



DataFlex to New Heights

# DataFlex – What's Hot and What's Coming Soon

Stephen W. Meeley

# What's Coming ~~Soon~~ Now...

- > Introducing DataFlex 2018 - 19.1 Alpha 1
  - > **Language and Code Cleanup**
  - > **Compiler Warnings**
  - > **Enhanced DPI Awareness**
  - > Web Framework
    - > History management and meaningful URLs
    - > Expandable lists
    - > Horizontal grid scrolling
    - > Material Design Theme
  - > Embedded manifest files
  - > Further connectivity enhancements
  - > Update to CodeJock 18.3
- > Still Under Construction
  - > Online documentation
  - > New installer

# Language and Code Cleanup

- > DataFlex 19.1 provides a way for you to clean up some of your old code
  - > Optional process and is mostly handled through a new warning system
  - > Your old applications will continue to run the same
  - > There are some cases where your existing application may generate compiler errors. These are easily addressed
- > If you receive compile errors, it will be for one of three reasons:
  - > You are using some very old commands that have been removed from the basic command set (i.e., these commands have finally been removed from FMAC). They have been moved to OldFMACCommands.pkg
  - > You are using some classes that are obsolete and have been removed from DFAllent.pkg. These classes still exist and may still be used. They can be added individually as needed or added by using the OldDFAllent.pkg package
  - > The compiler is better detecting and reporting errors. This is always a good thing. The most likely error is a redefinition of a symbol via the Define command. This has never worked and until now it has just silently done the wrong thing

# Compiler Warnings

- > Language and Code Cleanup is cumbersome, so the Studio and compiler support a new warning system to systematically find and work through the massive amount of code that makes up the product (and your applications)
- > The Studio provides compile time warnings for...
  - > Obsolete commands
  - > Obsolete global 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
  - > Inclusion and use of obsolete packages (raised when a pkg is used)
- > Warnings are just that – warnings. A project with warnings will run just fine
- > Showing warnings can be enabled and disabled at the project level
  - > At a more advanced level, you can also control warnings as the file or code level
- > The Studio shows the warnings in the output panel. This makes it easy to jump to the code in question
- > Migrated projects will default to suppress warnings, new projects will default to show warnings
  - > We debated this (right up until yesterday evening) and will likely continue to do so during Alpha

# How much cleanup can there be?

- > For instance, I migrated these projects from 19.0 with no support for old classes or commands and compiler warnings on...
  - > Order Entry – no errors, 16 warnings
  - > Database Explorer – 19 errors, 72 warnings
  - > Studio – 99 errors, 2,934 warnings
  - > Database Builder – 319 errors, 3,395 warnings
- > And then I did this...
  - > Added Use OldDFAllent.pkg
  - > Added Use OldFMACCommands.pkg
  - > Turned off warnings
  
  - > Order - clean compile
  - > Database Explorer – 1 error (duplicate struct we now define)
  - > Studio – 3 errors (redefinition of symbols)
  - > Database Builder – 2 errors (redefinition of symbols)



**Enhanced DPI Awareness**

# Enhanced DPI Awareness

- > A number of changes in the runtime, packages, Studio and Migration Wizard work together to provide your Windows applications a more consistent and controllable level of DPI awareness...
  - > More accurate scaling metrics
  - > Form heights are now consistent and defaults set at the workspace level
    - > You have the choice to scale comboforms (default is on)
    - > Existing (migrated) workspaces will stay at 13, spacing of 2 (unless you change them)
    - > New workspaces will default to 12, spacing of 2
  - > Proper Icon scaling in cImageList32
  - > Improved DPI awareness in Buttons and TreeViews

# About Scaling Metrics...

- > We have changed the way the conversion between horizontal dialog units and pixel units are calculated
  - > This method is identical to the old method, which is the official way Microsoft says it should be done, but it is performed at a higher precision
  - > The result is the impact of rounding is much smaller. What this does is greatly reduce the difference in layout at different DPI – specifically there will be a less clipping at 125%
  - > In case you are interested in the particulars...
    - > 100% rounds up (creating extra space)
    - > 125% rounds down (creating less space)
    - > 150% rounds slightly up
- > This change makes these differences less dramatic
- > This makes moving between different DPIs much more consistent
- > This also makes it easy to design at any DPI you want on your computer with the knowledge that it will look good at different targets



# How will this impact your applications?

- > If you designed at 100%, you now have a little less space, which might result in clipping at 100%
  - > If this is the case, you already have real problems at 125%
  - > Once you clean things up at 100% (or at any DPI), it works well everywhere
- > We added a new desktop property, `pbLegacyDialogRatio`, in `cApplication` to control this
  - > By default it is false, which uses the more accurate metrics. Setting this True makes things worked like they used to
- > Since this is not a compiler change, any application will use this new behavior
- > If you look carefully you will notice that the view sizes are actually a little different at each DPI (100% less wide, 125% wider, 150% about the same)
- > Note - If you try this at different DPIs you must log out and sign in after any DPI change. If you just change the setting, the results you get are probably wrong

# But what about the rest of Alpha 1?

- > Fear not, we'll be spending a lot of time on the other amazing aspects of DataFlex 2018
  - > We made the back button work!
    - > Harm, 15:30 today
  - > Getting your applications ready for the future
    - > John, 9:30 tomorrow
  - > Material Design
    - > Nigel, 10:00 tomorrow
  - > Building modern and secure Web applications
    - > Harm, 13:30 tomorrow



Let's Play!

# A notable (and significant) change...

- > We have changed a very old DD behavior concerning switched parents. Switched means changing from one parent to another. This has always been supported but it had a strange multi-user behavior when a switch was made by an external user. Consider this scenario...
  - > User 1 loads order, finds an order and changes the shipping
  - > User 2 loads order, finds an order, changes sales person and saves
  - > User 1 saves
- > The user1 save will switch back to the original Sales person it had. It should have accepted the change from user 2. Now it does, unless user 1 made some change in parent 1 themselves (e.g., switched to another record or edit data in there).
- > Two new DD messages are added:
  - > Function ParentSwitched Handle hoServerDDO Returns bSwitched
    - > Sent to the main DD and returns true if the passed hoServerDDO has been switched.
  - > Procedure SetParentSwitched Handle hoServerDDO Boolean bSwitched
    - > Sets the parent switched “state” for a particular hoServerDDO. This will probably be marked as private as it must be used with great care. We need this in WebApp to resynch DDO structures.
- > DD\_Debug.dg and WebDD\_Debug.wo have been updated to show these present switched states on the misc page.
- > You can test this on Order (windows and web) by switching parents both within a view and across multiple instances. In Order you can switch SalesP and Inventory Item. In Invt, you can switch Vendor.

# When?????

- > First, the oath...
- > Now, the link...

[https://www.educconference.com/DataFlex19\\_1Alpha](https://www.educconference.com/DataFlex19_1Alpha)



DataFlex to New Heights

**Thank you!**  
**Are there any questions?**

