



DataFlex Entwickler Tag 2019

# Project NextGen

## Harm Wibier

# DataFlex 2020

- › 64-bit capable
  - › Build 64-bit programs
  - › Use 64-bit components
  - › Still supports 32-bit..
- › Fully Unicode
  - › No more code-pages!
  - › Support multiple languages in a single application
- › Work started 2,5 years ago, planning even earlier
  - › Dedicated resources were hired







DataFlex Entwickler Tag 2019

64-bit

# 64-bit in the industry

- › 64-bit architectures
  - › Process larger chunks of data at a time
  - › Address more memory (>4GB)
- › First 64-bit version of windows available since 1999
- › Industry conversion is slow due to
  - › Minimal gain for average applications
  - › Support for 32-bit applications on 64-bit operating systems



# Why do I need 64-bit?

- › Communicate with other 64-bit software
- › To be more competitive
- › Performance gain
- › Some server environments 64-bit is required
- › Simply looks & sells better!



# Software development in 64-bit?

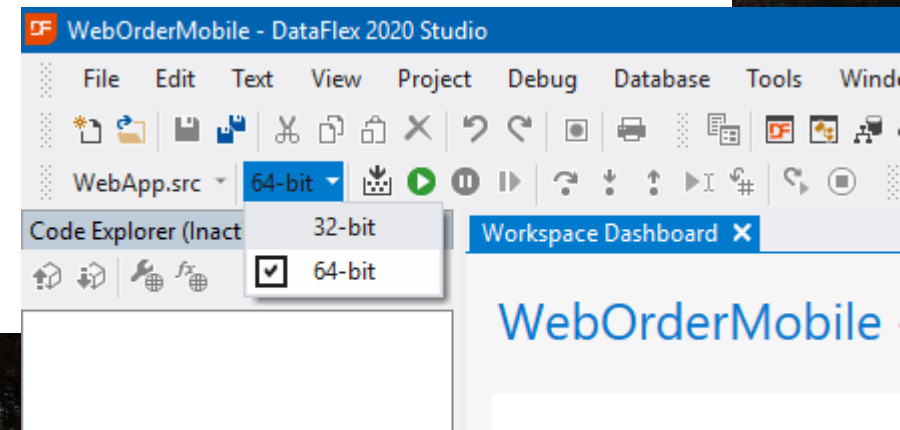
- › Memory pointers become twice as large
- › Some types change their size
  - › LongPtr
- › Integer usually stays 32-bit
  - › Can't store memory addresses any more!

# DataFlex 64-bit

- › Conversion of our C codebase
  - › Compiler, runtime, webapp server, studio, ..
- › New linker
- › Language changes to support 64-bit types
- › Replaced external\_function implementation
  - › Changes in calling conventions
  - › Deprecation of techniques used in 64-bit C compiler
- › Lots of package changes
  - › Usage of integer for pointer values
  - › To match windows 64-bit API's

# 64-bit capable

- › You will choose per project between 64-bit and 32-bit
- › We expect 32-bit to be around for a while
  - › A 64-bit application cannot use 32-bit DLLs
    - › This includes COM components
    - › All third party components you use need to be 64-bit
  - › You need time to migrate your code
- › New projects will be 64-bit by default

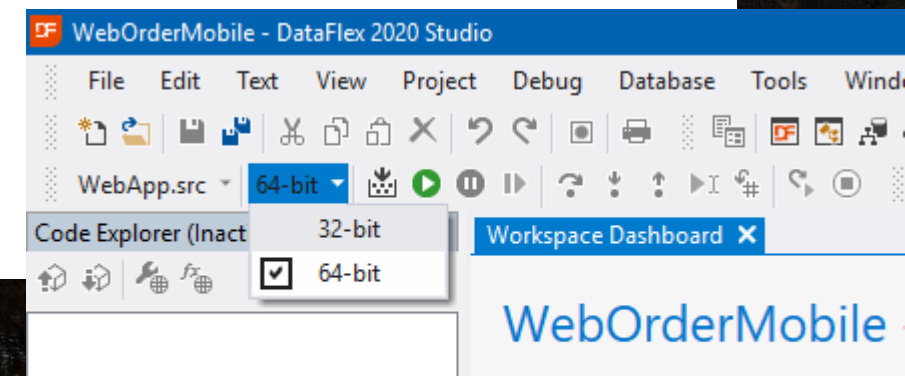
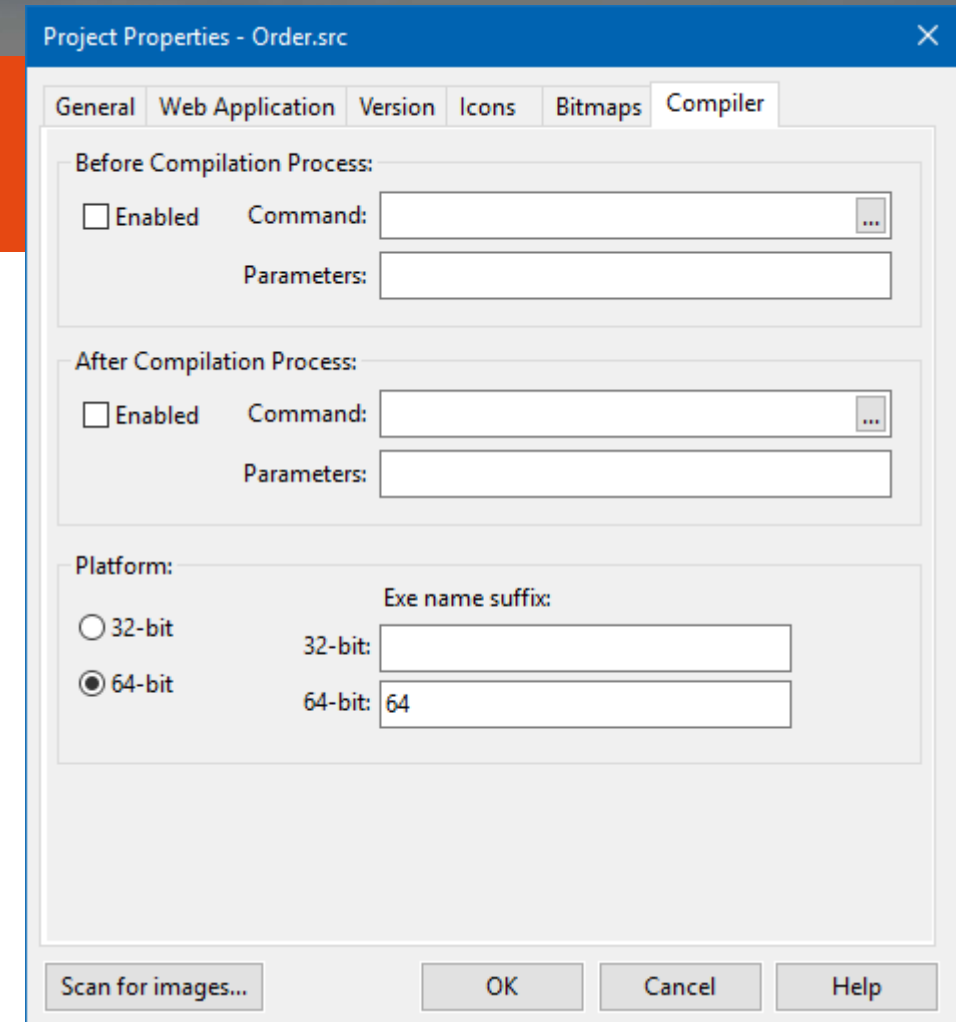




# Demo..

# Compiling for 64-bit

- › In the studio
  - › Choose in the studio toolbar
  - › Stored as a project setting
- › Configure .exe suffix
  - › Added to the .exe name
  - › Stored as project setting
- › Outside the studio
  - › Start DfComp from either Bin or Bin64



# Language changes

- › New LongPtr type
  - › Integer type that is the same size of a pointer
    - › 32-bit on 32-bit and 64-bit on 64-bit
- › Integer stays 32-bit
- › Pointer is now an Address
  - › Used to be an integer
- › Handle becomes LongPtr
- › New compiler switch

```
#REPLACE Pointer Integer  
#REPLACE Handle Integer
```



```
#REPLACE Pointer Address  
#REPLACE Handle LongPtr
```

```
#IFDEF IS$WIN64
```

```
#ELSE
```

```
#ENDIF
```



# Pointers & Address changes

- › Pointer is the recommended type for working with memory pointers
- › Pointer is now a replace for the Address type
- › Special behaviors when converting strings to addresses are now removed
  - › Use new PointerToString or MemCopy functions instead

# Package changes

- › Various Integer to LongPtr changes
  - › External API's
  - › Window messages
- › Several Integer to Pointer changes
  - › Invalid usage of Integer
    - › Bad habits since the beginning of DataFlex
- › Changes from Address into Pointer
  - › For consistency
- › Struct alignment changes
  - › C compiler aligns struct members
    - › <https://fresh2refresh.com/c-programming/c-structure-padding/>

# Demo..





Unicode

# Unicode?

- › Non-Unicode programs can only handle a single language
  - › Only 255 different characters can be used
    - › Codepage determines which language
    - › Conversion between codepages is lossy
  - › ANSI & OEM are encodings using this principle
- › Unicode programs can mix languages
  - › 1.114.112 different characters can be used
  - › UTF-8, UTF-16, UTF-32 and UCS-2 are Unicode encodings



# Character encoding

## › ASCII

1	0	0	0	1	0	0
---	---	---	---	---	---	---

- › 7 bits per character (127 characters)
- › Latin letters, numbers, reading symbols

## › ANSI

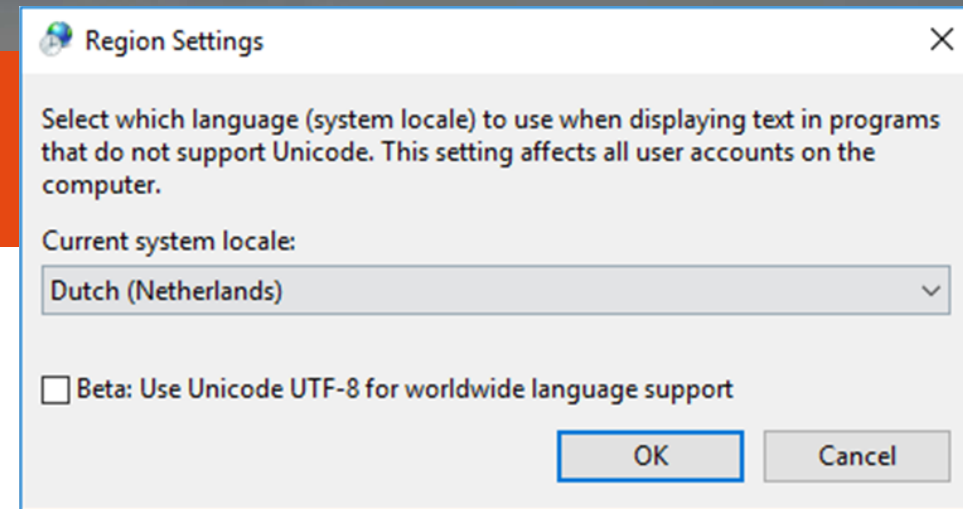
1	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

- › 8 bits per character (255 characters)
- › First 127 characters compatible with ASCII
- › Codepage determines the other 128 characters

## › OEM

1	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

- › 8 bits per character (255 characters)
- › First 127 characters compatible with ASCII
- › Codepage determines the other 128 characters





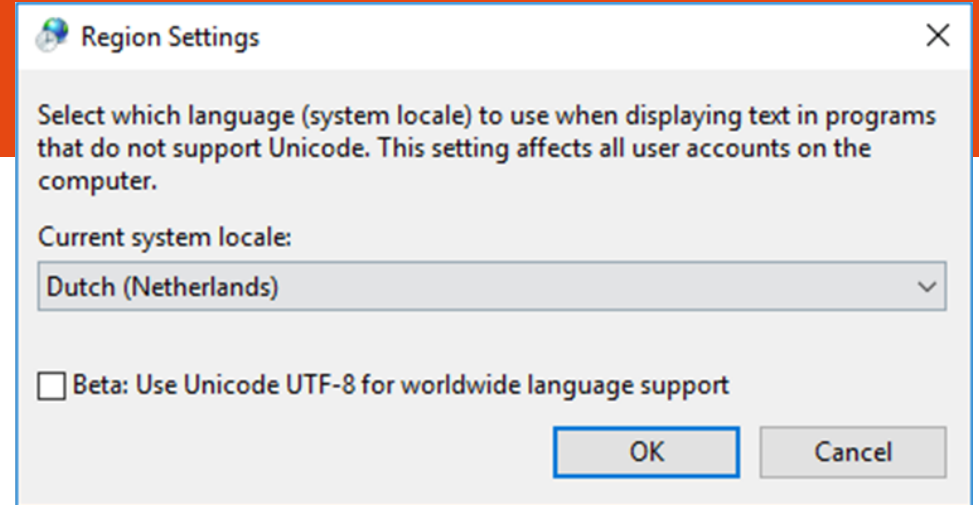
# Unicode character encodings

- › Unicode 1.0
  - › 16-bit (65.536 characters)
    - › UCS-2
      - › Fixed length encoding (2 bytes per character)
      - › Obsolete subset of UTF-16
- › Unicode 2.0
  - › 21-bit (1.114.112 characters)
    - › UTF-8
      - › Variable length encoding (1 byte per character up to 4 bytes)
    - › UTF-16
      - › Variable length encoding (2 bytes per character up to 4 bytes)
      - › UTF-16BE vs UTF-16LE
      - › Optimal for Asian texts (but not for XML / HTML)
    - › UTF-32
      - › Fixed length encoding (4 bytes per character)

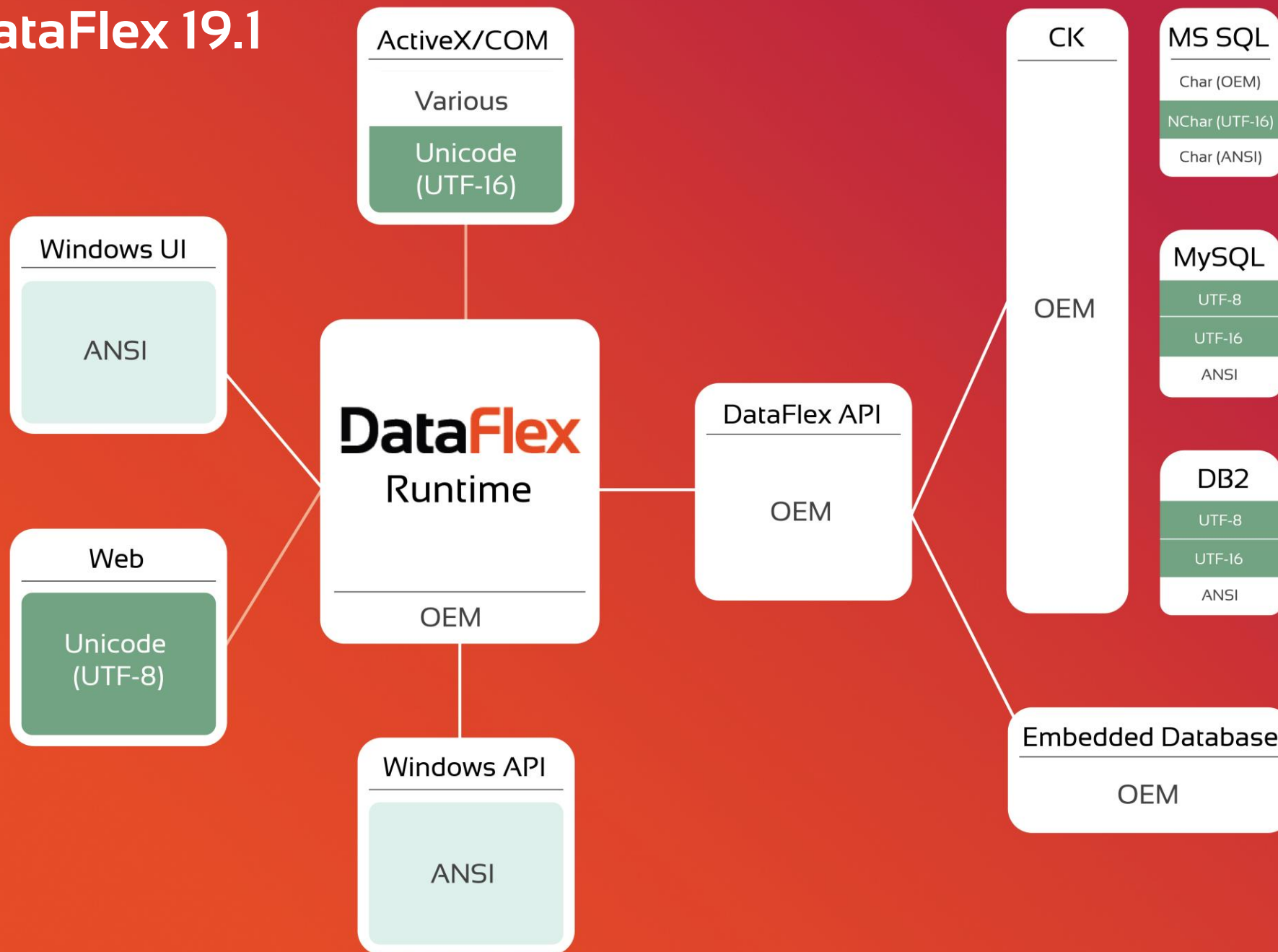


# Windows

- › Started with ANSI (8-bit)
  - › Supports OEM codepages
- › Moved to UCS-2 (16-bit)
  - › New WideChar API's (double byte)
  - › Still supports ANSI API's
- › Moved to UTF-16 (16-bit or more)
  - › Changed their double byte API's
  - › Still supports ANSI API's



# DataFlex 19.1

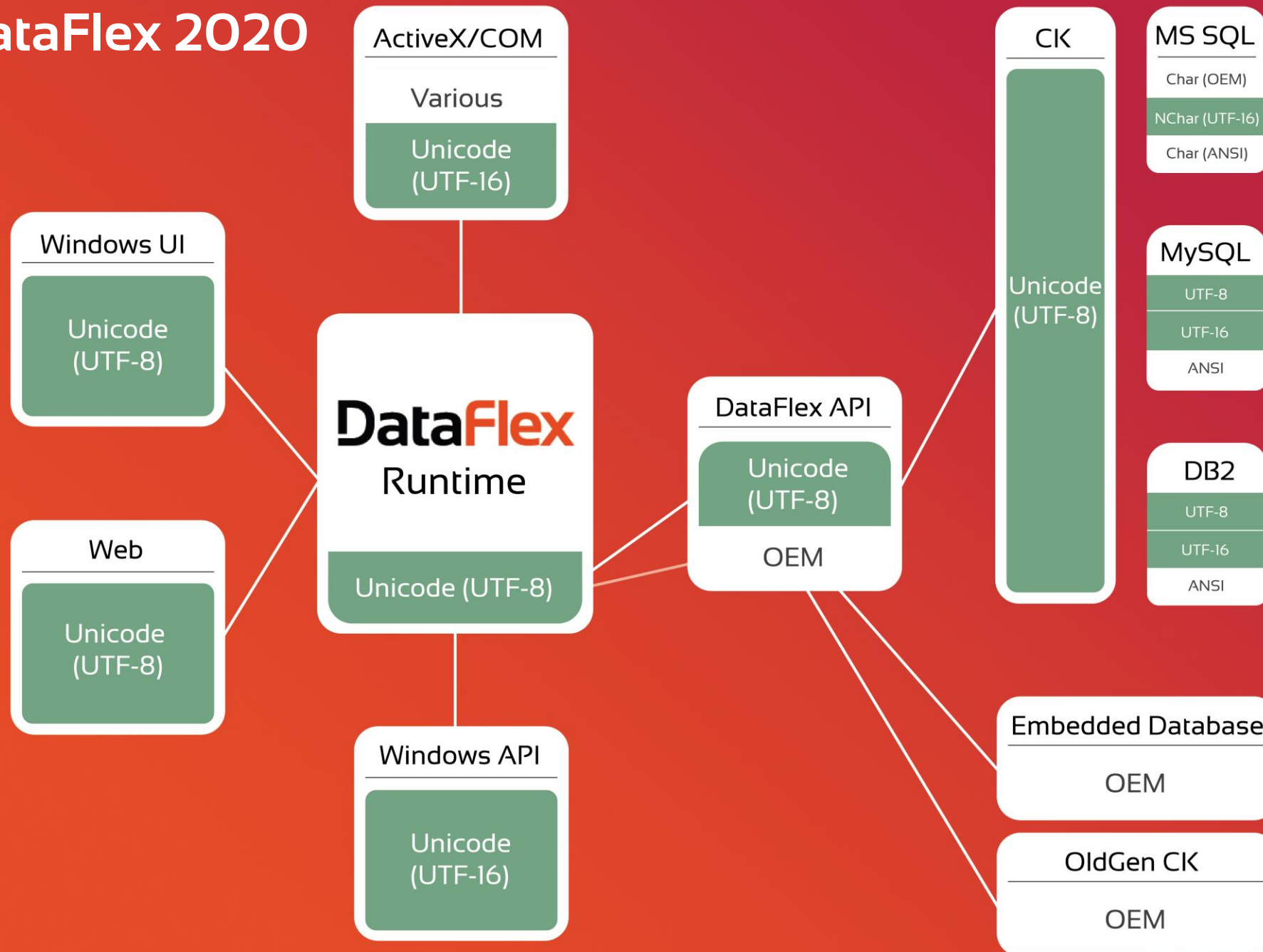




# Unicode in DataFlex

- › Strings will be UTF-8
  - › Best backwards compatibility
  - › Best for WEB
  - › Best for western languages
- › Sources will be stored as UTF-8 with BOM
  - › Non-ASCII characters allowed in string literals and comments
  - › Sources without BOM will be interpreted as OEM
- › DataFlex will talk to the wide Windows API's
  - › New WString type for automatic conversions to UTF-16

# DataFlex 2020



# Unicode databases

- › SQL will be the expected database
  - › Connectivity kits for MS SQL, DB2 and ODBC will support Unicode
    - › **MS SQL:** Use NVarChar / NChar fields
    - › **MySQL:** Use character set utf8mb4 on fields
    - › **DB2:** Use codeset UTF-8 on database
- › Legacy drivers
  - › A compatibility layer will convert data to OEM (losing special characters)
    - › **Embedded database:** No Unicode but compatibility with OldGen
    - › **Btrieve / Pervasive**
- › Third party drivers
  - › Will receive technical specifications for conversion soon



# Language changes

- › **Length(sString)** will return the number of characters
- › **SizeOfString(sString)** will return the number of bytes
- › **Pos, Mid, Left** and others will work with characters

# Demo..

# WString type

- › New string type in the DataFlex language
  - › Automatic conversion to UTF-16 when moving strings into wstring
  - › Used for calling external API's that work with UTF-8

```
External_Function PathFileExists "PathFileExistsW" shlwapi.dll ;  
    WString wsPath ;  
    Returns Integer
```

- › Only use when calling external API's!
  - › String is faster, string operations on WString convert back to String!
- › Late addition to the project



# Package changes

- › Switch to the Unicode windows API's
  - › The W versions of functions
  - › Compatibility wrappers are made
    - › Our packages and tools call wide functions directly
- › Use `SizeOfString` instead of `Length` to get string size
- › Don't use strings for binary data

# Environment

- › Deployment
  - › All tools will be 64-bit (Studio, DB Explorer, ..)
    - › Also builds and debugs 32-bit applications
  - › WebApp Server will be 64-bit
    - › Also runs 32-bit applications
  - › Client installer will both 32-bit and 64-bit
    - › Will work on 32-bit only machines
- › Every application will be working with Unicode
- › The DataFlex Reports team is working hard on a Unicode and 64-bits version
- › WinPrint2 will be 64-bit and Unicode

A panoramic view of a city skyline at dusk. In the foreground, a river flows from the bottom left towards the center, with a suspension bridge crossing it on the left. A green promenade with trees and people runs along the riverbank. The middle ground is filled with a dense urban landscape, including a prominent tall, dark, modern skyscraper on the left, a historic church spire with a cross, and various other buildings. The background shows a hazy cityscape under a cloudy sky. The text "Technology Preview" is overlaid in the center in a bold, orange font.

# Technology Preview



# Technology Preview

- › First public release of DataFlex 2020
- › Pre-alpha version
  - › We need feedback:
    - › What do you run into while converting your application?
- › Limitations
  - › Preliminary connectivity kits
    - › Only store Unicode data with MS SQL with NChar / NVarChar
    - › Other databases are completely untested
- › Available later this week!

# Planning

- › Finish conversion of the connectivity kits
- › Process feedback from technology preview
- › Convert the samples to MS SQL



DataFlex Entwickler Tag 2019

**Vielen Dank für Ihre Aufmerksamkeit!**  
**Haben Sie Fragen?**