OPENTEXT™

Enabling the Digital World

Team Developer 7.0

Teil 3

OpenText Workshop

Agenda

OpenText
TD 7.0
Workshop

- TD 7.1 Ausblick
  - Multi-Threading
  - Team Features

Multithreading

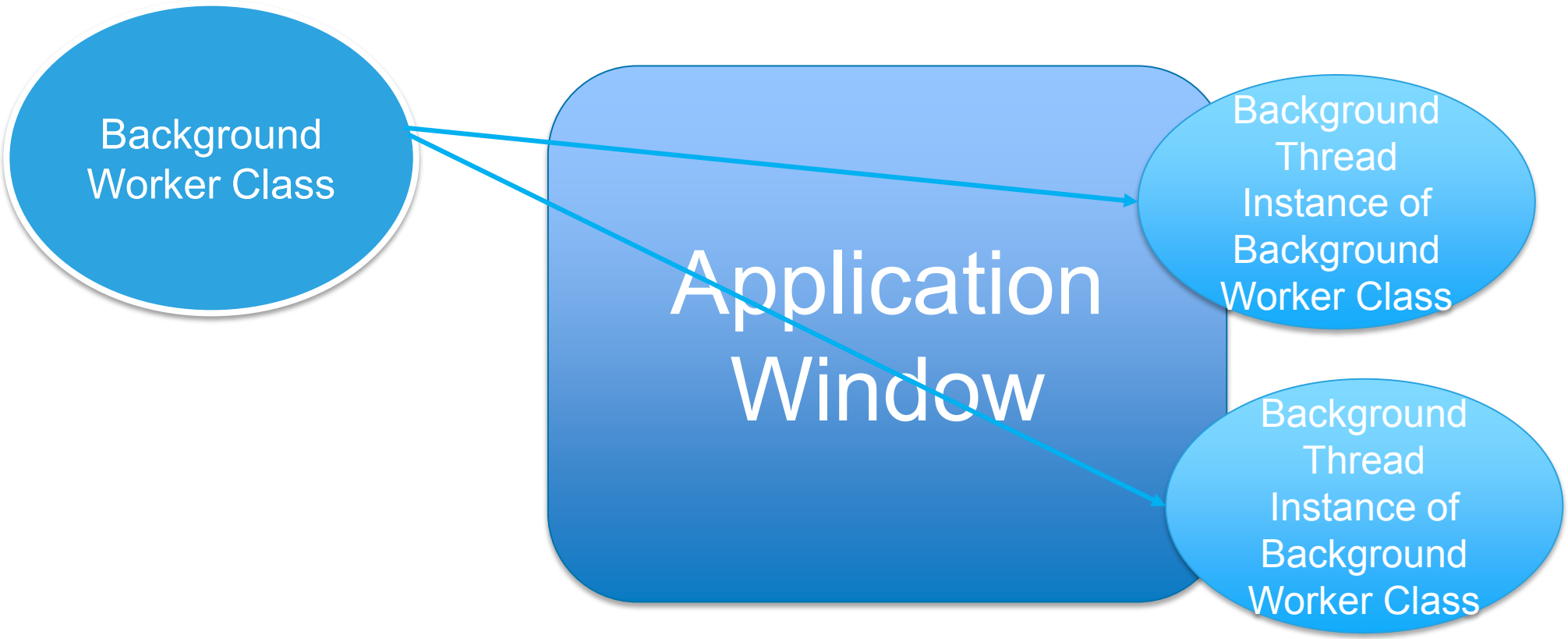
OpenText
TD 7.0
Workshop

- Ziel:
  - Definierte Abläufe einer Anwendung sollen im Hintergrund abgearbeitet werden
  - Der Anwender soll weiterhin mit der Anwendung arbeiten können
  - Einfache OOP basierende Implementation
  - Kontrolle und Interaktion mit der Hauptanwendung, basierend auf Events

Multithreading

OpenText
TD 7.0
Workshop

- Implementierung



Multithreading

OpenText
TD 7.0
Workshop

- Implementierung
  - Definieren einer Background Worker Class
  - Hinzufügen der Background Worker Class in ein (MSI-) Window
  - Innerhalb des Window Objekts
    - Start des Background Thread mit SalBackgroundWorkerStart(oBkgdWorker)
    - Der Thread wird gestartet und das Thread Event System wird initialisiert

Multithreading

OpenText
TD 7.0
Workshop

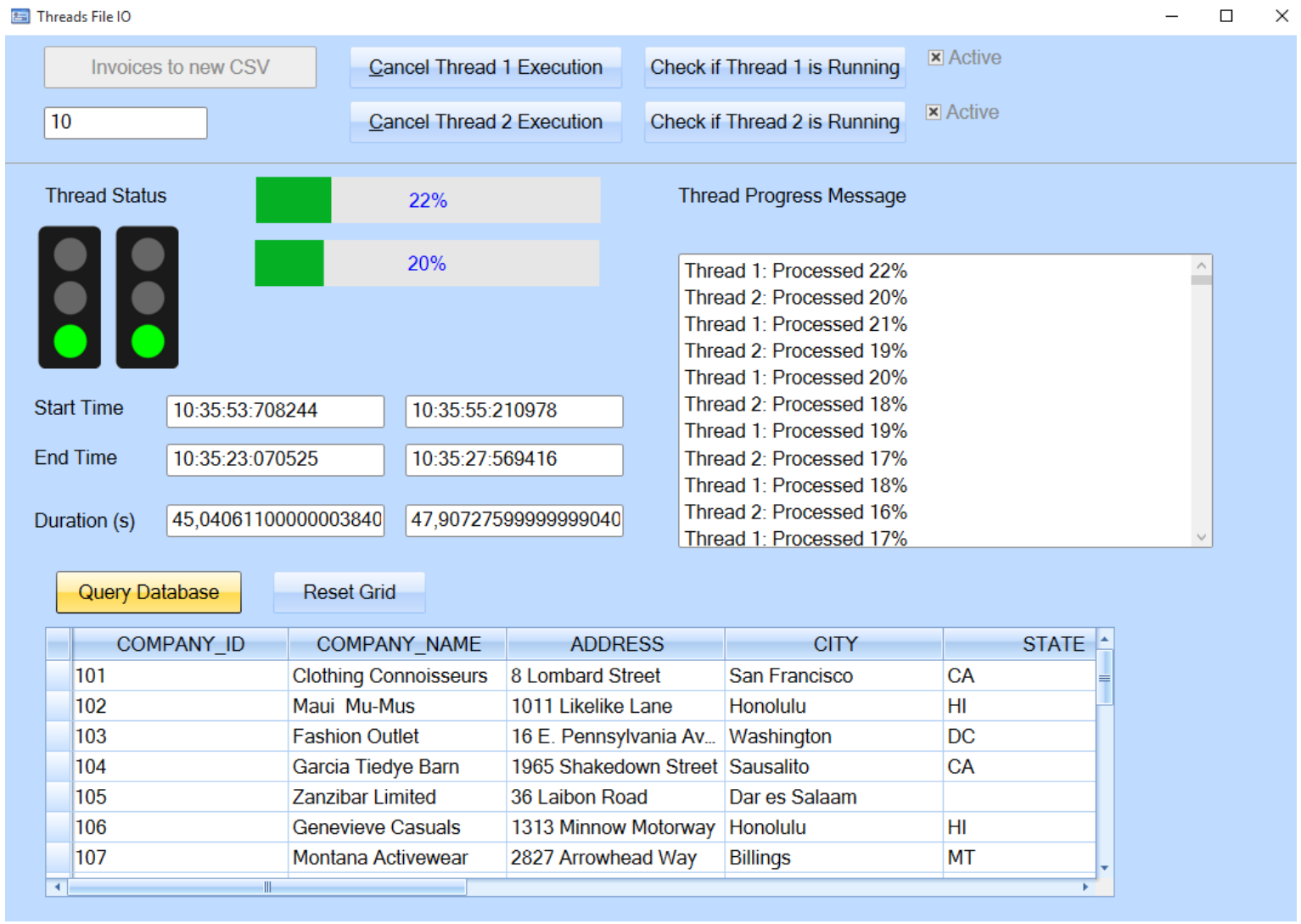
- Beispiel einer Background Worker Class:

- Background Worker Class: bwFileIO
  - Description:
  - Derived From
    - Class: fcCreateCSVFromCSV
  - Instance Variables
    - String: sInFile
    - String: sOutFile
    - Number: nTimes
  - Functions
  - Thread Start
    - Local variables
    - Actions
      - Call fCreateCSVFromCSV(nTimes, sInFile, sOutFile)

- **Thread Klassen Definition**
- **Instanz Variablen zum initialisieren des Prozesses**
- **Thread Code**
  - **Lang laufende Database Prozesse**
  - **Lang laufende .NET Reports**
  - **Etc.**
- **Achtung: Keine GUI interaction möglich!**

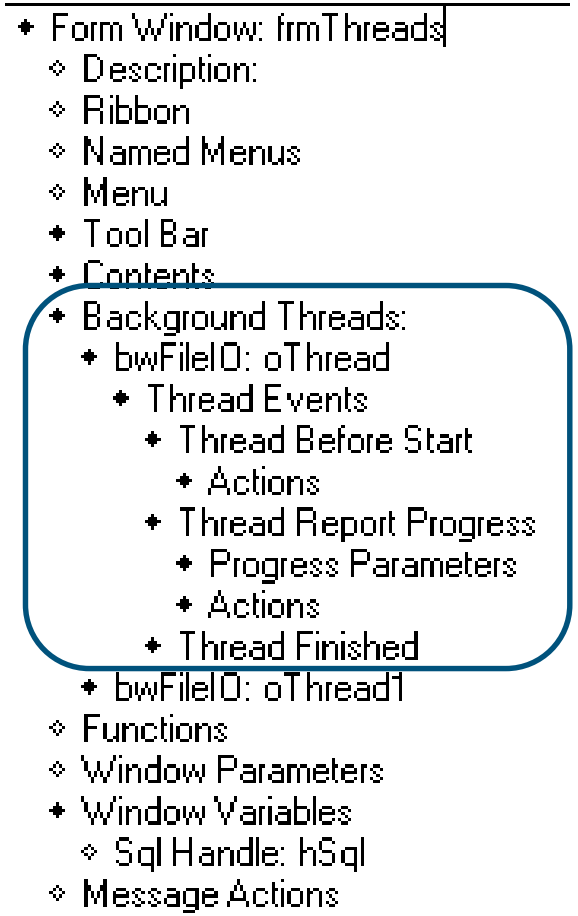
Multithreading

OpenText
TD 7.0
Workshop



Multithreading

OpenText
TD 7.0
Workshop



- Thread Instanzen beinhalten Thread-Events
- SalBackGroundWorkerStart(ThreadInstance) startet den Thread
- Thread Events werden ausgelöst

Event: Thread Before Start

OpenText
TD 7.0
Workshop

- Thread Before Start
  - Actions
    - Call SalListInsert(lbThreadProgress, 0, "Entering On Thread 1 Before Start.")
    - Call SalListInsert(lbThreadProgress, 0, "Thread 1 in file = invoice-in.csv, out file = invoice-out.csv.")
    - Set oThread.sInFile = "invoice-in.csv"
    - Set oThread.sOutFile = "invoice-out.csv"
    - Set oThread.nTimes = dfLoops
    - Call SalPicSet(picThread, orange, PIC_FormatPNG)
    - Call SalMeterSetRange(prbThread, 0, 100)
    - Call SalMeterSetStepSize(prbThread, 1)
    - Call SalMessageBox("Thread 1 is going to start", "Thread Hint", MB_Ok)
    - Call SalPicSet(picThread, green, PIC_FormatPNG)
    - Call SalListInsert(lbThreadProgress, 0, "Leaving On Thread 1 Before Start.")

- Initialisierung des Hintergrund Prozesses
- Thread startet am Ende des Events
- Returns FALSE wenn bereits ein Hintergrund Prozess dieser Worker Klasse läuft.

Event: Thread Report Progress

OpenText
TD 7.0
Workshop

- Thread Report Progress
  - Progress Parameters
  - Actions
    - Call SalListInsert(lbThreadProgress, 0, "Thread 1: " || strProgress)
    - Set prbThread = nProgress

- Hintergrundverarbeitung läuft
- Zeige Fortschritt
- Usw.

Event: Thread Finished

OpenText
TD 7.0
Workshop

- Thread Finished
  - Actions
    - Call SalPicSet(picThread , red, PIC_FormatPNG)
    - Call SalListInsert(lbThreadProgress, 0, "Thread 1 has finished.")
    - Set dfEndTime = SalDateCurrent( )
    - Set dfDuration = (dfEndTime - dfStartTime) * 24 * 60 * 60
    - Call SalDisableWindow(pbThreadCancel)
    - Call SalEnableWindow(pbExport)

- Thread wurde beendet
- Daten darstellen
- Usw.

Starten eines Thread

OpenText
TD 7.0
Workshop

```
Pushbutton: pbExport
  Message Actions
    On SAM_Click
      Set dfStartTime = SalDateCurrent( )
      Call SalBackgroundWorkerStart( oThread )
      Set dfStartTime1 = SalDateCurrent( )
      Call SalBackgroundWorkerStart( oThread1 )
      Call SalEnableWindow( pbThreadCancel )
      Call SalEnableWindow( pbThreadCancel1 )
      Call SalDisableWindow( pbExport )
      Set prbThread = 0
      Set prbThread1 = 0
```

- Der Background Worker wird als Instanz gestartet und löst die Eventverarbeitung aus

Threading API

OpenText
TD 7.0
Workshop

- SalBackgroundWorkerStart(oBkgdWorker)
- SalBackgroundWorkerReportProgress(objBkgdWorker, nProgress, sMessage)
- SalBackgroundWorkerIsBusy(objBkgdWorker)
- SalBackgroundWorkerCancel(objBkgdWorker)
- SalBackgroundWorkerIsCanceled()

Team Features

OpenText
TD 7.0
Workshop

- Integration zu Source Management & Versioning Systemen
  - GIT / SVN / Team Foundation Server

OpenText
TD 7.0
Workshop

OpenText™

www.opentext.com

 twitter.com/opentext

 facebook.com/opentext

 linkedin.com/company/opentext