	Australia Clothing Outlet	0,00
47	111	Betsy Ross American Fashions	0,00
54	111	Betsy Ross American Fashions	0,00
6	101	Clothing Connoisseurs	1.536,75
Count		Total	
54		47.786,95	

Einzelzeiten

Combo 1: 0,1160 Sekunden

Combo 2: 0,0920 Sekunden

Grid: 0,0680 Sekunden

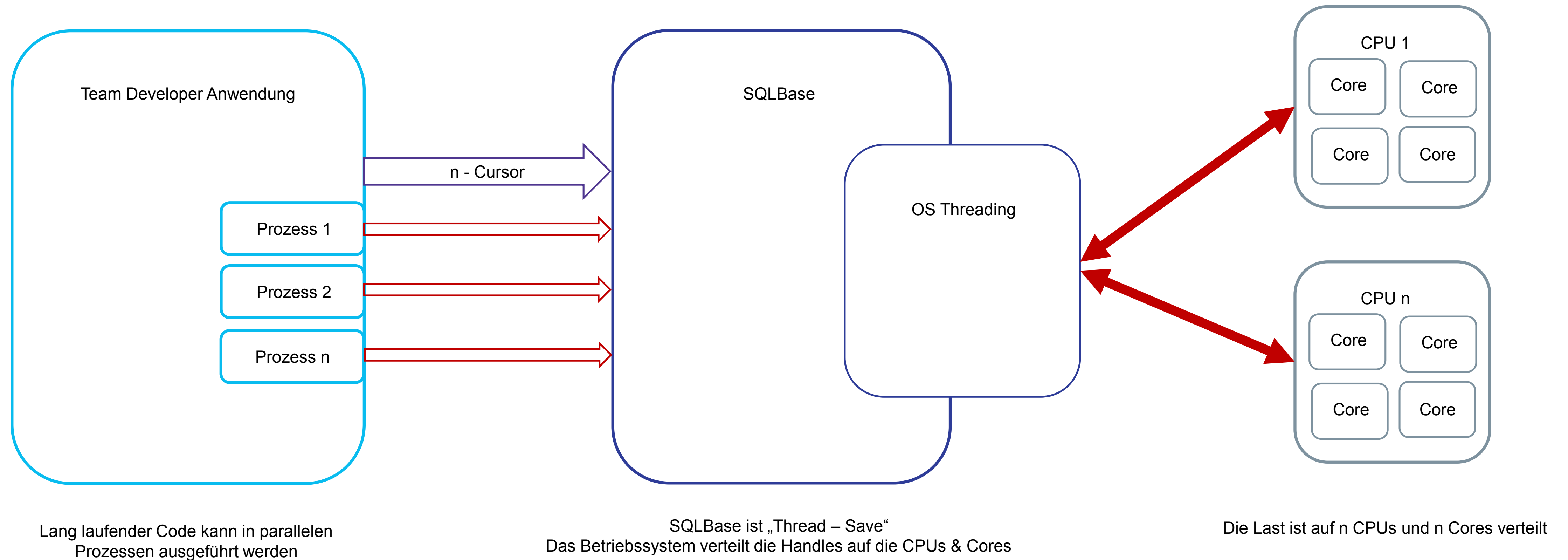
Summe: 0,2760 Sekunden

Reduzierte

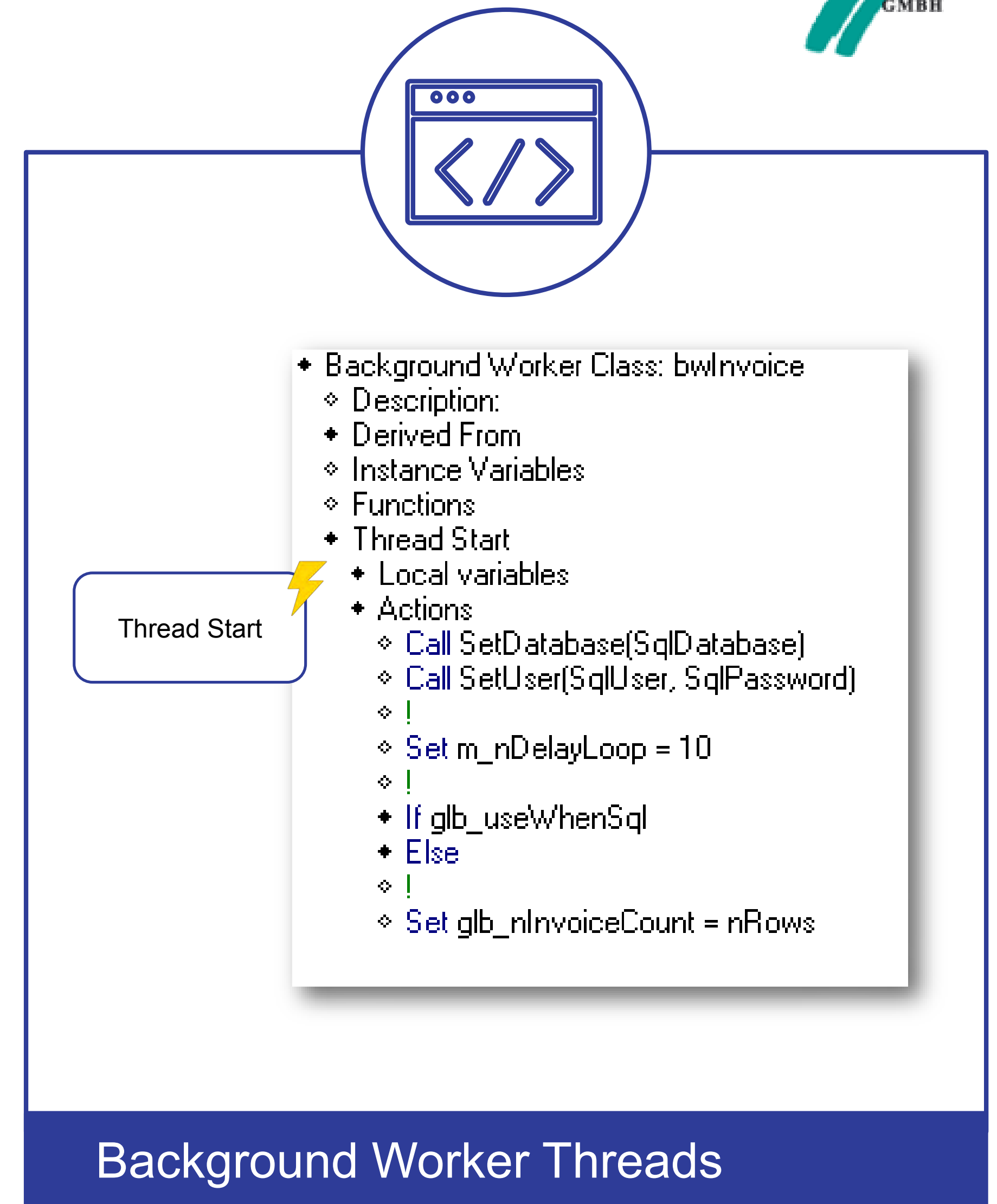
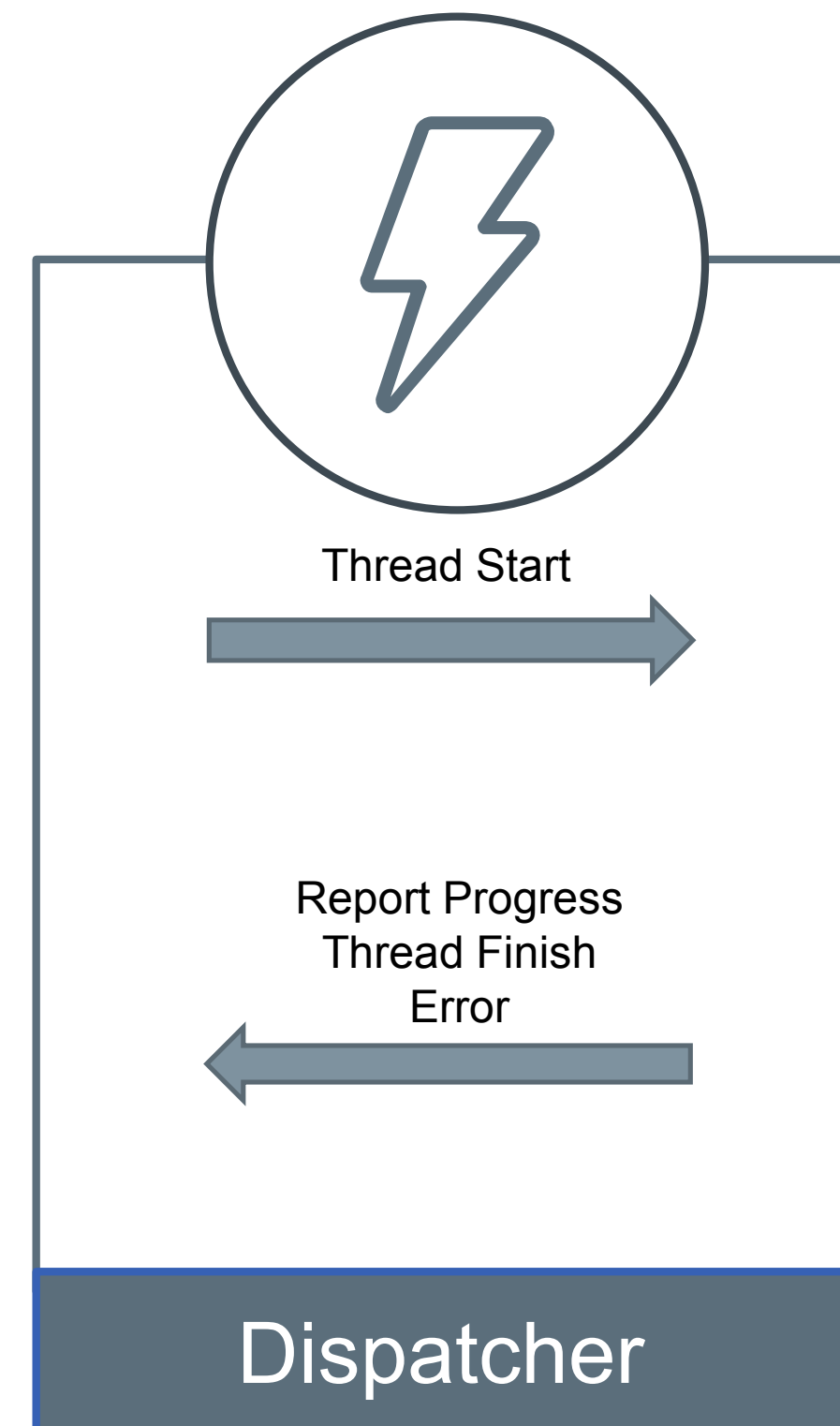
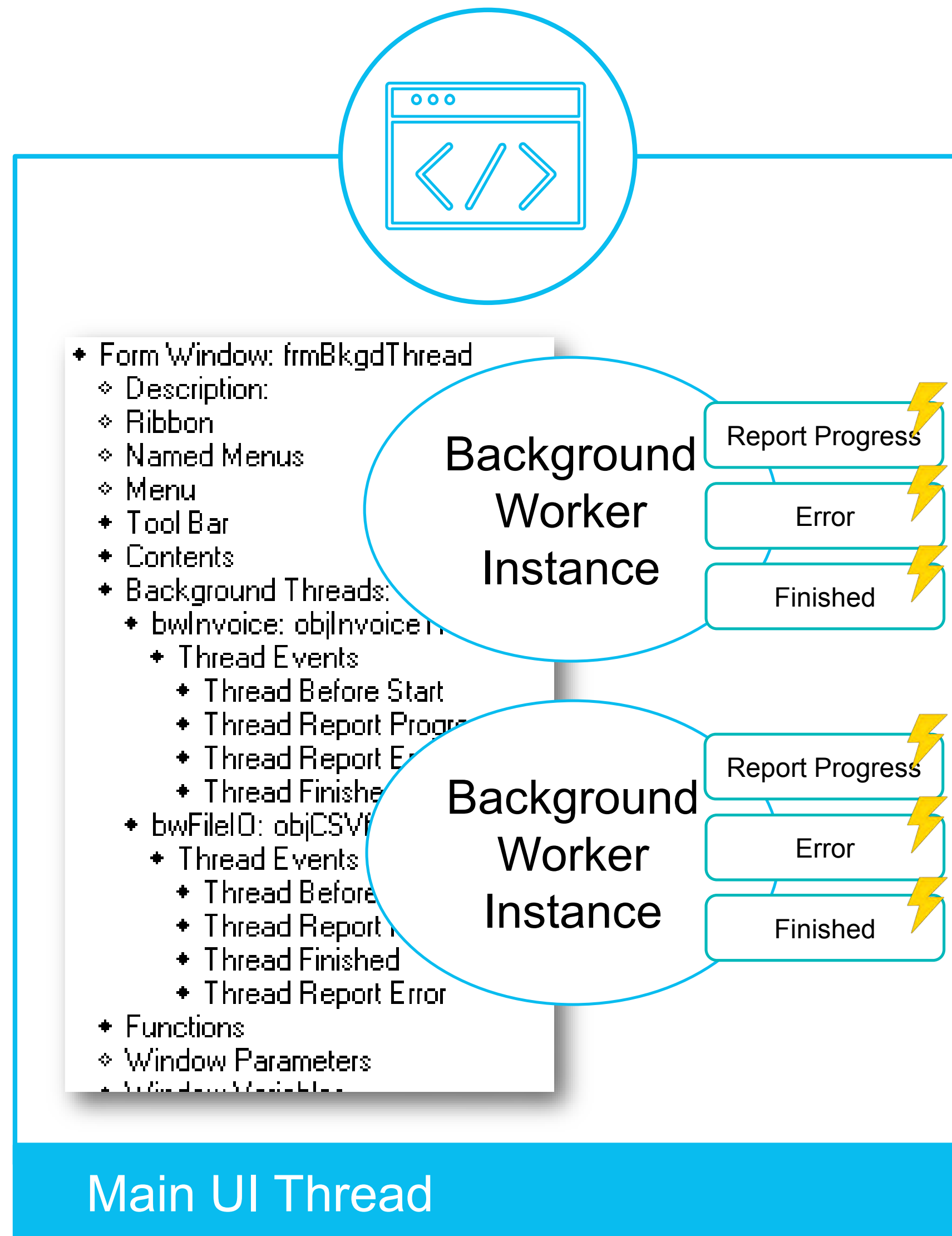
Wartezeit -0,2040 Sekunden

„Gupta Evolution“: Multi Threading in TD 7.1

- Versionen TD7.1 & SB12.1 (Oracle, MS-Server,)



Architektur



Thread Events

- **Thread Before Start**

- Der Code innerhalb des Thread startet erst, wenn der Event 'Thread Before Start' abgearbeitet ist: Setzen der Instanzvariablen, Übergabe von Variablen, Initialisierung.

- **Synchroner Event**

- **Thread Report Progress**

- Gesendet mit der API Funktion ***SalBackgroundWorkerReportProgress(nProgress, sMessage)***

- Sendet *nProgress* für einen prozentualen Fortschritt aus dem Background Prozess als Wert und den Message String.

- **Asynchroner Event**, der Hintergrund Prozess läuft weiter. Der Event Code läuft nur dann, wenn die GUI ohne Interaktion mit dem Anwender ist.

- **Thread Finished**

- Wird ausgelöst, wenn der Hintergrund Prozess beendet ist. Z.B.: Update der GUI mit Anzeige der Werte.

- **Asynchroner Event**

- **Thread Error**

- Wird ausgelöst, wenn ein unbehandelter Fehler im Hintergrund Prozess ausgelöst wird. Es ist sehr wichtig, dass keine MessageBox oder andere GUI Interaktion ausgelöst wird! Der Fehler sollte z.B. im Thread Error Event behandelt werden!

- **Asynchroner Event**

Neue Threading API

Neue API für die Ausführung von Hintergrund Prozessen:

- Hauptanwendung
 - ***SalBackgroundWorkerStart(oBkgdWorker)***
 - ***SalBackgroundWorkerIsBusy(objBkgdWorker)***
 - ***SalBackgroundWorkerCancel(objBkgdWorker)***
 - ***SalBackgroundWorkerAnyRunning(hWnd)***
- Hintergrundprozess
 - ***SalBackgroundWorkerReportProgress(nProgress, sMessage)***
 - ***SalBackgroundWorkerIsCanceled()***

Threads Window

- Erlaubt die Auswahl einzelner Prozessen
- Übersicht über Prozess Status:
 - *Running* – Thread führt Code aus
 - *Suspended* – Steht am Breakpoint, ist aber nicht aktiv (z.B. ein anderer Thread wird ge-debugged)
 - *Debugging* – Der aktive Thread wird ge-debugged
 - Hinweis: Unterschiedliche Darstellung der aktiven Zeilen (Mehrere Zeilen können aktiv sein, dunkle Farbe zeigt aktive Position und Thread an)

```

* Background Worker Class: bwlInvoice
  * Description:
  * Derived From
  * Instance Variables
  * Functions
  * Thread Start
  * Local variables
  * Actions
    * Call SetDatabase(SqlDatabase)
    * Call SetUser(SqlUser, SqlPassword)
      * |
      * Set m_nDelayLoop = 10
      * |
      * If glb_useWhenSql
      * Else
      * |
      * Set glb_nInvoiceCount = nRows
    * |
  * Functional Class: fcCreateCSVFromUDV
  * Background Worker Class: bwFileIO
  * Description:
  * Derived From
  * Instance Variables
  * Functions
  * Thread Start
  * Local variables
  * Actions
    * Call fcCreateCSVFromUDV( sInFile, sOutFile )
  * Application Actions
  * Form Window: frmBkgdThread
  * Description:
  * Ribbon
  * Named Menus
  * Menu
  * Tool Bar
  * Contents
  * ! thread sql operations
  * Pushbutton: pbPopulateClass
  * Message Actions
  * On SAM_Click
  * If cbGenError
  *   * Set glb_nGenSqlError = 1
  * Else
  *   * Set glb_nGenSqlError = 0
  * |
  * If cbUseWhenSqlError
  *   * Set glb_useWhenSql = TRUE
  * Else
  *   * Set glb_useWhenSql = FALSE
  * |
  * If NOT SalBackgroundWorkerStart( objInvoiceThread )
  *   * Call SalMessageBox(" Thread did not start - already running?", "objInvoiceThread", MB_Ok)
  * |

```

Thread Name	Thread ID	State
Main	170672	Running
objInvoiceThread	174016	Debugging
objCSVfile	170196	Suspended

Background Threads – Datenübergabe „Best Practice“

- Option 1: Übergabe der Daten „Thread Finished“

The screenshot shows a software window titled 'Thread Sample' with two tabs: 'Option 1' and 'Option 2'. The 'Option 1' tab is active. Inside, there is a sub-window titled 'Using Instance Variables' with a 'Thread Start' button. The interface is divided into two main sections: 'Manueller Start' and 'Threads'.

Manueller Start:

- Sequenziell: Combo Box (empty), Zeiten: 0,1700
- Liste: Anak, Tortuga; Becker, Frank; Fernandez, Alexis; Gena, Bernard; Geraldo, Helen. Zeiten: 0,0200
- Tabelle:

Style	Beschreibung	Preis
Five-O T...	These all cot...	8
Hang-Te...	Why would y...	15
Hawaiian...	Straight fro...	11
Kona Kh...	Khakis with ...	14
Lanai Lin...	Our hottest ...	9
Maui Tan...	Our tropical ...	6
North Sh...	It's not just a...	12

 Zeiten: 0,0100
- Summe: 0,2000

Threads:

- Threads: Combo Box (empty), Zeiten: 0,1800
- Liste: Anak, Tortuga; Becker, Frank; Fernandez, Alexis; Gena, Bernard. Zeiten: 0,1300
- Tabelle:

Style	Beschreibung	Preis
Five-O To...	These all cot...	8
Hang-Te...	Why would y...	15
Hawaiian ...	Straight fro...	11
Kona Kha...	Khakis with ...	14
Lanai Lin...	Our hottest ...	9
Maui Tan...	Our tropical ...	6
North Sh...	It's not just a...	12

 Zeiten: 0,0600
- Sum Einzelzeiten: 0,3700
- Form Aktiv nach: 0,1700

Background Threads – Datenübergabe „Best Practice“

- Option 1: Übergabe der Daten „Thread Finished“

- ◆ Background Worker Class: bwcCombo
 - ◇ Description: Background Worker zum Füllen von Combo Box
- ◆ Derived From
 - ◇ Class: clsCombo
- ◇ Instance Variables
- ◇ Functions
- ◆ Thread Start
 - ◇ Local variables
 - ◆ Actions
 - ◇ Call FillCombo()

Background Worker Klasse

- ◆ Thread Finished
 - ◆ Actions
 - ◆ While nCount < bcCombo.nItem
 - ◇ Call SalListAdd(cmbThread , bcCombo.sCompany[nCount])
 - ◇ Set nCount=nCount+1

Background Prozess wurde beendet: UDV in Child Window übertragen

- ◆ Functional Class: clsCombo
 - ◇ Description:
 - ◇ Derived From
 - ◇ Class Variables
 - ◆ Instance Variables
 - ◇ String: sCompany[*]
 - ◇ Number: nItem
 - ◇ Constructor/Destructor
 - ◆ Functions
 - ◆ Function: FillCombo
 - ◇ Description:
 - ◇ Returns
 - ◇ Parameters
 - ◇ Static Variables
 - ◆ Local variables
 - ◆ Actions
 - ◇ Set nItem = 0
 - ◇ Set sSQL = 'select Company_Name from Company into |:sCompany[nItem] order by Company_Name'
 - ◇ Call SqlConnectionEx(hSQL, 'ISLAND', 'SYSADM', 'SYSADM')
 - ◇ Call SqlPrepareAndExecute(hSQL, sSQL)
 - ◆ While SqlFetchNext(hSQL, nRet)
 - ◇ Call SqlConnection(hSQL)
 - ◇ Return TRUE

Ermitteln der Daten für die Combobox

Background Threads – Datenübergabe „Best Practice“

- Option 2: Übergabe der Daten „Thread Report Progress“

The screenshot displays a software window titled "Thread Sample" with two tabs: "Option 1" and "Option 2". The "Option 2" tab is active, showing a "Using Progress" dialog box. This dialog has two main sections: "Manueller Start" (Manual Start) and "Thread Start".

Manueller Start (Left Panel):

- Sequenziell:** A "Combo Box" and a "Liste" containing: Anak, Tortuga; Becker, Frank; Fernandez, Alexis; Gena, Bernard; Geraldo, Helen.
- Zeiten:** 0,0100 (for Combo Box), 0,0200 (for Liste), 0,0100 (for Tabelle).
- Tabelle:** A table with columns: Style, Beschreibung, Preis.

Style	Beschreibung	Preis
Five-O T...	These all cot...	8
Hang-Te...	Why would y...	15
Hawaiian...	Straight fro...	11
Kona Kh...	Khakis with ...	14
Lanai Lin...	Our hottest ...	9
Maui Tan...	Our tropical ...	6
North Sh...	It's not just a...	12

Thread Start (Right Panel):

- Threads:** A "Combo Box" and a "Liste" containing: Anak, Tortuga; Becker, Frank; Fernandez, Alexis; Gena, Bernard.
- Zeiten:** 0,1900 (for Combo Box), 0,1400 (for Liste), 0,0700 (for Tabelle).
- Tabelle:** A table with columns: Style, Beschreibung, Pr.

Style	Beschreibung	Pr
Five-O To...	These all cot...	
Hang-Te...	Why would y...	
Hawaiian ...	Straight fro...	
Kona Kha...	Khakis with ...	
Lanai Lin...	Our hottest ...	
Maui Tan...	Our tropical ...	
North Sh...	It's not just a...	

Summary:

- Summe:** 0,0400 (highlighted in orange)
- Sum Einzelzeiten:** 0,4000 (highlighted in orange)
- Form Aktiv nach:** 0,1500 (highlighted in orange)

Background Threads – Datenübergabe „Best Practice“

- Option 2: Übergabe der Daten „Thread Report Progress“

- ◆ Background Worker Class: bwcCombo_2
 - ◇ Description:
 - ◆ Derived From
 - ◇ Class: clsCombo
 - ◇ Instance Variables
 - ◇ Functions
 - ◆ Thread Start
 - ◇ Local variables
 - ◆ Actions
 - ◇ Call FillCombo_Progress()

Background Worker Klasse

- ◆ Thread Report Progress
 - ◆ Event Parameters
 - ◇ Number: nProgress
 - ◇ String: strProgress
 - ◆ Actions
 - ◇ Call SalListAdd(cmbThread, strProgress)

Background Prozess läuft: UDV in Child Window übertragen

- ◆ Function: FillCombo_Progress
 - ◇ Description:
 - ◇ Returns
 - ◇ Parameters
 - ◇ Static Variables
 - ◆ Local variables
 - ◇ String: sSQL
 - ◇ Sql Handle: hSQL
 - ◇ Number: nRet
 - ◇ Number: nCount
 - ◆ Actions
 - ◇ Set nCount = 0
 - ◇ Set sSQL = 'select Company_Name from Company into :sCompany[nCount] order by Company_Name'
 - ◇ Call SqlConnectEx(hSQL, 'ISLAND', 'SYSADM', 'SYSADM')
 - ◇ Call SqlPrepareAndExecute(hSQL, sSQL)
 - ◆ While SqlFetchNext(hSQL, nRet)
 - ◇ Call SalBackgroundWorkerReportProgress(nCount+1, sCompany[nCount])
 - ◇ Set nCount=nCount+1
 - ◇ Call SqlDisconnect(hSQL)
 - ◇ Return TRUE

Ermitteln der Daten für die Combobox

Background Threads – Datenübergabe „Best Practice“

- Übergabe der Daten an ein Grid / Table

- ◆ Thread Report Progress
 - ◆ Event Parameters
 - ◆ Actions
 - ◆ If gridThread.colDescription != STRING_Null and gridThread.colStyle != S
 - ◇ Call SalTblInsertRow(gridThread, TBL_MaxRow)
 - ◆ If nProgress = 0
 - ◇ Set gridThread.colStyle = strProgress
 - ◆ If nProgress = 1
 - ◇ Set gridThread.colDescription = strProgress
 - ◆ If strProgress = 'Preis'
 - ◇ Set gridThread.colPreis = nProgress

Thread Report Progress

- ◆ bwcGrid: bcGrid
 - ◆ Thread Events
 - ◆ Thread Before Start
 - ◆ Thread Finished
 - ◆ Actions
 - ◆ While nCount2 <= bcGrid.nItems
 - ◇ Call SalTblInsertRow(gridThread, TBL_MaxRow)
 - ◇ Set gridThread.colStyle = bcGrid.sStyle[nCount2]
 - ◇ Set gridThread.colDescription = bcGrid.sDescription[nCount2]
 - ◇ Set gridThread.colPreis = bcGrid.nPrice[nCount2]
 - ◇ Set nCount2=nCount2+1

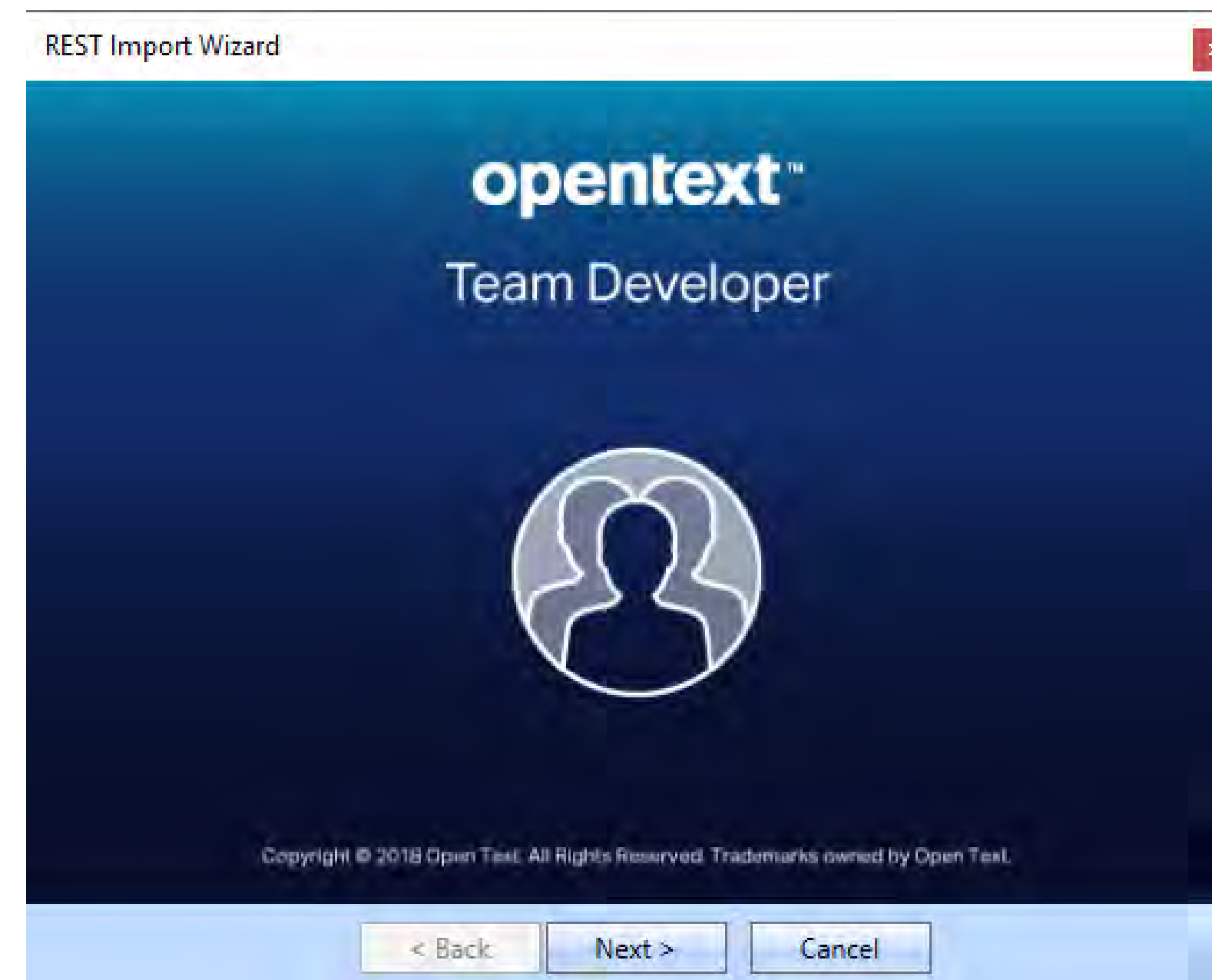
Thread Finished

Background Threads – Datenübergabe „Best Practice“

- Grundsätzliche Überlegungen...
 - Wann machen Background Threads Sinn?
 - Sicher nicht wenn z.B. SalTblPopulate(), SalListPopulate() usw. mit „einfachen & optimierten“ SQL-Statements genutzt werden.
 - Das Starten des Background Threads länger dauert als das Ausführen des kompletten SQL-Statements.
 - Sinnvoll, wenn ‚komplexe Daten‘ in ‚komplexe Forms‘ übertragen werden müssen: Tabs mit unterschiedlichen Inhalten.
 - Langwierige DB Operationen
 - Wiederkehrende Transaktionen („File I/O“, Einlesen & Aufbereiten von Daten)
 - Der Anwender wird in seiner Arbeit mit der Anwendung „blockiert“
 - Er wartet bis die Anwendung „wieder da“ ist.
 - Analyse des „Warum“: Analyse mit dem „Profiling-Viewer“

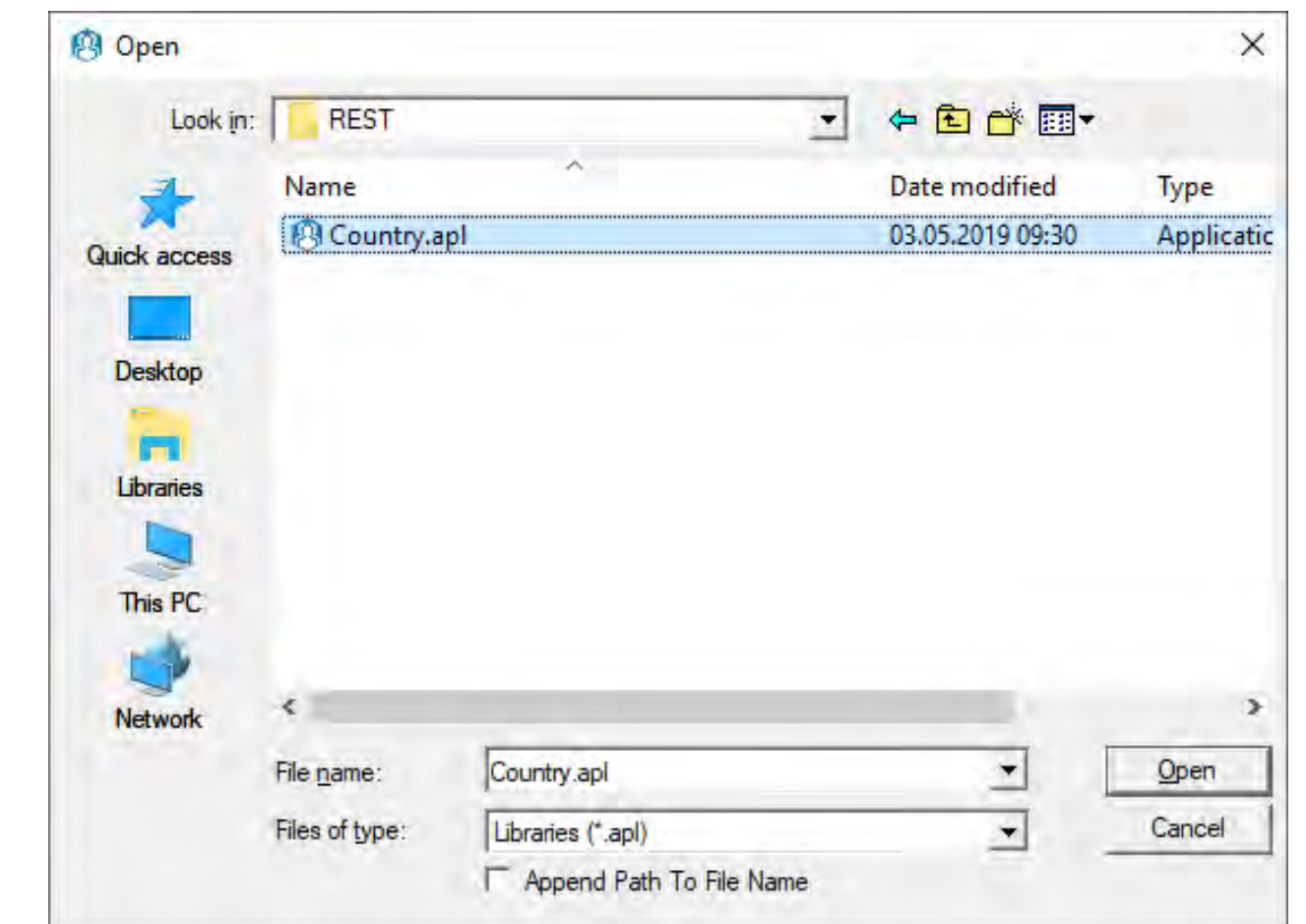
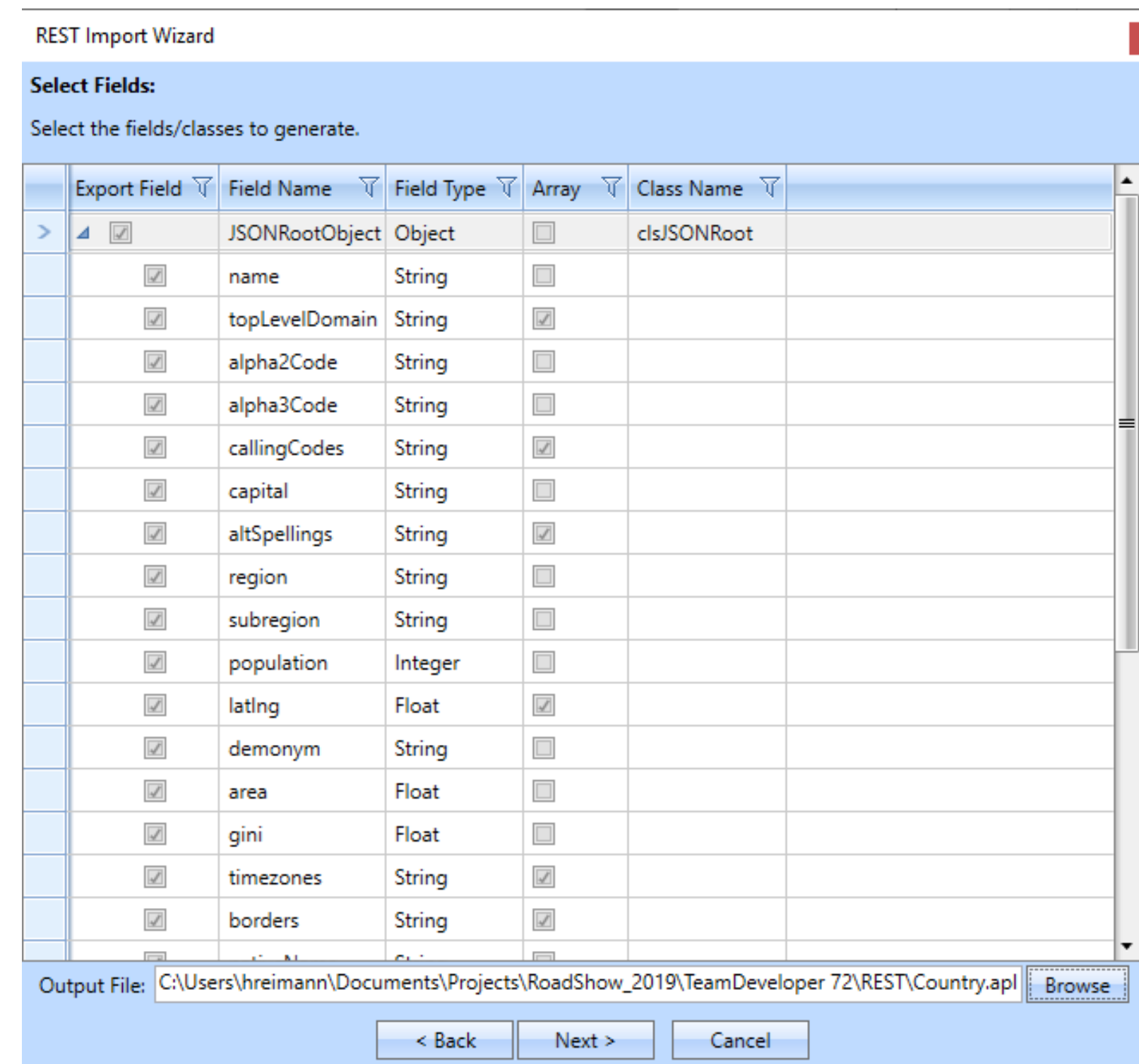
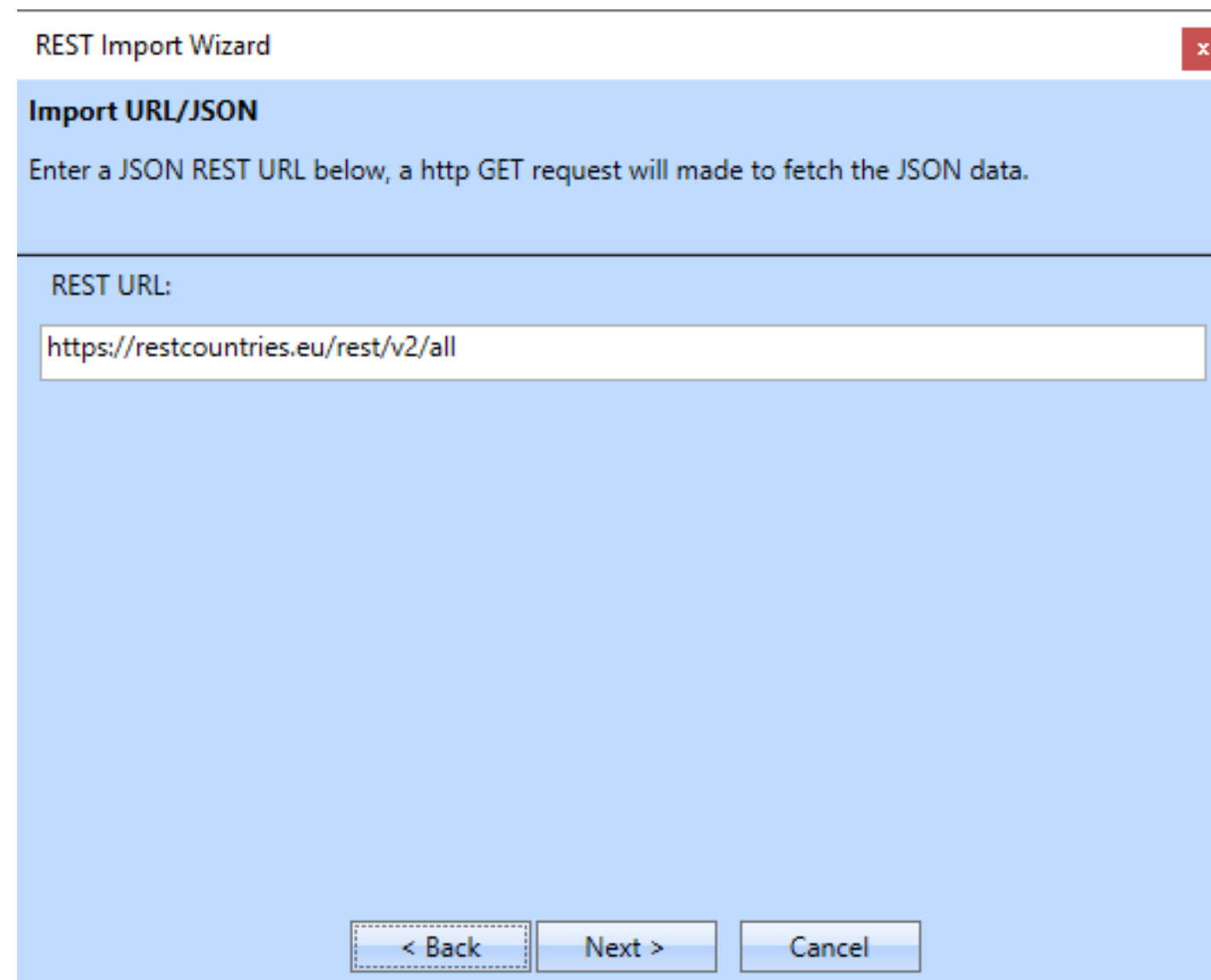
„REST-Importer“

- Neues Werkzeug um Funktionale Klassen basierend auf REST- WS zu generieren



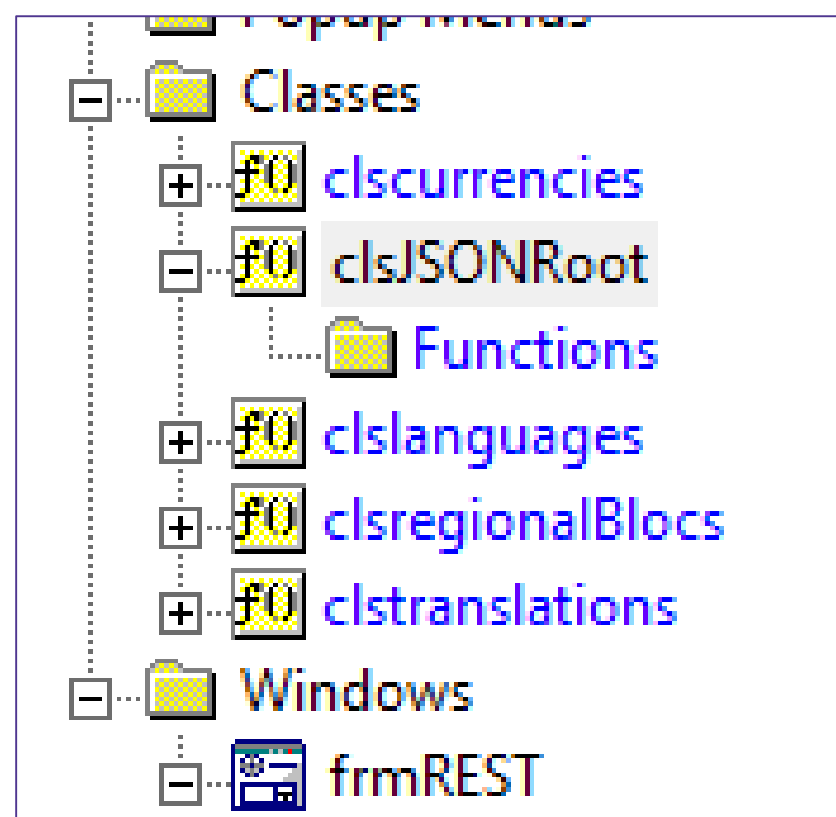
„REST-Importer“

- Neues Werkzeug um Funktionale Klassen basierend auf REST- WS zu generieren



„REST-Importer“

- Generierte Funktionelle Klassen:



- ◆ Instance Variables
 - ◇ String: name
 - ◇ String: topLevelDomain[*]
 - ◇ String: alpha2Code
 - ◇ String: alpha3Code
 - ◇ String: callingCodes[*]
 - ◇ String: capital
 - ◇ String: altSpellings[*]
 - ◇ String: region
 - ◇ String: subregion
 - ◇ Number: population
 - ◇ Number: latlng[*]
 - ◇ String: demonym
 - ◇ Number: area
 - ◇ Number: gini
 - ◇ String: timezones[*]
 - ◇ String: borders[*]
 - ◇ String: nativeName
 - ◇ String: numericCode
 - ◇ clsCurrencies: currencies[*]
 - ◇ clsLanguages: languages[*]
 - ◇ clsTranslations: translations
 - ◇ String: flag
 - ◇ clsRegionalBlocs: regionalBlocs[*]
 - ◇ String: cioc

„REST-Importer“

- REST – WS Aufruf:
 - SalJSONDeserializeUDV(...)
 - SalJSONSerializeUDV(...)

- Abfrage eines REST-WS:

```
◇ Call SalJSONDeserializeUDV( "oREST_WS", dfURL, sREST_User, sREST_PWD, sErr )
◇ !
◆ If sErr != STRING_Null
◇ Set dfHauptstadt = oREST_WS.capital
◇ Set dfRegion = oREST_WS.region
◇ Set dfBezirk = oREST_WS.subregion
◇ Set dfEinwohner = oREST_WS.population
◇ Set dfNativ = oREST_WS.nativeName
◇ Set dfAlpha2Code = oREST_WS.alpha2Code
◇ Set dfAlpha3Code = oREST_WS.alpha3Code
◇ Set dfCioc = oREST_WS.cioc
◇ !
```


REST - API Erweiterung

- REST – WS Aufruf:
 - SalJSONDeserializeUDVEx(sObjectName, sURL, sUserName, sPassword, **arHeaders**, **arParams**, sError)
 - SalJSONSerializeUDVEx(sObjectName, sURL, nHttpVerb, sResult, sUserName, sPassword, **arHeaders**, **arParams**, sError)

Zusätzliche Übergabe von Header / Parameter als Arrays

Report Designer



- Alternative zum Report Builder !?

OpenText Gupta Report Builder 7.2 - [AuswertungGes.qrp]

Report Objects:

- Input Items
- Input Variables
- Formulas
- Totals
- Crosstabs
- Break Groups
 - E_ID1

Report Design:

Report Header: 0 Lines

Page Header:

Ergebnisse Gesamtauswertung

Name: S_ID;Name24
 Geb.-Datum: Geb_Datum2
 Disziplin: Kaliber30
 Klasse: KL26Klasse27
 Serien: Coun Start-Dat: Start21 End-Dat: Ende22 Pers. Vorgabe

E_ID	Datum	Serie	Ringe	Schuß	DS	Verschenkt	Waffe
10er	9er	8er	7er	6er	5er	4er	3er
10er	9er	8er	7er	6er	5er	4er	3er

Header: E_ID1: 0 Lines

Detail Block:

E_ID1	Datum2	Serie3	Ringe4	Schuß5	DS6	Verschenkt7	Waffe8	
A1111	A1212	A1313	A1414	A1515	A1616	A1717	A1818	A1919

Footer: E_ID1: 0 Lines

Page Footer:

Report Builder 7.2

Gupta Bericht Designer 2.3

Report1: AuswertungGes [modified]

Layout: Formula31 x Formula62 x Formula65 x Formula67 x Formula73 x

Report Design:

Report Header

Page Header:

Ergebnisse Gesamtauswertung

Name: S_ID;Name24
 Geb.-Datum: Geb_Datum2
 Disziplin: Kaliber30
 Klasse: KL26Klasse27
 Serien: Coun Start-Dat: Start21 End-Dat: Ende22 Pers. Vorgabe: Vorgabe29

Klasse	S_Vorgabe	Serie	Ringe	Schuß	DS	Verschenkt	Waffe	Auszeichnung		
10er	9er	8er	7er	6er	5er	4er	3er	2er		
10er	9er	8er	7er	6er	5er	4er	3er	2er		
								Auszeichnung9		
A1111	A1212	A1313	A1414	A1515	A1616	A1717	A1818	A1919	A2020	Hinweis10

Header: E_ID1

Detail Block:

Vorgabe29	inge4	Schuß5	DS6	Verschenkt7	Waffe8	Auszeichnung9				
A1111	A1212	A1313	A1414	A1515	A1616	A1717	A1818	A1919	A2020	Hinweis10

Footer: E_ID1

Page Footer:

Report Footer:

Report Designer

Report Designer

- Alternative zum Report Builder !?
 - Web-Reporting = Erzeugen der Reports auf dem **WEB-Server**
 - Einsatz in **Web Services**
 - HTML oder PDF Ausgabeformat
 - Kein Report Messaging (nicht Event-Gesteuert!)
 - Existierende QRPs können geladen und als Report Designer Templates (*.rpx) abgespeichert werden

Report Designer (Erzeugen eines Reports in WS)



- Report Designer API
 - **SalWebReportCreate**
(sRPXFile, bOutputFile, reportVars, reportData, nOutputType, sLang)
 - Anzeige des Reports im Browser! (Z.B. .NET xBAP Anwendung)
 - **SalWebReportCreateToFile**
(sRPXFile, sOutputFile, reportVars, reportData, nOutputType, sLang)
 - Generieren eine Report Files (HTML oder PDF)
- Nur für .NET Build Targets

Sonstige Neuerungen (API)

- `SqlConnectionEx(hSql_Handle, sDatabase, sUser, sPassword)`
 - Z.B. in Background Threads

```
◆ Actions
  ◇ Set nItem = 0
  ◇ Set sSQL = 'select Company_Name from Company into
               :sCompany[nItem] order by Company_Name'
  ◇ !
  ◇ Call SqlConnectionEx( hSQL, 'ISLAND', 'SYSADM', 'SYSADM' )
  ◇ !
  ◇ Call SqlPrepareAndExecute( hSQL, sSQL )
  ◆ While SqlFetchNext( hSQL, nRet )
```


Sonstige Neuerungen - API

- `Set sVerschlüsselt = SalStrEncrypt(sText, sKey)`
 - Verschlüsselt einen String mit AES – Verschlüsselung basierend auf einem Key
- `Set sText = SalStrDecrypt(sVerschlüsselt, sKey)`
 - Entschlüsseln eines String der mit `SalStrEncrypt` verschlüsselt wurde

◇ `Set dfVerschuesselt = SalStrEncrypt(dfOriginal, dfKey)`
◇ `Set dfEntschluesselt = SalStrDecrypt(dfVerschuesselt, dfKey)`

Sonstige Neuerungen - API

- **SalGridsRowExcludedByFilter()**
 - Überprüft, ob bestimmte Zeilen in einem Grid durch den Anwender ausgeblendet wurden.
 - Benutzen eines Filters
 - **SalGridsRowExcludedByFilter(grid1,nRow)**

◇ [Call SalGridsRowExcludedByFilter\(grid1, 2 \)](#)

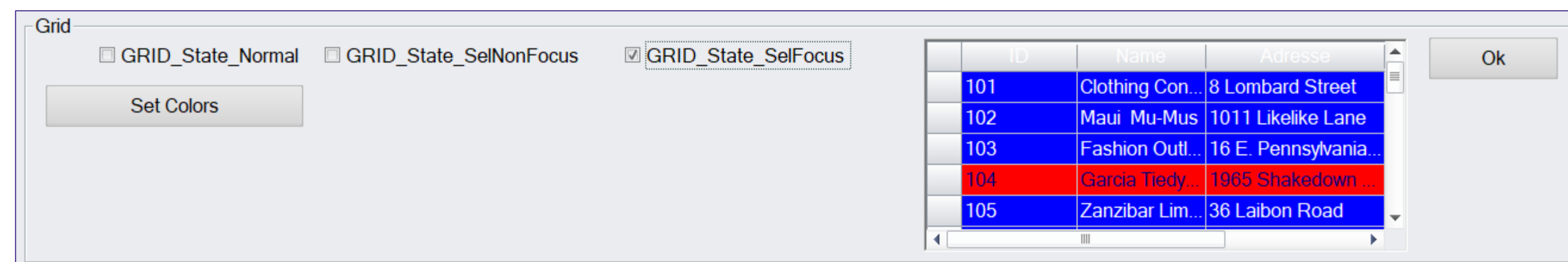
Sonstige Neuerungen - API

- Text / Hintergrund Farben ändern je nach Status einer Zeile:
 - SalGridBackColorSet(hGrid, nStatus, nFarbe)
 - SalGridTextColorSet(hGris, nStatus, nFarbe)

```

♦ On SAM_Click
  ♦ If cbGrid_State_Normal
    ◇ Call SalGridBackColorSet( grid1, nGridState, COLOR_Blue )
    ◇ Call SalGridTextColorSet( grid1, nGridState, COLOR_White )
  ♦ If cbGRID_State_SelfFocus
    ◇ Call SalGridBackColorSet( grid1, nGridState, COLOR_Red )
    ◇ Call SalGridTextColorSet( grid1, nGridState, COLOR_DarkBlue )
  ♦ If cbGRID_State_SelNonFocus
    ◇ Call SalGridBackColorSet( grid1, nGridState, COLOR_DarkGreen )
    ◇ Call SalGridTextColorSet( grid1, nGridState, COLOR_DarkRed )

```



Sonstige Neuerungen - API

- Text / Hintergrund Farben je nach Status einer Zeile ermitteln:
 - SalGridBackColorGet(hGrid, nStatus)
 - SalGridTextColorGet(hGrid, nStatus)

```

◇ Set nColorBNormal = SalGridBackColorGet( grid1, GRID_State_Normal )
◇ Set nColorBNoSelect = SalGridBackColorGet( grid1, GRID_State_SelNonFocus )
◇ Set nColorBSelect = SalGridBackColorGet( grid1, GRID_State_SelFocus )
◇ !
◇ Set nColorTNormal = SalGridTextColorGet( grid1, GRID_State_Normal )
◇ Set nColorTNoSelect = SalGridTextColorGet( grid1, GRID_State_SelNonFocus )
◇ Set nColorTSelect = SalGridTextColorGet( grid1, GRID_State_SelFocus )

```

Grid

GRID_State_Normal GRID_State_SelNonFocus GRID_State_SelFocus

Set Colors Get Colors

Hintergrund	16711680	32768	255
Text	16777215	128	8388608

ID	Name	Adresse
101	Clothing Con...	8 Lombard Street
102	Maui Mu-Mus	1011 Likelike Lane
103	Fashion Outl...	16 E. Pennsylvania...
104	Garcia Tiedy...	1965 Shakedown ...
105	Zanzibar Lim...	36 Laibon Road

Sonstige Neuerungen - API

- Strings überprüfen mit „Regular Expressions“
 - SalRegexMatch(strInput, nStart, nEnd, strRegex)
 - ◇ **Set** bMatch = SalRegexMatch(dfText, dfStart, dfEnde, dfMatch)
 - SalRegexSearch(strInput, nStart, nEnd, strRegex, nPosition, strMatched)
 - ◇ **Call** SalRegexSearch(dfText, dfStart, dfEnde, dfSearch, dfPosition, dfMatch)
 - SalRegexReplace(strInput, nStart, nEnd, strRegex, strReplace, strResult)
 - ◇ **Call** SalRegexReplace(dfText, dfStart, dfEnde, dfSearch, dfReplace, dfOut)
 - SalRegexReplaceAll(strInput, nStart, nEnd, strRegex, strReplace, strResult)
 - ◇ **Call** SalRegexReplaceAll(dfText, dfStart, dfEnde, dfSearch, dfReplace, dfOut)

Beispiel: Regular Expression: Reima(n+) **oder** [\w-]+@([\w-]+\.)[\w-]+ (Mail)

Hinweis: ECMAScript – Format in TD: <https://docs.microsoft.com/en-us/cpp/standard-library/regular-expressions-cpp?view=vs-2017>

“This is closest to the grammar used by JavaScript and the .NET languages”

Sonstige Neuerungen - API

- Komprimieren / Dekomprimieren von Strings nach / von Binary
 - SalStrCompress(sInput, binOutput)
 - SalStrUncompress(binValue, sOutput)

- ◇ Call SalStrCompressEx(dfIn, binOut)
- ◇ Call SalStrUncompressEx(binOut, dfOrg)

Sonstige Neuerungen - API

- Neue „Formate“: ENC_BIN / ENC_BASE64
 - `binStr = SalStringToBinary(sInput, nEncoding)`
 - `sOut = SalBinaryToString(binInput, nEncoding)`

```
◇ Set binStr = SalStringToBinary( dfInStr, ENC_BASE64 )  
◇ Set dfOutStr = SalBinaryToString( binStr, ENC_ANSI )
```

- `nEncoding` = das jeweilige Ziel!

Hinweis:

`ENC_BASE64` wird häufig in WS genutzt: z.B. beim Übertragen von Dokumenten in REST WS

Sonstige Neuerungen - API

- Grid

- Flags für SalTblSetTableFlags()

- TBL_Flag_VirtualScroll

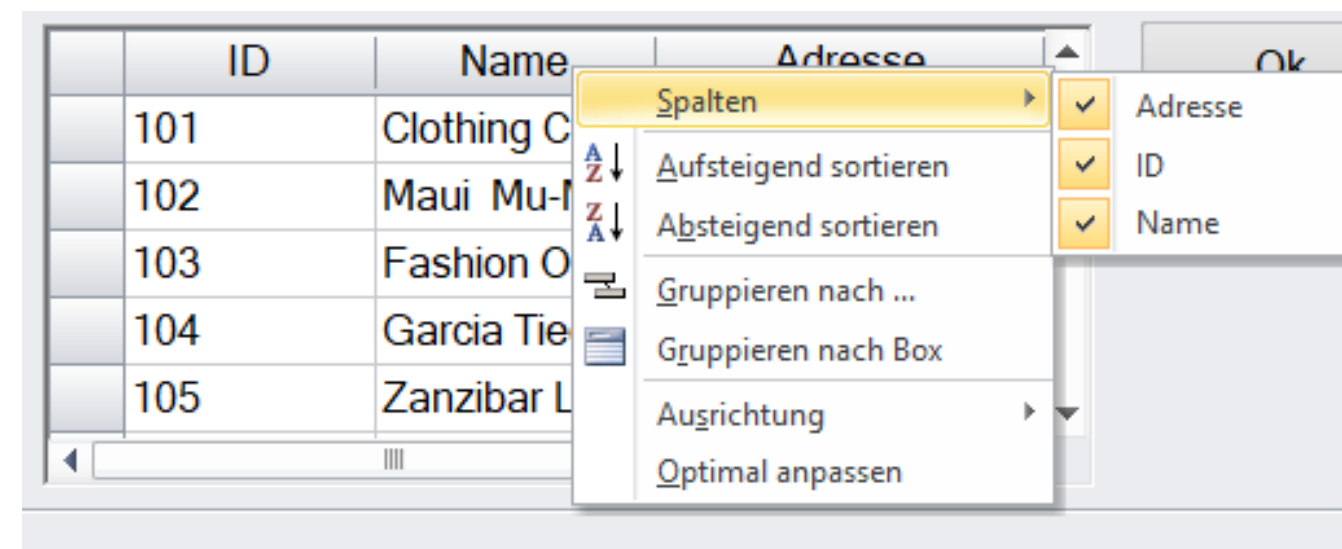
- Nur für das Grid
- In Verbindung mit SalTblPopulate: Bessere Performance
- Sortieren & Gruppieren wird abgeschaltet!

- TBL_Flag_CtxMenuSortAlpha

- Wenn TRUE: Sortieren in alphabetischer Reihenfolge anstatt Position der Spalte

◇ Call SalTblSetTableFlags(grid1, TBL_Flag_VirtualScroll, TRUE)
◇ Call SalTblPopulate(grid1, hSQL, sSQL, TBL_FillAll)

◇ Call SalTblSetTableFlags(grid1, TBL_Flag_CtxMenuSortAlpha, TRUE)
◇ Call SalTblPopulate(grid1, hSQL, sSQL, TBL_FillAll)

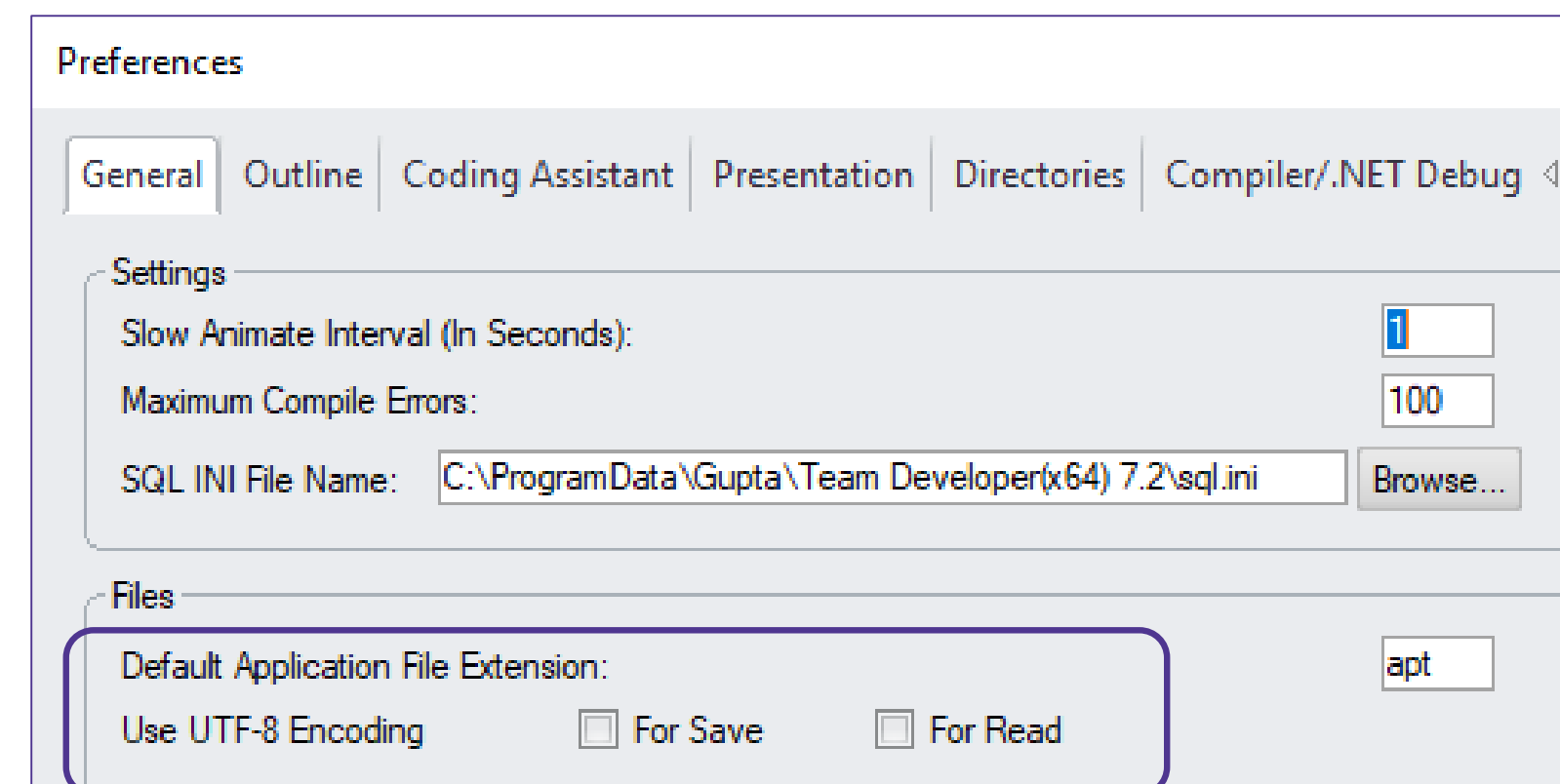


Sonstige Neuerungen - API

- Tab - Control
 - bAktivated = SalTablsWindowActivated(hTab, nTabIndex)
 - ◇ **Set** bAktivated = SalTablsWindowActivated(myTab, 2)
 - Set hWnd = SalTabAttachWindowAsChild(hTab, nTabIndex, hNewChild)
 - ◇ **Set** nWnd = SalTabAttachWindowAsChild(hTab, nIndex, hNewChild)
- hNewChild: Zur Laufzeit generiertes Objekt

Sonstiges

- Preference Settings:
 - Speichern als UTF-8 als alternative zu UTF-16
 - Einige Sourcecode - Archivierungssysteme arbeiten besser mit UTF-8
 - UTF-16 = Standard



Sonstiges

- SqlPassword
 - Ist nun im Arbeitsspeicher (RAM) verschlüsselt
 - Nicht mehr lesbar beim Debuggen oder beim zurückschreiben in eine Variable
- SQL Server 2017 Support
- Windows Server 2019 Support
- Support for Oracle 18c (auch schon in TD7.1.2)

opentext™

Danke!



twitter.com/opentext



linkedin.com/company/opentext



opentext.com