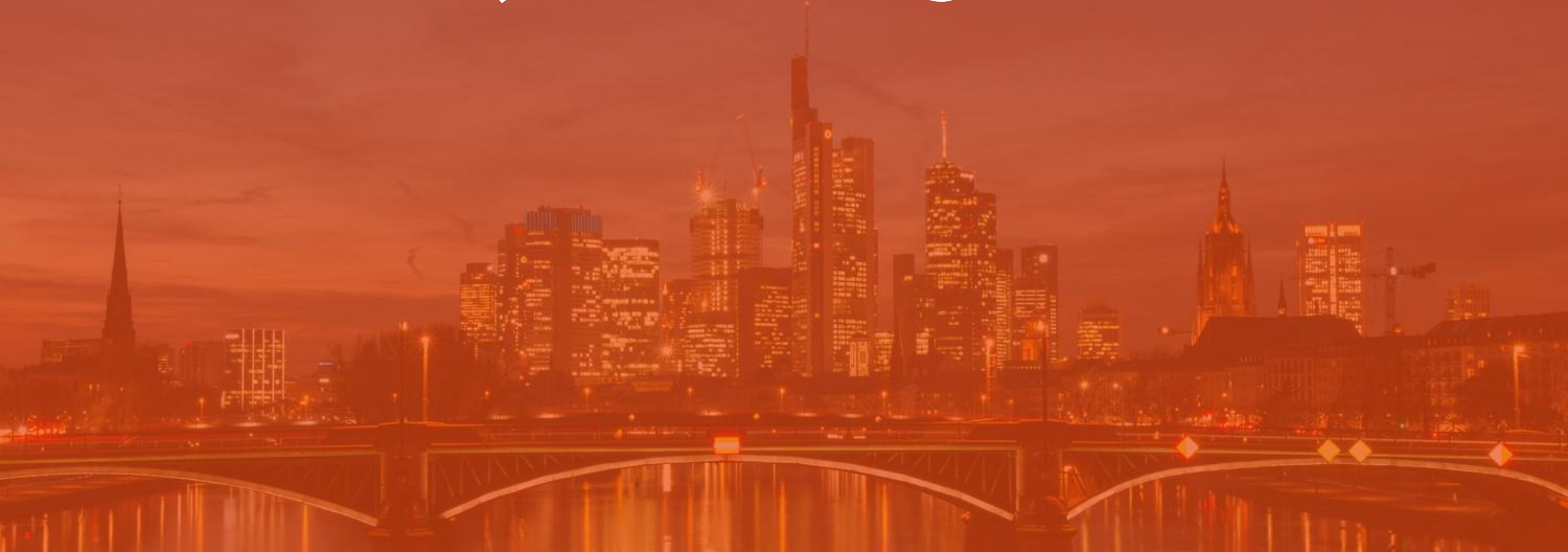# DataFlex Entwickler Tag 2017

Harm Wibier

DataFlex

DataFlex 19
SQL Server Integration

Long-term goals

- Make SQL the default for DataFlex applications
  - Replace the embedded database
  - Long term goal over multiple releases
  - MS SQL is the preferred vendor
    - Alternatives are available and supported
- First step is improving the SQL Experience

Goals for DataFlex 19.0

- Simplify usage of SQL databases
  - Bring connections from the driver into the studio
  - Improved login process
  - Improved convert, connect, reconnect and repair process

- Improve usability of existing databases
  - Client-only indexes
  - Read-only SQL mode
  - Clustered indexes

SQL in DataFlex
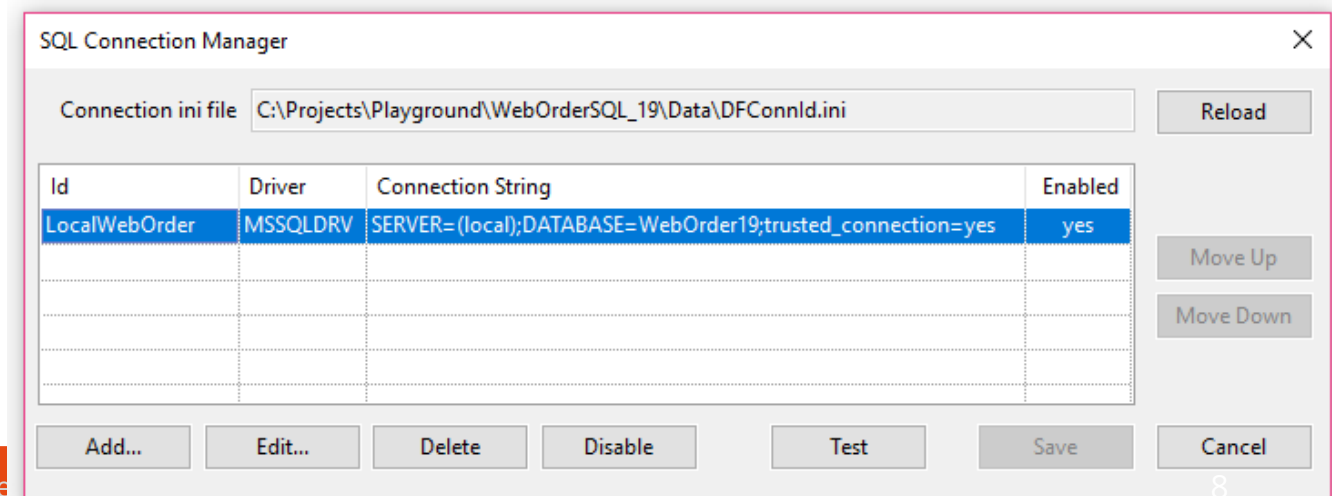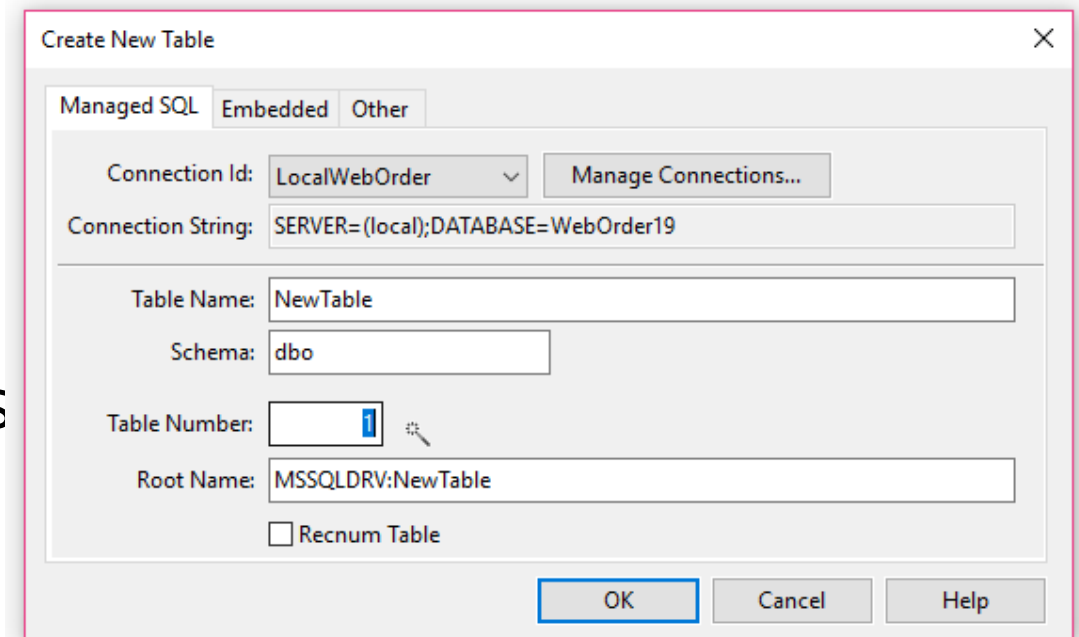
SQL in DataFlex

- Driver translating DataFlex database access into SQL access
  - No application changes compared to embedded
    - Open, Find, Clear, Save_Record operations all work
  - Connectivity kit translates database API calls
    - Intermediate file (.int) configures driver per table
    - Kits available for MS SQL, Pervasive SQL, Generic ODBC (preconfigured for MySQL and Oracle)

- Embedded SQL provides additional method for accessing data

Connections

- Establishing a connection is how you access SQL data
- This is done with a connection string
- A connection string defines
  - Server
  - Database name
  - Login credentials
  - Possibly more
- You connect by logging in to your database
- Once logged in tables can be accessed

Managed connections

- ● Managed in the studio
  - ○ Stored in DFConnId.ini
  - ○ Managed by cConnection class
  - ○ Encrypted passwords
  - ○ Multiple connections possible

Demo..

- Enough with the slides already..

Indexes DataFlex vs SQL

- SQL
  - Primary and Foreign Keys
  - Unique indexes? We don't need no stinkin' unique indexes!
  - We don't need no stinkin' indexes at all...
- DataFlex
  - Relationships (hard or soft)
  - Unique indexes
  - Indexes for everything you want to find or sort by
- Notice a pattern? So much revolves around indexes (in other words, finding stuff)

Client & Server Only Indexes

- Server
  - Exists both on the server and in DataFlex
  - Must be unique
- Server Only
  - Exists at the server, but not in DataFlex
  - Compatibility with non DataFlex environments
- Client Only
  - Exists in DataFlex only (defined in .int)
  - Must be unique

Connect to existing database

- Use Connect/Repair wizard
  - Detects relations based on foreign keys
- Non-unique indexes
  - Added as server-only indexes
  - Client-only index will be added completed with PK

Demo..

- Enough with the slides already..

Convert existing database

- Use conversion wizard
  - Table number, names, indexes remain
  - Choose
    - Define PK's (yes)
    - Convert from recnum to standard (yes)
    - ANSI or OEM (ANSI)
  - Considerations
    - Overlaps / underlaps
    - Make a backup

WebApps

- You don't want the browser to show the SQL connection dialog
- It can't be shown on the server
- Encryption key is only known by your app

- Use the template to create your own SQLLogin.exe

Demo..

- Enough with the slides already..

Data Dictionary Filters

- Define SQL filter on data dictionary
- Offloads constraining to SQL Server
- Use syntax of SQL where clause
- Helper functions
  - SQLStrFileFieldName
  - SQLStrLike

```
Object oCustomer_DD is a Customer_DataDictionary
    Set pbUseDDSQLFilters to True
    Set psSQLFilter to "[Customer].[Name] LIKE '%Computer%'"
End_Object
```

Runtime indexes

- ● Create index without restructure

```
Create_Index
Set_Attribute DF_INDEX_NUMBER_SEGMENTS
Set_Attribute DF_INDEX_SEGMENT_FIELD
Set_Attribute DF_INDEX_SEGMENT_DIRECTION
Delete_Index
```

- ● Do not use structure_start & structure_end
- ● Great for business processes

Embedded SQL

- Run custom queries to easily access sets of data

```
Handle hoStmt hoConnect
String[][] aData

Get SQLConnectionId of ghoConnection "OrderData" to hoConnect
Get SQLOpen of hoConnect to hoStmt
Send SQLExecDirect of hoStmt "SELECT * FROM [Invt];"
Get SQLFetchResultsetValues of hoStmt to aData
Send SQLClose of hoStmt
Send SQLDisconnect of hoConnect
```

Connection switching

- Switch between databases
  - Use **RedirectConnectionId** to change connection at runtime
  - Keep original connection alive for fast switching
  - Structure must be the same
  - Your responsibility to clear file buffers
- Paves the way for multi-tenant applications
- Particularly useful for web applications

- Alternative method: **DF_DATABASE_DEFAULT_DATABASE**

Optimize for SQL

- Identity columns instead of auto-increment
- Client-only indexes for possible slow indexes
- Runtime indexes
- Use SQL filters on data dictionaries
- Use views

Vielen Dank für Ihre Aufmerksamkeit!

Haben Sie Fragen?

DataFlex