

DataFlex 2014

Overview

- Implemented product branding
- Enhanced the DataFlex Studio
- Extended the Windows & Web framework
- Enhanced database connectivity
- Improved documentation
- Worked on bug fixes and suggestions
- Extended example applications
- Created new utilities

DataFlex Branding

- The newest release of our flagship development and deployment suite is DataFlex 2014 - 18.0.
- The “Visual DataFlex” identity you are accustomed to has been replaced with the new “DataFlex” branding nomenclature.
- How does this branding impact the installed product?

DataFlex Branding

- What we changed...
 - Default locations
 - Titles and nomenclature
 - Application internals
 - Registry branches
 - Service names
 - COM registration names
 - Documentation
- What we didn't change...
 - Physical component names
 - Class and method names

Studio

- Major changes:
 - Table Viewer
 - Native data types support
 - Enhanced Web Previewer
- Lots of “little gems” that are “really handy”

Table Viewer

- Select any table
- Columns can be resized, moved, added or removed (Field Chooser), sorted
 - Easily restore the default layout
- Edit (raw) or delete data
 - Mainly for ease of use when dealing with test data
- Does not replace Database Explorer

Native Data Types

- You now have the choice of displaying column types as *logical* types (i.e., DataFlex Types) or *native* types (i.e., SQL types)
 - Developers working with SQL are used to using native types
 - One method is not necessarily better than the other
- Supported throughout the Studio (Table Editor, Code Explorer, Code Editor, etc.)
- Native types are still mapped to DataFlex types and the best type is chosen automatically
- Lengths are checked for validity and sensibility

Web Previewer

- Select components directly in Previewer
- Synchronizes with
 - Code Explorer
 - Code Editor and Properties
- Toggle on and off with F7
- Refresh (context menu)
- Supports:
 - Card containers
 - Disabled and hidden tabs
 - Disabled menus and items

Gems

- Class Palette Filtering
- Find in Files options
- “Create New...”
 - In context
 - Captions in templates
- New Toolbars
 - Find Results
 - Message Trace
- Web Applications
 - Disable Web Application
 - Copy application url in Workspace Explorer

Gems

- SQL – close tables and logout when closing workspace
- Tabs
 - Simplified, copy file path and close group
- Table explorer filters (per workspace / per user)
- Expand tree in Code Explorer
- Debugger tooltips
 - Truncate at 600 characters
- Move table
- Font scaling improvements

Web Framework

- New [Controls](#)
 - File Uploader (form and button)
 - Column Button
 - Column Link
 - Column Image
 - Progress Bar
 - Suggestion Form (in grids too)
 - Group
 - DatePicker

Web Framework

- New standard struct used to transmit JSON data between client and server controls
 - tWebValueTree is the format for the transport mechanism for data between the client and server
 - This change is not backwards compatible and may require changes in your code
- New JavaScript mixin system (df.mixin) allows classes to be generated based on a predefined class but inheriting from another class
- Custom key handlers (AddKeyHandler and RemoveKeyHandler)
- Errors as client-actions (correct order)
- Field error API is public (showControlError, hideControlError and hideAllControlErrors)

Web Framework

- Information balloons above (or below) controls (ShowInfoBalloon and HideInfoBalloon)
- Custom CSS for rows in grids and lists (OnDefineRowCSSClass)
- Custom Tooltips for columns in grids and lists (OnDefineTooltip)
- API for manipulating data in manually loaded grids and lists (DataSetUpdateRow, DataSetAppendRow, DataSetInsertRowBefore and DataSetDeleteRow)
- New properties for client metrics (piScreenWidth, piScreenHeight, piWindowWidth and piWindowHeight)

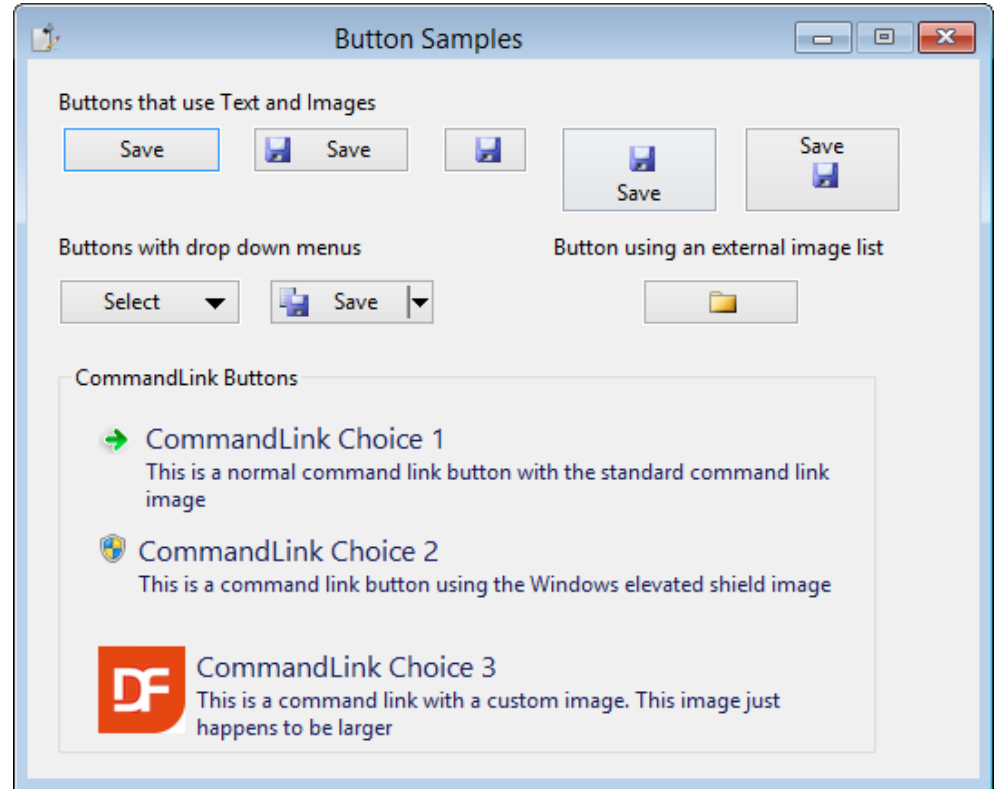
Web Framework

- Lots of fixes for browser differences
- Improved numeric entry filter, data entry filter and mask entry filter
- Datetime datatypes (includes masking and input filtering)
- Updated WebOrder example
- Increased Column default from 10 to 12 (better divisibility)
- Updated DFOs and Templates

- Much more (talk to Harm...)

Windows Framework

- New Controls
 - Split button
 - Command link button
 - Suggestion form (in grids too)
 - Button
 - pbShield
 - psImage
 - pilmageSize
 - pilmageMarginLeft
 - pilmageMarginRight
 - pilmageMarginTop
 - pilmageMarginBottom
 - pelmageAlign



Windows Framework

- Alignment improvements
 - Label, textbox and form
 - Combos are “special”

My Project

File View Report Navigate Window Help

Clear/Add Clear all Save Delete Remember

AlignmentSample 17.1

Default Settings 13

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ. ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

Make Things 12

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ. ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

Make Things 11

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ. ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

Make Things 14

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ. ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

AlignmentSample 17.1

My Project

File View Report Navigate Window Help

Clear/Add Clear all Save Delete Remember

AlignmentSample

Default Settings 13

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ: ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

Make Things 12

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ: ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

Make Things 11

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ: ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

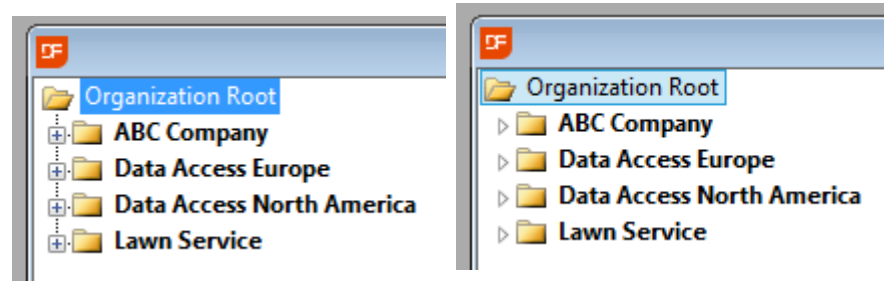
Make Things 14

ZZZXXXZZZ ZZZXXXZZZ TXTBXTXTBXZZZXXXZZZ ZZZXXXZ: ZZZXXXZZ ZZXXXZZZ ZXXXZZZ

AlignmentSample

Windows Framework

- CreateNamed - creates dynamic objects with names (which is helpful when debugging!)
- Popup_Modal - allow views and dbViews to be invoked modally
- Better icon handling with high DPI
- cTimer class – replacement for DFTimer
- ReadOnlyColor and pbNoEnterReadOnly in edit controls
- Sort combos in grid columns
- Explorer style for TreeView (pbExplorerStyle)



Data Dictionaries

- Support SQL Filters
- New constraint messages
 - DDOConstraints
 - Information about the constraints active for a DDO
 - DDOConstraintFindMeta
 - The DD find optimization meta information is now exposed
 - DDOConstraintNeedsRelate
- Error messages include table name
- DD_Debug improvements
 - Implementation of DDOConstraints and DDOConstraintFindMeta

Connectivity

- Includes latest connectivity kits (6.0.0.30 so far)
- Improved integration with the entire system (runtime, DDs, frameworks, Studio, etc.)
 - Expect too see more of this in upcoming releases
- Revision 6.0 (debuted in DataFlex 17.1) started the ball rolling with really significant additions and enhancements
 - Support for MSSQL 2008/2012 types
 - Support for DB2 10.1 types
 - Improved type mapping
 - Improved restructure (didn't mess up all that improved type mapping)
 - Auto Reconnect

Connectivity

- Improved support for native datatypes
 - Continued refinement at the driver level
 - Fully modeled in the Studio
- SQL Filter integration
- ESQL Performance

SQL Filter integration

- SQL Filters have been around for a long time
- Results of a forum discussion in May last year
- Pre and post finds
- Meta data about constraints
- Full DD support
- Fully supported in suggestion forms (Windows and Web)

SQL Helper Functions

- They help you assemble your SQL string
- Function SQLStrLike Integer iField String sSearch Returns String
- Function SQLStrAppend String sOrigFilter String sFilter Returns String
- Function SQLStrFileFieldName Integer iField Returns String
- Function SQLEscapedStr String sSQL Returns String

SQL Like

```
// Use helper function to build the SQL filter for you
Object SalesP_DD is a Salesp_DataDictionary
  Set pbUseDDSQLFilters to True
  Procedure OnConstrain
    String sFilter
    Get SQLStrLike Field SalesP.Name "ab" to sFilter
    Set psSQLFilter to sFilter
  End_Procedure
End_Object
```

ESQL Performance

- Optimization that focuses on getting data instead of getting information about the data
 - SQLColumnValue
 - SQLFetchRowValues
 - SQLFetchResultSetValues
- Results reported in the Forum
 - $256 \text{ columns} * 61,311 \text{ rows} = 15,695,616 \text{ values fetched}$
 - $17.1 = 508.31 \text{ seconds (120 rows per second)}$
 - $18.0 = 85.119 \text{ Seconds (720 rows per second)}$
 - $18.0 \text{ using SQLFetchRowValues} = 38.932 \text{ seconds (1,575 rows per second)}$

Client Web Services

- The client web service class and the Studio's WSDL parser have received a major overhaul. The two most significant changes are better support for nullable/nilable data types and better generation of struct names.
- Web services may expect that optional parameters (null values) will not be passed
 - When the data is passed (even when technically allowed by the service description) the service raises an error
 - You can now specify variables, both simple and complex, as null and decide not pass empty data which makes dealing with the service much easier
 - Previously you would have to perform low level customizations to alter the XML (complicated) or you would not use the service at all

Client Web Services

- Complex web-services often generate complex struct definitions
 - Could require special name "mangling"
 - Resulted in names with suffixes consisting of mixes of underscores, numbers and letters
 - Now most names have no special characters resulting in classes that are much easier to read and use

FleXML

- pvNodeValue
 - Used to get/set XML node values directly from/to a variant string (BSTR)
 - Data of any size can be used and there is no translation between Unicode and OEM
- Better handling of asynchronous document loading
 - This consists of fixes, new interfaces and better documentation
 - pbAsync – now defaults to False
 - pbReadyState – was not returning the proper value
 - LastParseError – new

CodeJock

- Updated from 15.3.1 to 16.3.1
- Mainly to get fix for better display of icons on high DPI displays
- Also keeps us current (getting too far behind makes updating more complex)
- There are new themes
 - Office2013.dll and VisualStudio2012.dll
 - Windows8.cjstyles
 - Note that these need to be copied from Lib to your workspace Programs folder when you use them
 - There is documentation on how to use resource themes (the dlls) starting with DataFlex 17.1

Character Encoding and Translation

- Background
 - Character data is encoded in various formats
 - Single byte
 - ANSI & OEM Codepages
 - Multi-byte Unicode
 - UTF-16 & UTF-8
 - Different systems require different formats
 - This is not all automatic and there is no single standard
 - You must know the encoding format before you work with it
 - You handle these different types by conversion

Character Encoding and Translation

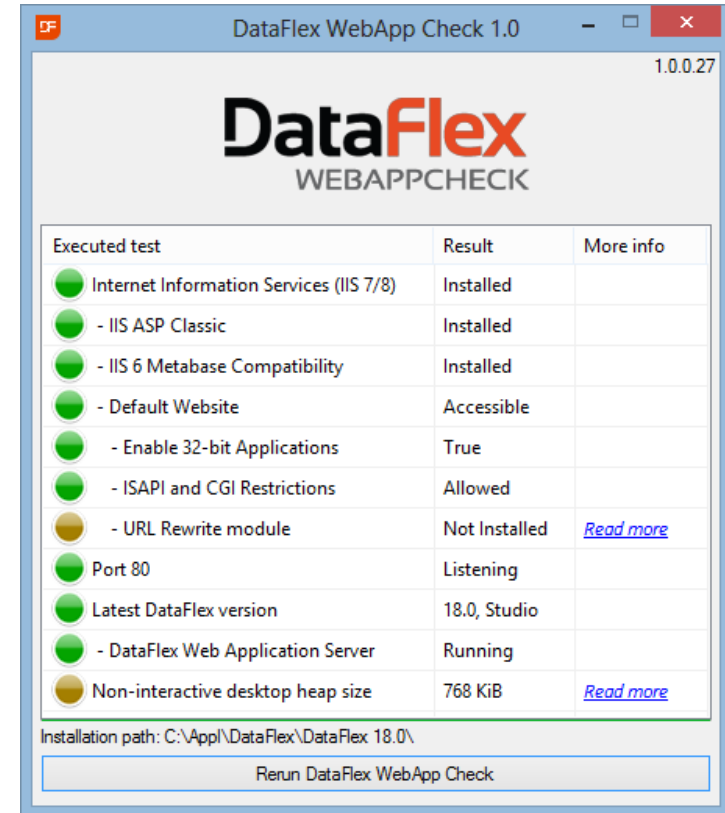
- Background
 - This is handled via character conversion
 - You need to do this on all systems
 - DataFlex does a lot of these conversion for you
 - DataFlex base is the single byte OEM string
 - It also supports Unicode inside of Variant (BSTR)
 - It also supports pure memory access
 - It automatically converts between ANSI and Unicode, when needed
 - DataFlex does a lot more of this than you may realize
 - New 18.0 classes have been provided that let's you explicitly convert between various encoding formats and data types

Character Encoding and Translation

- New Helper Classes
 - cCharTranslate
 - Convert between ANSI, OEM, UTF-8 and UTF-16
 - Convert between String, Variant Strings and Memory
 - cSeqFileHelper
 - Reads and writes various formats from (ANSI, OEM, UTF-8, UTF-16)
 - Reads and writes to DataFlex String and Variant data types
- Changes in XML interface can be used with this
- This can also bypass the DataFlex maximum string size limitation

WebApp Check

- Utility to determine if the environment is properly configured to host DataFlex Web Applications
- Also provides information about most current DataFlex installation
- Can be run in two modes: full and revision-specific auto-run (command line parameter with revision)
- You can embed this in your application installation (we did!) to make sure that the host is properly configured



SignDataFlexProject

- Windows asks permission for elevated rights and unsigned executables look a bit like a threat
- Helps to automate digitally signed components
- Need a tool (codesign or signtool) and a certificate (Verisign, Thawte etc)
- We use signtool because it's more automated
 - Uses the machines certificate store and not prompt for a password each time
- Use from the Studio to sign current project
- Updates the project dependency file
 - No recompile in the Studio on running after signing

DataFlex Utilities

- DataBase Explorer
 - Bug fixes and suggestions implemented – see lists in What's New
 - SQL Filter support
- DataBase Export
 - Bug Fixes
- Flexerrs
 - Major update includes use of CodeJock grids (sorting), a simplified interface and a form on the toolbar to type in a specific error number
- Workspace Migration
 - Reduced number of confirmations to streamline the process

Examples

- Web Order
 - Adds examples for all new controls
 - Pay special attention to the Demo menu
 - Column Types, File Upload, Suggestion Forms, etc.
- Order Entry
 - New Demo menu for Suggestion Forms and Buttons
 - We liked this technique so much in WebOrder we will use it more to consolidate examples
- Specialized Components
 - Order Generator
 - Restructured the Embedded SQL example to use the message interface instead of legacy command interface

A silver MacBook Air laptop is open on a wooden desk. To the left of the laptop is a glass of water. In the foreground, a smartphone lies on a piece of paper. The background is a blurred indoor setting with warm lighting.

Thank you for your time

Have a nice day!