DataFlex 19.0 Overview and “Update Builds”

Stephen W. Meeley
DISD 2018
Dallas Texas, March 2018

DataFlex 18.x

Just in case you are not using the latest...

DataFlex 2014 - 18.0

- Rebranding
- Web
  - Version 2 of the Web Framework
  - Grid/List, Upload class, SuggestionForm
- Windows
  - Buttons with images, LinkButton, dbSuggestionForm
  - Optional modal views
- Studio
  - Table Viewer
  - Native SQL Type Modeling
- Data Dictionaries
  - SQL Filtering
- Client Web Services - nullable data type support
- Character encoding translation class

DataFlex 2015 - 18.1

- Web
  - Mobile framework!
  - Complex web properties (structs and arrays)
  - Previewer evolved into interactive designer
- Data Dictionaries
  - Big performance and memory improvements
  - Improved missing (deleted) record handling
- Connectivity
  - Identity column
  - GUID
  - Azure
- Studio - Improved debugger performance
- UChar Array improvements - tables, sequential files, String to UChar functions
- Many new struct and array functions

DataFlex 2016 - 18.2

- Primarily created to fix bugs and implement suggestions
- Web
  - Version 2 of the Mobile Web Framework
  - ParentCombo
  - Flat desktop theme
  - DD Debugger
- Windows
  - Windows 10 Support
  - Parentcombo, month calendar
- Studio
  - Pre/post compile processing
- Data Dictionaries
  - Smart Relates
- Extensive new set of DateTime functions

Why do we look back so much?

- Not everyone is using the latest version
- Even those who are may not be using all the latest features or use the latest version for all their work
- Tomorrow morning John will talk about some of the many changes that may have slipped past your notice, regardless of which version you are using

DataFlex 19.x

DataFlex 2017 - 19.0

- Connectivity
  - Managed Connections
  - Improve indexing support
- Windows Framework
  - Enhanced high DPI Support
  - Updated look
  - An MDI alternative style
- Studio, Database Explorer and Database Builder
  - More natural developing using SQL
  - Updated capabilities and look
- Web Framework
  - Server Web Properties
  - New Controls
- Web Application Server
  - Simplified deployment
  - 64-bit support
- Runtime – JSON Parser

DataFlex 2017 - 19.0

Connectivity

Connectivity Goals

- Fully integrate Connection IDs into Tools and Applications
- Improve indexing support
- Improve the flow of SQL-based application development
  - Primary Keys
  - Foreign Keys
  - Clustered Indexes
- Improve the database login process
  - Encrypted credentials
- Make the Convert, Connect, Reconnect process easier to use and more robust

- Maintain backwards compatibility!

Main Connectivity Changes

- Managed Connections
  - Provides an integrated approach for creating, connecting and maintaining SQL Databases
- New Convert, Connect, Repair Wizards
  - Makes the entire process of working with SQL Databases much easier and more powerful
- Improved Index Support
  - Support for partial SQL Indexes
  - Supports dynamic indexing for SQL
  - Watch “Let me help you find that…” video

- Using Managed SQL Connections – John Tuohy
- How to Update Your Applications with Managed SQL Connections - Me

Other Connectivity Features

- Environment management
  - Mimimum_Client_Version
  - Match_Client_Server_Version
- Updated default data type mapping
- Restructure control
  - All table changes allowed
  - Only .int file changes allowed
  - No changes allowed
- Fast database switching at runtime
  - RedirectConnectionID
- Support for SQL Server 2017

Windows Framework

Windows Framework

- Improved DPI Awareness
  - The key to making the UI look consistently good across a variety of high-DPI display settings

- Flat Style Icons
  - All of the menu and toolbar icons in our tools and all of the standard icons used in the Framework menu and toolbar are updated to the new style (and high DPI format)

- An alternative approach to MDI in Windows

Improved DPI Awareness

- DPI awareness is key to making the UI look consistently good across a variety of high-DPI display settings

- The font improvements in 17.x already took care of all the *text* in Windows controls, our controls and Codejock controls, but…
  - Codejock controls did not resize images (images in toolbars and menus did not change)
  - Codejock controls were not using complete DPI text scaling (font only, not overall spacing)

- Updating to a newer version improves the DPI awareness in the Codejock menus and toolbars. That’s a start…

Improved DPI Awareness

- There is also an issue with images
  - The images used in applications are all stored in a single 16x16 size.
  - Windows will attempt to virtualize and scale on its own, but this often results in images that are not the correct size or blurry

- Codejock and Windows expects you to provide multiple sized image in a icon file
  - It will then choose the best size for the current DPI
  - The goal is to always have the right size image to choose from – no scaling

- You can see DPI scaling issues in many Windows applications. This is a process of refinement

Improved DPI Awareness

- The Windows Framework implements the new Codejock capabilities
  - Automatically selects the best icon size for the DPI setting of the display when using resource icons (compiled in)

- We replaced all our simple, single-size icons and bitmaps with Icons that contain multiple sizes (16x16, 24x24 and 32x32)

- You will probably face the same decision about adding icon sizes
  - The good news is that if you only use are our icons; you're done

- Remember - if you don't want to use the DPI aware icons there is a property to turn off the new capabilities

Flat Style Icons

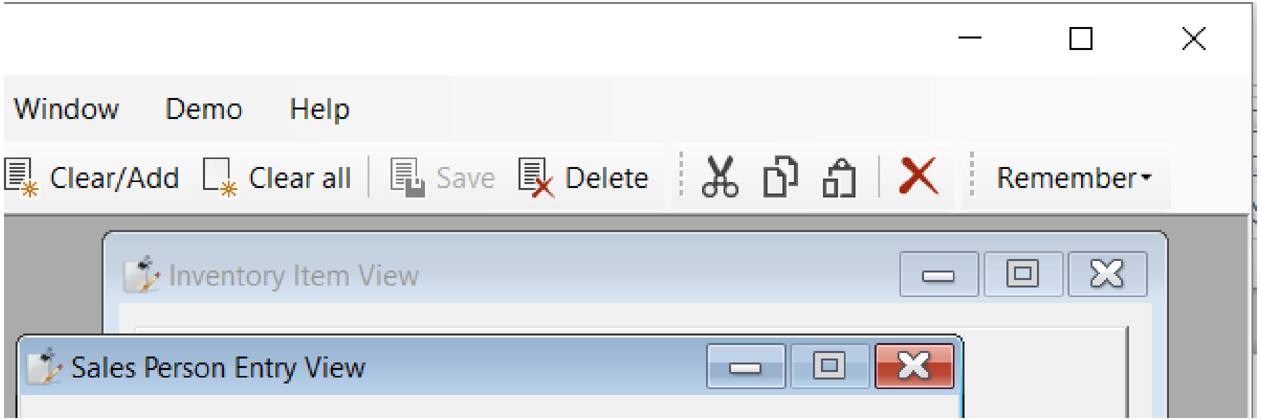
- We determined that updating the older style icons would be even better if we updated to newer, “flat style” icons.
- All of the standard icons used in the Windows Framework menu and toolbar have been updated and are located in the Bitmaps folder.
- If you wish to continue to use the old set of bitmaps and icons they are located in a subfolder (DataFlex 2016 Bitmaps) and you can simply copy them to your workspace Bitmaps folder so they will be found and used first.

A New Look for the Tools

- These changes also led to improvements in the Studio, Database Explorer and Database Builder
  - They all have a new look
  - Applied all of the DPI changes
  - Replaced images with multi-sized, flat icons
    - We have a lot of images!
  - Updated our theme
  - Refreshed Dashboard and Start Center looks

Traditional MDI Interface

- Microsoft doesn’t like MDI Interface so much...
  - In Windows 10 – Caption and System buttons are not painted correctly...

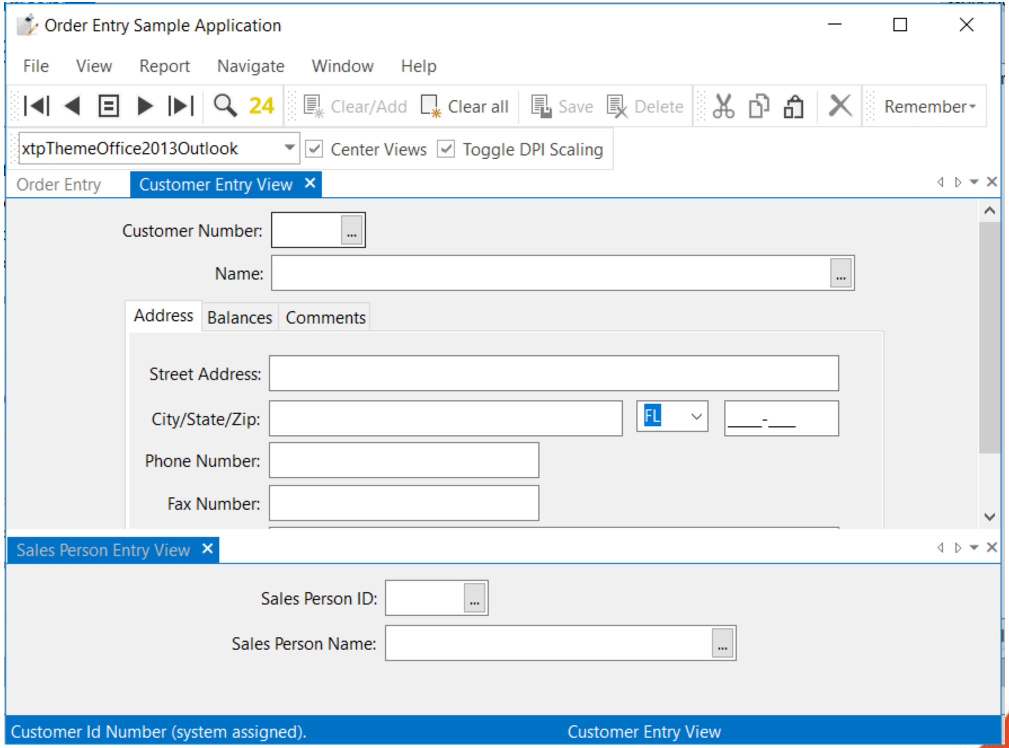


An Alternative MDI Style for Windows

- Microsoft has not done a good job of keeping the appearance of MDI dialogs up to date
  - Although this is a great and practical application style, Microsoft is moving away from this
  - It doesn’t look like this will change
  - This makes an application appear old fashioned
- Rather than fight this, we are supporting an alternative look for MDI style applications...

An Alternative MDI Style for Windows

- Tabbed Workspaces
  - Instead of using overlapping windows your views are placed in tabbed workspaces
  - Views can be split vertically or horizontally, allowing you to see multiple views at one time
  - Views are selected by selecting their tab
  - To work properly special sizing, centering and scrolling of views must be supported – it's all automatic



xtpThemeOffice2013Outlook ☑ Center Views ☑ Toggle DPI Scaling

City/State/Zip: FL

Customer Id Number (system assigned). Customer Entry View

An Alternative MDI Style for Windows

- The alternative MDI style uses a combination of Codejock Tab Workspaces and our scrolling panel class support

- Using the alternative MDI style requires code changes
  - Your main application must enable tab-workspaces
  - Each view must add a scrolling panel and scrolling container object
  - In addition, you may want to further customize your views to take full advantage of this new style

- This is an optional style

Database Explorer

- Log changes made to data
  - tablenumber, tablename, rowid, timestamp, network user, operation (create/edit/delete)
- Supports Managed Connections
- Provides more information about the backend structure
- Updated grids and lists throughout
- Improved filtering
- Modern layout, capabilities and feel

Web Framework

Server Web Properties

- Web properties were designed to configure everything that the client needs to know about (mainly the state of the user interface)
- Applications use web properties for all kinds of state data. This is data that is not needed on the client but needed for the page or session
- In 19.0, Server Web Properties provide a secure method to store application state data
- Lowers data consumption to improve performance on slow networks

- Harm will teach you all about this later (Spoiler alert: It’s really easy to do)

Web Framework Controls

- DataFlex 19.0 includes exciting additions to the control set for the Mobile / Touch style
  - Color Picker
  - List Menu
  - Floating Panel
  - Highlight Column
  - Accordion Control
  - Collapsible Group
  - Styled Checkbox and Radio
  - Swipe Buttons (in Lists) and Cards
- Harm has a session about how to use these new controls tomorrow morning

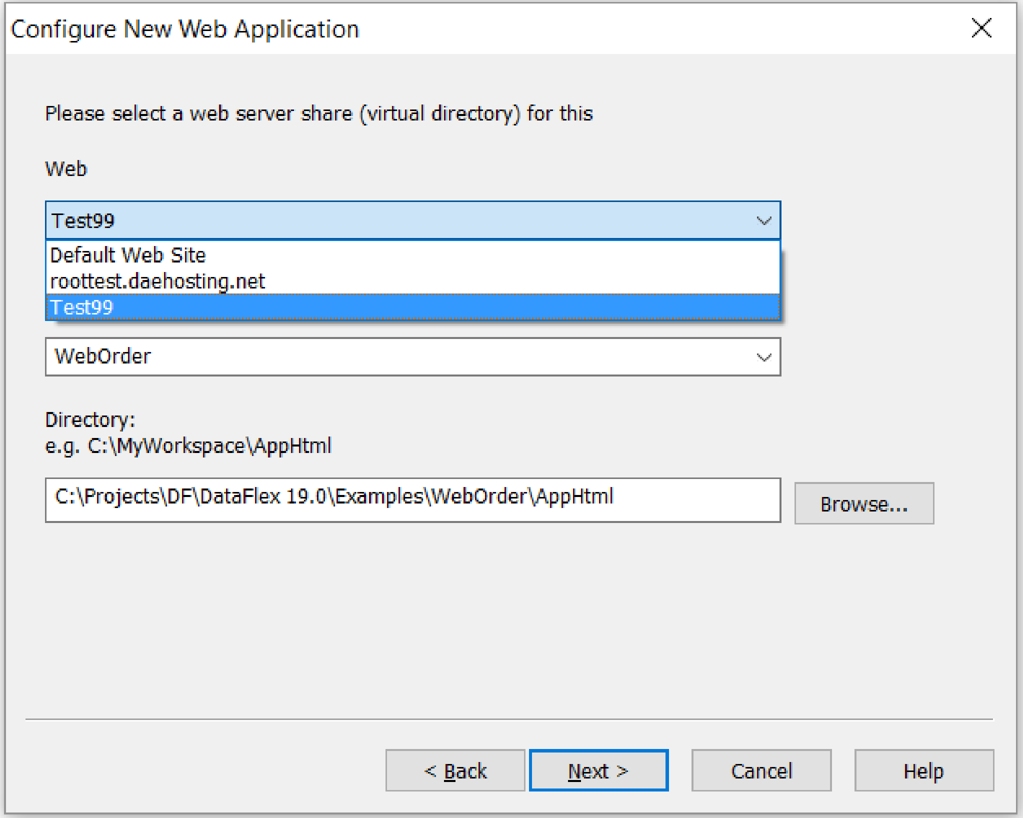
DataFlex Styler

- https://styler.dataaccess.eu/
- Login with Data Access ID

WebApp Server

WebApp Server

- 64-bit Support
  - Simplifies installation and configuration
  - Allows both 32-bit and 64-bit on the same server
  - Applies to the Server process only, DataFlex processes are still 32-bit
- Administrator allows applications to be configured to run on other than the default web site
- New Command Line Interface for Managing Web Applications
  - Allows more control over which website is used



Configure New Web Application

Please select a web server share (virtual directory) for this

Web

Test99

WebOrder

Directory:
e.g. C:\MyWorkspace\AppHtml

C:\Projects\DF\DataFlex 19.0\Examples\WebOrder\AppHtml

Command Line Interface

- While many developers use the Web Application Administrator to manage their web applications, it is often desirable to automate the process via the command line interface
- DFManageVDir.exe replaces WinCrtVDir.exe
- Used during installation and called from the Studio and Administrator (DFManageIIS.dll)

- Geeky Details
  - Create or remove virtual directories
    - DFManageVDir.exe create “{website}" "WebOrder_19" "C:\DataFlex 19.0 Examples\WebOrder\AppHTML" /overwrite
    - DFManageVDir.exe remove “{website}" "WebOrder_19“
  - Use /quiet mode to hide console

  - Register or remove ISAPI Handler
    - DFManageVDir.exe add-wso-restriction
    - DFManageVDir.exe remove-wso-restriction

Runtime JSON Parser

Why a JSON Parser?

- Designing the server side web properties we needed the ability to...
    - Take DataFlex data of any type (and complexity) and convert it to JSON
    - Take JSON data of any type (and complexity) and convert it to a DataFlex type
- We added this capability to the runtime
- As long as we did that, we made this public so you can do the same thing
- As long as we were doing that we added an interface for a full JSON parser

More about JSON

- Introduction to JSON in DataFlex – Harm
  - Last session today
- Getting the most from REST and JSON – John
  - Tomorrow

A Change in Strategy...

A change in strategy...

- Following a product release we would plan and implement the next revision, mainly focusing on new DataFlex features and enhancements.
  - Bug fixes (and doc improvements) were often left for later in the process
- The new release contained both new features and fixes to the previous revision.
- Some enhancements planned for a particular release are deferred and delivered in the next major product release.
- The result of the sequencing of bug fixes and deferred enhancements into a new revision is that developers didn’t get their benefits until the next major DataFlex revision was published.
- To deliver bug fixes and deferred enhancements to developers faster, we changed our approach.

Updated Release Builds

- Fix bugs in a new revision's code branch at the beginning of its development cycle
  - Special attention paid to bugs in the "new" stuff
- Developers' reports of needed refinements or changes received soon after a release are addressed at the beginning of a new release project
- Deferred enhancements are completed at the beginning of a new release project
- When a reasonable volume of the above has been completed in the new release code branch, some that will not affect prior-build compatibility or product stability are back-ported to the released product and delivered as a new build (just like a new Alpha or Beta)
- DataFlex 19.0.33.4 was the first DataFlex build to be delivered as a result of this new scheme.

What’s Next?

DataFlex Next

- In addition to the 19.0 update build, and the significant investments we've been making in the next generation of DataFlex we've also been busy crafting the next release of the current generation; DataFlex 2018 – 19.1.

- While we often prefer to disclose product details at the same time we are prepared to put an Alpha in your hands, we decided to unveil some details here at DISD.

- The full slate of features is not yet locked down but we want to point out some of the main features

Here’s some of what we’ve been up to...

- Language and code cleanup
- Compiler warnings
- Further connectivity enhancements
- Web Framework
  - History management and meaningful URLs
  - Expandable lists
  - Horizontal grid scrolling
  - Material design theme
- More Windows DPI enhancements
- Embedded manifest files
- Update to CodeJock 18.3
- Online documentation

Language and Code Cleanup

- Remove really old commands from FMAC
  - Don’t work in windows or web applications (character mode and pre-OOP)
  - Moved to [Really]OldFmacCommands.pkg
- Mark old classes obsolete and private
  - Get them out of the doc
  - Older web classes, MAPI, Crystal, etc.
- Remove packages that don’t belong in Web applications
- Restructure some class hierarchies
- Improved handling of duplicate definitions
- Modernize all public code and documentation
  - We’re also cleaning up our internal code

Compiler Warnings

- The Studio will provide compile time warnings for...
  - Obsolete commands
  - Obsolete global functions
  - Obsolete external_functions
  - Use of Indicate command instead of Move command
  - Use of “Local” in commands to define local variable
  - Use of “public” or "private" in property commands
  - Obsolete classes
  - Use of desktop methods without GLOBAL or DESKTOP
  - Obsolete packages (raised when a pkg is used)

- Warnings can be controlled at the project, file or code level

History Management and Meaningful URLs

- We are making the back button work!
- ASP Web applications used to rely on full page reloads to record history
- AJAX applications rely less on page reloads (refresh parts with data loaded using AJAX calls)
- Single page applications do not reload the page at all so there is no history to record (and no support for deep links)
  - Web Framework applications are single page applications
- Modern browsers solve this problem have APIs for manually maintaining history
  - history.pushState
  - history.replaceState
  - popState event

History Management and Meaningful URLs

- We use the WebApp Framework JavaScript engine to handle history back and forward operations and also change the displayed URL outside of a page load
- Provide a low-level API
- Standard URL generation for both drill-down and desktop style
- Standard history management
- Modes - URLs & History, History only, off
- Deep links allow URLs to navigate to specific points in the application (but require and respect login)
- Customize generated URLs – view names, record id generation, custom details

When?

- Until we complete the process of determining which changes will make the “final cut” it’s impossible to lay out meaningful targets
- However, since we commonly use Alpha releases to get more mature components into your hands in advance of the complete product profile we can say that we are targeting our initial Alpha release before the end of Spring 2018

Thank you for your time