



Team Developer 7.0

Teil 1

OpenText Workshop 2017

Agenda (Teil 1)

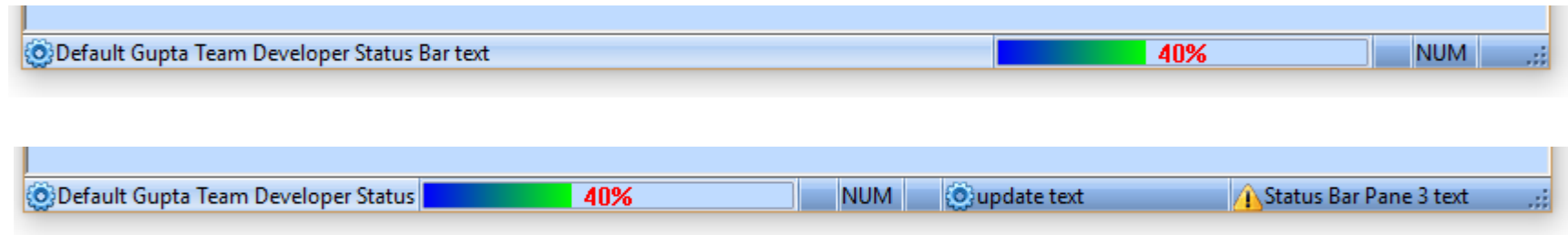


- Team Developer 7.0
 - Neues GUI Control
 - API Erweiterungen
 - Win32 / Win64
 - REST Web Services

Team Developer 7.0



- Neues GUI Control: **Status Bar**
 - Multiple Panes
 - Text
 - Tooltips
 - Progress Bar
- Vollständige API



Team Developer 7.0



■ Status Bar API Funktionen

- Erzeugen einer Pane
 - `SalStatusAddPane()`
- Pane sichtbar / unsichtbar
 - `SalStatusSetPaneVisible()`
- Text anzeigen / lesen
 - `SalStatusSetText()`
 - `SalStatusSetPaneText()`

■ Pane Icon

- `SalStatusSetPanelImage()`

■ Erzeugen einer Progress Bar

- `SalStatusCreateProgressBar()`
- Arbeiten mit der Progress Bar erfolgt mit der API der Progress Bar

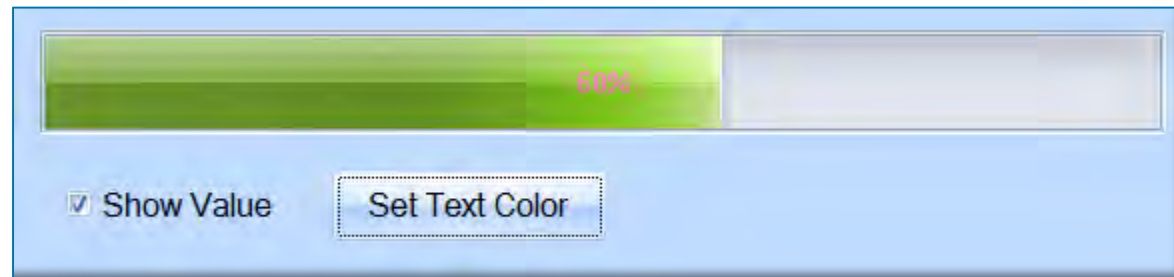
■ Tooltip der Pane setzen

- `SalStatusSetPaneTooltipText()`

Team Developer 7.0



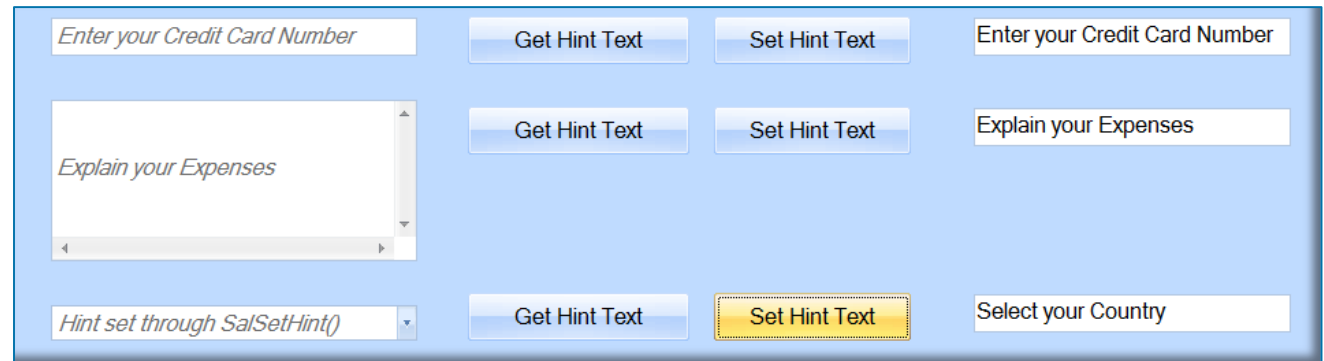
- Progress Bar Erweiterungen
 - Neue API Funktionen zur Anzeige von Werten und Textfarbe:
 - `SalMeterSetShowPercent()`
 - `SalMeterSetTextColor()`



Team Developer 7.0



- Erzeugen und Lesen von Hinweistexten
 - Daten Felder
 - Multiline Felder
 - Combo Boxen
- `sHint = SalGetHintText(hWnd)`
- `SalSetHintText(hWnd, sHint)`



Team Developer 7.0

- Picture Control Erweiterungen
 - Unterstützung von EMF Vektor Grafiken
 - EMF = Enhanced Windows Metafile
 - Vektor Grafiken werden häufig im Fertigungsbereich eingesetzt
 - Previews eines EMFs werden nun im Picture Control dargestellt

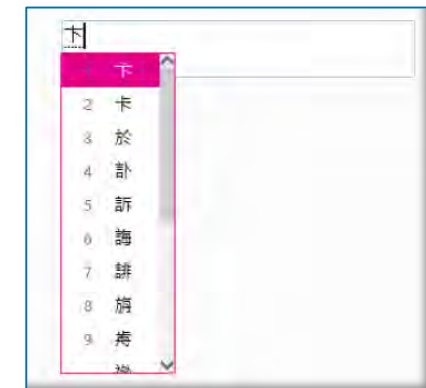
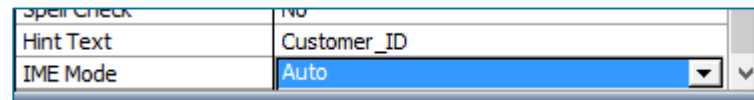
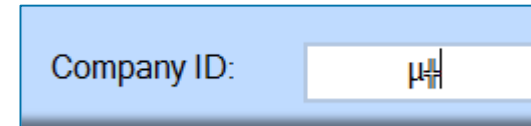


Team Developer 7.0



- IME Erweiterungen

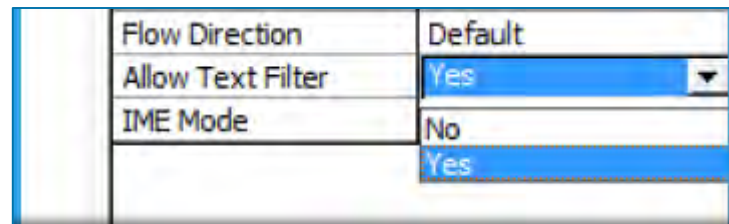
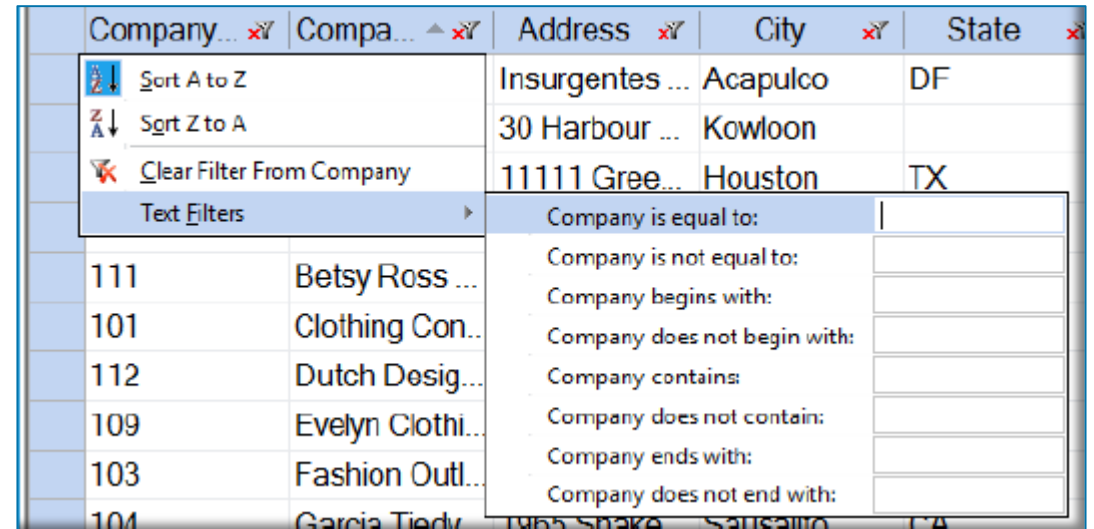
- IME Editor für Sprachen wie
 - Chinesisch / Japanisch / Arabisch / Kyrillisch
 - Auch bekannt als ‘Sonderzeichen’
- TD unterstützt nun diese grafischen Symbole für
 - Datenfelder, Multiline Felder, Combo Boxen, Grid & Table Spalten usw.
 - Attribute Inspector: Auto, Active, Inactive
 - Über PopUp Menu aktivieren / deaktivieren



Team Developer 7.0



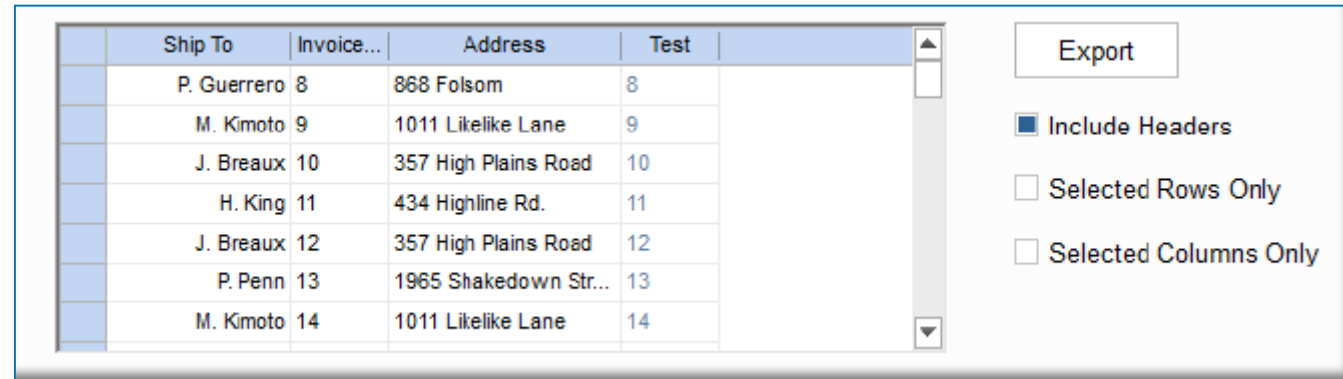
- Grid Erweiterungen
 - Daten filtern innerhalb der Anwendung
 - Auch für Win32 & Win64
 - Neue Eigenschaften im Attribute Inspector
 - Spracheinstellung der Systemmenüs basierend auf der Spracheinstellung des Computers



Team Developer 7.0

■ Grid Erweiterungen

- Daten Export
 - Einschränkung auf n-Zeilen
 - Einschränkung der n-Spalten
 - Mit / ohne Kopfzeile
- Excel Ausgabe nun im *.xlsx Format

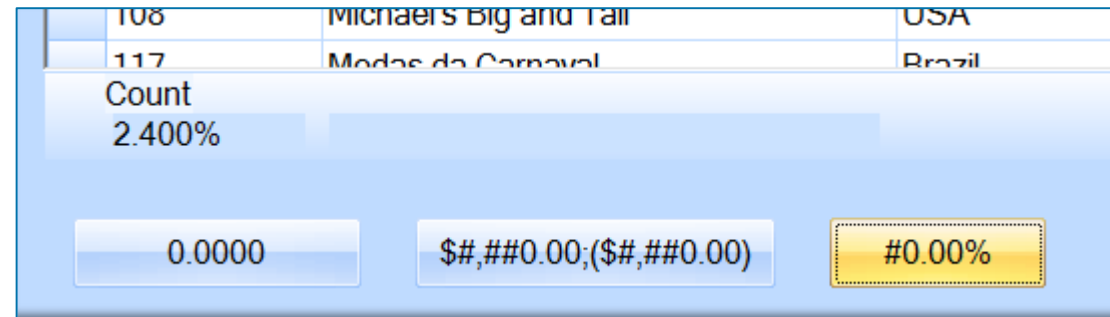


```
• If NOT cbIncludeHeaders
  ◊ Set nExportOptions = EXPORT__HideHeaders
  ◊ Set sStatusMessage = "EXPORT__HideHeaders"
• If cbRows
  ◊ Set nExportOptions = nExportOptions | EXPORT__SelectedRows
  ◊ Set sStatusMessage = sStatusMessage || "EXPORT__SelectedRows"
• If cbColumns
  ◊ Set nExportOptions = nExportOptions | EXPORT__SelectedCols
  ◊ Set sStatusMessage = sStatusMessage || "EXPORT__SelectedCols"
◊ Call SalGridDataExportEx ( tblPrdkt, "test-export.xlsx", Format__EXCEL, FALSE, nExportOptions)
```

Team Developer 7.0



- Grid Erweiterungen
 - Summary Bar Format Einstellung
 - `SalGridSetColumnFmtPicture()`
 - Datum Formate
 - Zahlen Formate



Team Developer 7.0



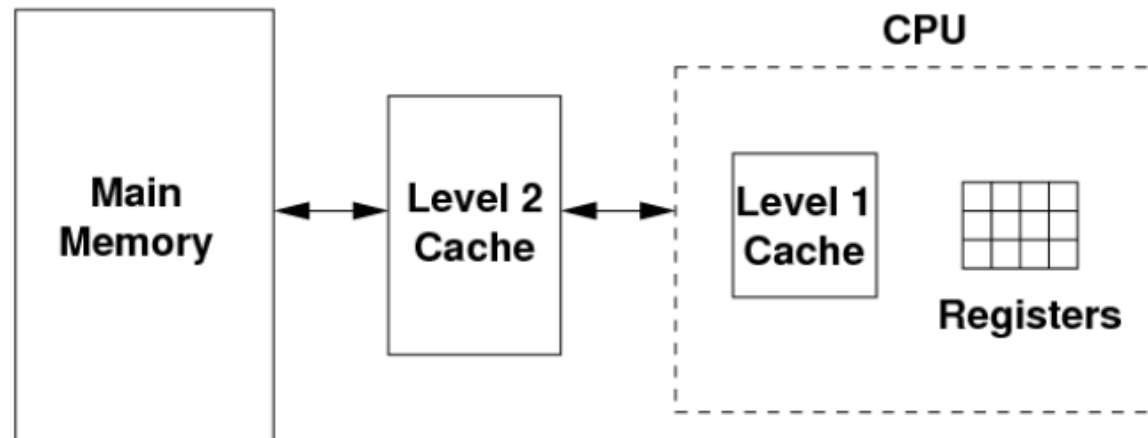
- Team Developer Win64
 - Größte Neuerung im Team Developer 7.0!
 - Seit mehr als 10 Jahren werden die CPUs von Intel/AMD mit der 64-Bit Architektur gefertigt und von den Computerherstellern verbaut
 - Jedes Betriebssystem seit XP und Server 2003 besitzt eine 64-Bit Architektur
 - Weltweit werden 92,8% aller neuen PCs, die mit Windows ausgeliefert werden, mit einem 64-Bit Betriebssystemen ausgeliefert

Microsoft Outlook (32 Bit)	0,2%	160,6 MB	0,1 MB/s	0 MBit/s
Microsoft PowerPoint (32 Bit)	0,5%	57,0 MB	0 MB/s	0 MBit/s
MyFirstApp64.exe	0,7%	9,0 MB	0 MB/s	0 MBit/s
Skype for Business (32 Bit)	0%	59,4 MB	0 MB/s	0 MBit/s

Team Developer 7.0



- Team Developer Win64
 - Was bedeutet 64-Bit?
 - Prozessoren, die 64-Bit unterstützen, nutzen ein 64-Bit Register
 - Das bedeutet: Es kann mehr Speicher (RAM) verwaltet werden
 - Die Grenze bei 32-Bit liegt bei 4GB (theoretischer Wert)



Team Developer 7.0



- Team Developer Win64
 - Wow64 = “Windows on Windows 64”
 - Stellt vereinfacht einen Emulationslayer für 32-Bit Anwendungen zur Verfügung
 - Übersetzen von System-Calls, die von 32-Bit Anwendungen in 64-Bit ausgeführt werden und gibt 32-Bit Strukturen an die Anwendung zurück
 - 32-Bit System Bibliotheken sind im SysWow64 Ordner!
 - 32-Bit Anwendungen werden in ‘Program Files (x86)’ installiert

Team Developer 7.0

- Warum TD Win64?
 - Bessere Performance
 - Native Ausführung im 64-Bit Betriebssystem
 - Schnellere Ausführung der Systemcalls (fastcall <> stdcall)
 - Der 64-Bit Compiler kann die zusätzlichen CPU Register nutzen
 - Es kann mehr Speicher für die Anwendung bereitgestellt werden



Team Developer 7.0



- Warum TD Win64?
 - 64-Bit Datenbank Treiber können genutzt werden
 - 64-Bit externe Bibliotheken können genutzt werden
 - Für die Zukunft:
 - Was plant Microsoft mit Wow64?
 - Wird 32-Bit eingestellt?

Team Developer 7.0

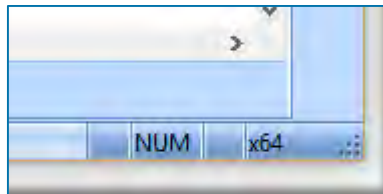
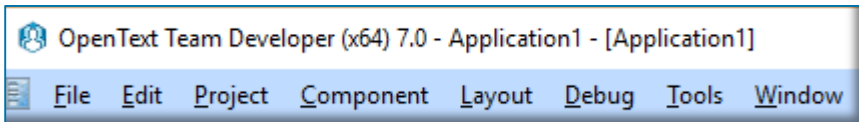


- Der Unterschied:
 - Team Developer wird mit zwei Installern ausgeliefert: Einer für x86 und der zweite Installer für x64
 - Beide Versionen können auf einem PC installiert werden
 - Die zuletzt installierte Version entscheidet, welche IDE gestartet wird, wenn eine Sourcecode Datei im Explorer ausgewählt wird
 - Das 'Arbeitsverzeichnis' ist x64 oder x86 entsprechend der gewählten Version
 - IDE, RB und andere Tools des TD7.0 64-Bit sind native 64-Bit Prozesse
 - RB Templates sind binär kompatibel

Team Developer 7.0



- Der Unterschied:
 - Wie kann man die einzelnen IDEs auseinander halten?
 - Start Menu -> OpenText Gupta Team Developer (x64)
 - Titelleiste der IDE ist angepasst
 - Status Zeile zeigt Win32 <> x64
 - About - Dialog zeigt die aktuelle Bitness



Team Developer 7.0



- TD7.0 64-Bit wird mit einer SQLBase 64-Bit ausgeliefert
- Unterstützte Datenbank Router
 - SQLBase 12.0
 - SQLBase 11.7
 - SQLBase 11.6 (mit 11.7 Router)
 - Oracle 11g R2
 - Oracle 12c
 - Microsoft SQLServer 2008, 2012, 2014, 2016
 - SAL ASE 16 (Sybase)
 - OLEDB (64 Bit Version)
 - ODBC (64 BIT Version)

Team Developer 7.0



- Der Unterschied:
 - Registry
 - HKEY_CURRENT_USER_\Software\Gupta\SQLWindows(x86) 7.0
 - Program Data
 - Identisches Verzeichnis für x64 und x86 Versionen des TD

Team Developer 7.0



- Wie wird migriert:
 - In der Weise, wie TD die Outline binär speichert, ist es nicht möglich, *.app files, die mit einer 32-Bit Version erzeugt wurden, in der 64-Bit Version zu laden
 - Ausschließlich 'text mode' Outlines können in 32-Bit und 64-Bit bearbeitet werden
 - Die IDE wird die Sourcen automatisch in den Textmode migrieren und öffnen:
 - Es wird eine entsprechende Warnung ausgegeben, da der original File modifiziert wird
 - Alle Änderungen des Files werden in einen <orig_file>.bak gespeichert

Team Developer 7.0



- Wie wird migriert:
 - Dynalibs
 - Müssen re-compiliert werden, um in eine 64-Bit Anwendung gelinkt zu werden
 - **Achtung:** Überprüfen Sie die Datentypen für die Nutzung von externen Funktionen
 - Beispiel: Anwendungen nutzen Win32 API, um Menüs zu generieren
 - In MSDN wird beschrieben, dass diese Funktionen INT Typen nutzen; TD Anwendungen nutzen WORD Typen
 - Diese Aufrufe laufen problemlos in 32-Bit, aber funktionieren nicht in 64-Bit

Team Developer 7.0



- Wie wird migriert:
 - Daten Typen
 - Stellen Sie sicher, dass DWORD_PTR als 64-Bit Pointers eingesetzt werden, da DWORD nicht ausreichend ist, wenn C-Structs eingesetzt werden, um UDVs an externe Bibliotheken (DLLs) zu übergeben
 - ActiveX
 - 32-Bit ActiveX Controls werden **nicht** in 64-Bit Anwendungen laufen. Es muss die entsprechende 64-Bit Version eingebunden werden!

Team Developer 7.0



- SAL API:
 - BOOL SallsX64()
 - Mit der API kann ermittelt werden, ob die Anwendung in 32-Bit oder 64-Bit Version läuft. Dies ist sinnvoll, wenn Anwendungen / Bibliotheken in 32-Bit und 64-Bit zur Verfügung stehen sollen.
 - Beispiel: Sie haben unterschiedliche 'Binary Offsets' und benötigen die genaue Größe der Strukturen, um die Daten einer externen Funktion richtig zu lesen
 - Gibt TRUE zurück, wenn die Anwendung als 64-Bit Anwendung läuft

```
◆ If SallsX64( )  
  ◇ Call SalShowWindow( bkgd3 )  
◆ Else  
  ◇ Call SalHideWindow( bkgd3 )
```


Team Developer 7.0



- Warum REST Web Services?
 - Es gibt eine Vielzahl von Web-Services, die auf REST Technologie basieren:
 - Google, Amazon
 - OpenText Produkte bieten oft einen REST WS an

Team Developer 7.0



- REST Web Services
 - REST = Representational State Transfer
 - Plattform- und Sprachunabhängig
 - Stateless Client / Server Kommunikation
 - Einfaches HTTP Protokoll
 - Alles was mit SOAP WS gemacht werden kann, kann auch mit REST WS gemacht werden.

Team Developer 7.0



- REST ist einfach zu implementieren
 - Beispiel: SOAP Anfrage:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:body pb="http://www.acme.com/phonebook">
    <pb:GetUserDetails>
      <pb:UserID>12345</pb:UserID>
    </pb:GetUserDetails>
  </soap:Body>
</soap:Envelope>
```

- REST Anfrage:

```
http://www.acme.com/phonebook/UserDetails/12345
http://www.acme.com/phonebook/UserDetails?firstName=John&lastName=Doe
```

Team Developer 7.0



- Antwort einer REST Anfrage
 - Ist nicht an XML gebunden (so wie in SOAP)
- HTTP Protokoll:
 - Stateless
 - Kommunikation zwischen Server (Host) und Client:
 - Anfrage (Request)
 - Antwort (Response)
 - URL basierende Syntax:

```
http://www.acme.com/phonebook/UserDetails/12345
```

```
http://www.acme.com/phonebook/UserDetails?firstName=John&lastName=Doe
```

Team Developer 7.0



- HTTP Sprachworte:

- GET
- POST
- PUT
- DELETE
- HEAD
- TRACE
- OPTIONS

- HTTP Status Code:

- 1xx: Information
- 2xx: Erfolgreich z.B. 200 = Ok
- 3xx: Umgeleitet
- 4xx: Client Fehler z.B. 400 = bad request
- 5xx: Server Fehler z.B. 503 = Service unavailable

Team Developer 7.0



- Team Developer Implementation
 - JSON (JavaScript Object Notation)
 - Kurz & schnell
 - Einfacher als XML (Parsing)
 - Einfacher zu verstehen (selbst beschreibend)
 - Sprachunabhängig: Das Format ist lesbarer Text
 - Gleiche Syntax wie in JavaScript für die Erzeugung von Objekten

Team Developer 7.0



- Team Developer Implementation:

- Syntax

- Daten bestehen aus Name/Werte - Paaren

```
"COMPANY_NAME": "Maui Mu-Mus"
```

- Daten werden durch Kommata getrennt

```
{"COMPANY_ID": 101, "COMPANY_NAME": "Clothing Connoisseurs"}
```

- Geschweifte Klammern beschreiben Objekte

```
{"COMPANY_ID": 101, "COMPANY_NAME": "Clothing Connoisseurs"}
```

- Recheckige Klammer beinhalten Arrays

```
var IslandCompany = [  
    {"COMPANY_ID": 101, "COMPANY_NAME": "Clothing Connoisseurs"},  
    {"COMPANY_ID": 102, "COMPANY_NAME": "Maui Mu-Mus"}  
];
```

Team Developer 7.0



- REST API

- bOk = SaJSONDeserializeUDV(String sObjectName, String URL, String UserName, String Password, Receive String strError)
 - sObjectName: UDV oder Array name der JSON Daten
 - URL: Eine URL oder eine JSON Zeichenkette
 - Username: Authentifizierung der Anwenders
 - Password: Passwort des Anwenders
 - strError: Fehlerbeschreibung, wenn die Funktion FALSE zurückgibt

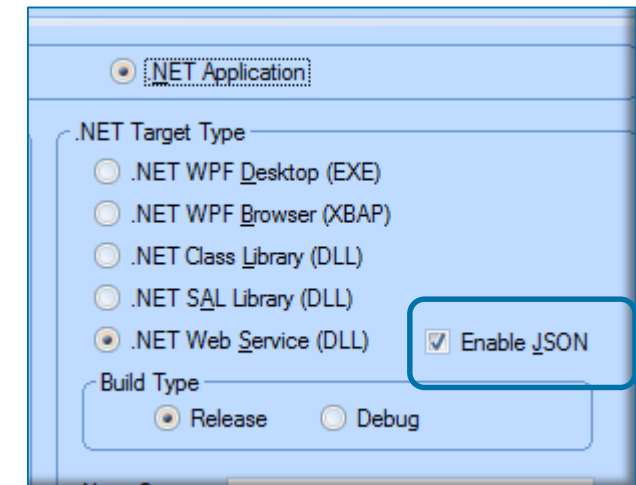
- REST API

- bOk = SalJSONSerializeUDV (String sObjectName, String URL, Number HTTP_VERB, Receive String sResult, String sUserName, String sPassword, Receive String strError)
 - sObjectName: UDV oder Array Name der JSON Daten
 - URL: Eine URL oder eine JSON Zeichenkette
 - HTTP_VERB: Konstanten HTTP_PUT, HTTP_HOST, HTTP_DELETE
 - sResult: Rückgabe des Ergebnisses vom Server
 - sUsername: Authentifizierung der Anwenders
 - sPassword: Passwort des Anwenders
 - strError: Fehlerbeschreibung, wenn die Funktion FALSE zurückgibt

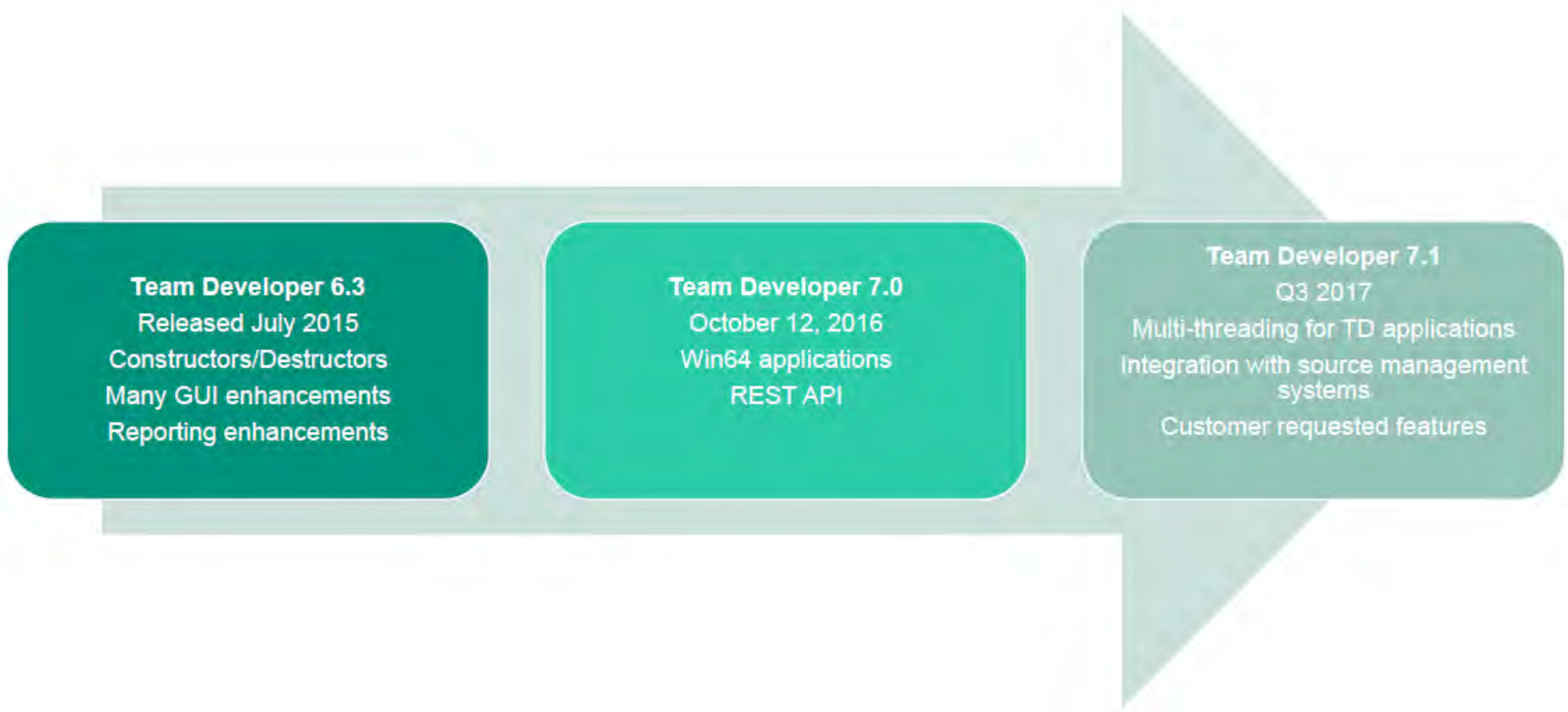
Team Developer 7.0



- Erzeugen eines JSON/REST Web-Services
 - Neue Option, um JSON WS zu generieren
 - Das Veröffentlichen erfolgt wie bei TD .NET WS
 - SOAP XML **und** JSON wird implementiert
- Kommunikation mit dem TD .NET JSON Service
 - Nur POST wird unterstützt
 - Der HTTP Header benötigt den Typ: application/json



Team Developer Roadmap





OPENTEXT™

www.opentext.com



twitter.com/opentext



facebook.com/opentext



linkedin.com/company/opentext