

Version 2.0.0

The product is renamed to “Access Jump Start, a Rapid Development Framework for Microsoft Access” from RDF. Component name changes from RDF to AJS are as follows.

Old RDF Name	New AJS Name
documents	
HCI-RDF-ReadMeFirst.pdf	AJS-ReadMeFirst.pdf
HCI-Access-RDF-Product-Benefits.pdf	HCI-AJS-Product-Benefits.pdf
HCI-Access-RDF-License.pdf	HCI-AJS-License.pdf
HCI-RDF-Release-Notes.pdf	HCI-AJS-Release-Notes.pdf
HCI-Access-RDF-How-To.pdf	HCI-AJS-How-To.pdf
HCI-Access-RDF-Checklist - Existing App.pdf	HCI-AJS-Checklist-Existing-App.pdf
HCI-Access-RDF-Checklist - New App.pdf	HCI-AJS-Checklist-New-App.pdf
HCI-Access-RDF-Checklist - New RDF.pdf	HCI-AJS-Checklist-New-AJS.pdf
HCI-Access-RDF-Checklist Reference.pdf	HCI-AJS-Checklists-Reference.pdf
HCI-Access-RDF-Reference.pdf (Overview and Template specific sections)	HCI-AJS-Overview.pdf
HCI-Access-RDF-Reference.pdf (RDF Library specific sections)	HCI-AJS-Library-Reference.pdf
RDF Default Options settings.pdf	HCI-AJS-Default-Options-Settings.pdf
files	
RDF_Template_FE.accdb	AJS_AppTemplateFE.accdb
RDF_Template_BE.accdb	AJS_AppDbTemplateBE.accdb
Other named RDF...	AJS...
Tables, forms and modules	
DBversion_BE	tblAJSversion_BE (to be deprecated)
tblRDF...	tblAJS...
DBversion_FE	tblAJSversion_FE (to be deprecated)
frmRDF...	frmAJS...
RDF...	AJS...

Objects moved to FE Template from AJS_Library

- frmRDFusers -> frmAJSusers
- frmRDFuserLogin -> frmAJSuserLogin

Feature Updates

1. HALcontrol – Module with related class module creates a “Modern on-off control” as found in Windows 11. Example:



2. HALapp – Wrap Application.Echo method with a single instance class so that status is stored as properties.
3. HALdb.EmptyTable – Add passthrough query operation for ODBC connected table (for speed).
4. HALdb.NiceDLookup – Replacement for the DLookup function so that the new CurrentDB instance will be explicitly closed and released at termination.
5. HALdb.TableFieldExists - Determine if a field exists in a specified table.

Bug Fixes

1. RDF Dev only mode does not exit normally (keeps trying to restart).



Access JumpStart

Release Notes (for v2.0.0)

Documentation changes

The HAL_Library and AJS_Library documents are simplified by removing detailed interface information. The developer is encouraged to read the source code directly for that level of detail.

Version 1.5.1

1. New RDF.EnvironmentMode property returns "Dev" or "Live" based upon options settings RDF_EnvAutoDetect and RDFEnvManualSetting. When Auto Detection is enabled then RDFAppSessionUtil.RDFFuserEnvironmentMode will be called to implement the developer's detection method.
2. New module [RDFAppSetupUtil] is a container for Developer-customizable functions which are used during RDF environment setup. The following two functions are moved here from module [RDFAppSessionUtil]:
 - a. RDFFuserOptOverride()
 - b. RDFFuserEnvironmentMode() (renamed version of RDF_EnvironmentMode)
3. New HALdb.niceDLookup method functions like DLookup except that it cleans up after itself, terminating all Database resources used to get the desired information.
4. All usages of CurrentDB have been modified to clean up after themselves by Closing the created DB instance when processing is complete.
5. Updated RDF.DbLink and RDF.DbUnlink with bug fixes and to work with MySQL and Azure SQL back end databases.

Version 1.5.0

1. Back End database linkage enhancements
 - a. New developer dialog to setup and manage multiple Back End databases. See "Manage DB Connections" on the Main developer menu.
 - b. New options to specify name of table which lists the available databases (RDF_DatabaseList) and name of the table of tables to be linked in each of those databases (RDF_DbTablesList). When editing option RDF_DatabaseList you can double-click the value to bring up the Connections dialog in the context of the tables you are specifying.
 - c. New [RDF.DbLink] and [RDF.DbUnlink] methods to control Back End database linkage. RDF Back End tables are in the DbCode=PRIMARY database which may be an Access file or an ODBC-connected database. You can also place your application Back End tables in the PRIMARY database if you wish. You can specify an arbitrary number of additional Back End databases of varying types (MS Access or ODBC), each with an arbitrary number of linked tables.
2. Option changes
 - a. New option to write Session logs to a table in the PRIMARY Back End database. (RDF_LogStatus=1)
 - b. Option RDF_DevMode moved to [DevTools] category from [AppEnv].
 - c. New option RDF_vbw_VBEprint in [DevTools] category: Displays vbW error trap information in Visual Basic Editor (VBE) Immediate window when set =True. Only applicable when in Development Mode (RDF_DevMode=True).
3. Removed "chatty" startup messages in the VBE Immediate window.
Set RDFAppSessionUtil.RDF_ShowStartTime = True to display component startup timing.
Function bootRDFtimerElapsed() can be used by application code to display elapsed time and/or return the number of milliseconds since start of the application session.
4. See the HAL Reference manual for new library functions, including HALutil.findProcInModules, HALutil.getFormProcLineCt, HALutil.getModuleProcLineCt.

Version 1.4.0

5. **Development mode-only startup for debugging purposes:** (New development feature introduced in Version 1.3 base and updated in 1.3.1).
Start up the Front End via Left Shift and then enter command RDFstartDevOnly in the immediate window. This will initialize the RDF environment and bring up the Developer Tools Menu as the main menu. No application code will be initialized or executed.
Optional parameter DevOnlyOvr for RDFstartDevOnly() allows Override to be specified for a development-mode only startup. (It defaults to whatever override is returned by developer function RDFuserOptOverride()).
6. Bug Fixes:
 - a. Update application versioning code (only applies if using OASIS / TortoiseSVN / Subversion tools).
 - b. Library updates to repair HALform.Change_Height_Of_SubForm().
 - c. Fixed Developer Menu to display new Checklist documents.
 - d. Fixed HALutil.CalcWorkdays to ignore time values and added ability to specify which days of week should be counted as the weekend.
7. Added Form Resizing on [Options Update] and [Menu Edit] forms.
8. RDF Library now displays Release and Version number in Application Title when opened directly (for development usage).
9. Updated Checklist documents.
10. (11763) Library Feature: New [HALdb] module for DB helper methods. Move some methods from HALutil and add methods for accessing SQL Server Db via ADO.
 - a. "EmptyTable" method enhanced to return False if table not found or error.
 - b. "TableExists" method enhanced to optionally search external Database files.
 - c. New methods: "QueryExists", "Connection", "CreateLocalSQLTable", "ExecuteSQLCommand", "OpenSQLRecordset"
11. New HALtemp module to create temporary tables in Front End and/or a separate Database file just for temporary content. The external file can be deleted at session termination. Methods: "UpdateTempTable" and "KillTempDbFile"
12. New HALutil.getTextTwips() method. Returns height and width of text in TWIPS
13. New HALutil.wrapText method. Wraps a string with Quotes or any other designated string and escapes the wrap character(s) in the original text. (Great for building SQL commands, etc.)
14. RDF.MsgBox updated with [msgHelp] and [msgContext] parameters.

Version 1.3.1

1. Bug Fixes:
 - a. Splash form displayed when SplashTime=0 is specified
 - b. vbWatchdog compiler constant (vbWinst) not getting reset correctly
 - c. RDI_TestSetDbProperty fails after manual update of option table
 - d. Big Red X shutdown from user-defined main menu fails
 - e. Startup allocation of the RDF Library sometimes hangs
 - f. Option update form not allowing update to base table when a Session Override is active
 - g. Function RDFuserOptOverride() has no way to specify an error return
2. (10471) The form Resizer (HAL_FormResizeCls) is enhanced with added functionality and a simpler call setup requiring less modification to application forms.
3. (10415) Add more detailed field values validation to [Options Update] form.

4. Added optional parameter DevOnlyOvr to RDFstartDevOnly() so Override can be specified for a development-mode only startup. It defaults to whatever override is returned by developer function RDFuserOptOverride().
5. (10510) Added function RDF_IsOffice64Bit() in startup code (can be used before RDF is initialized). It returns True in a 64-bit Office environment and False in a 32-bit Office environment.
6. The How-To document includes some new items and some additional explanations.

Version 1.3.xxx

1. RDF Options can now specify Access Db Properties
 - a. Module RDFAppSetupUtil is no longer used and all the settings there are now contained in the standard RDF Options table. This includes Access Db property settings which are now specified as standard RDF options.
 - b. You can also define your own custom Access Db property settings using the Application options table feature.
 - c. New methods HALutil.setDbProperty and HALutil.getDbProperty are available so that you can create and manage your own custom properties directly via VBA code too if you like.
 - d. To merge your existing options into the new Options table, see item 2.c in the Version 1.1 Release Notes below.
2. New Options Override feature simplifies transition from Development to Production environments. You can specify an "Options Override table" that will selectively modify the standard RDF Option table at run time.
 - a. You can write code to select a specific Override table at execution time based upon any characteristics that you choose. (See new function RDFuserOptOverride() in module RDFAppSessionUtil.)
 - b. A default implementation of Development versus Production environment operation is included. This is controlled via the "Change Operating Environment" button on the Developer Tools menu.
3. There is a modified RDF startup sequence. The Autoexec macro now runs RDFenvBoot() followed by RDFstartup(). Feel free to read these if you like but you are encouraged to not modify any of the code. Doing so could make RDF unstable or inoperable. Instead, there are certain code modules that you are encouraged to modify if necessary, as noted below in the description of new module RDFAppSessionUtil.
 - a. New module RDFenvStart contains function RDFenvBoot() and related code which initializes a startup timer and sets up access to the Options settings for the session.
 - b. New module RDF_APPLauncher contains function RDF_APPLaunch() which implements version checking and copying the FE master and related files from the shared storage location down to the user's PC for execution.
 - c. You can use function bootRDFtimerElapsed() at any time to get the elapsed time since Front End startup of the current session.
 - d. Module RDFAppStart contains function RDFstartup() which does all RDF initialization. It contains public constant [RDF_ShowStartTime] which, if set True, will cause RDF startup timings to be displayed in the immediate window. You may wish to set this for application debugging.
4. Development mode-only startup for debugging purposes: Start up the Front End via Left Shift and then enter command RDFstartDevOnly in the immediate window. This will initialize the RDF environment, without Override, and bring up the Developer Tools Menu as the main menu. No application code will be initialized or executed.

5. New module RDFAppSessionUtil contains functions intended to be customized by you as desired.
 - a. Module RDFAppSessionUtil contains public constant *RDF_ShowStartTime* which, if set True, will cause RDF startup timings to be displayed in the immediate window. You may wish to set this for application debugging.
 - b. For emergency backdoor password recovery: Open the BE database and modify your user login entry in [tblRDFusers] by deleting the UserPWD field and clicking the PWInvalid box. Then save the table and close the file. Start up the front end and log in with your UserLogin and the password that is returned by function RDFAppSessionUtil.RDFelp. You will be logged in and prompted to enter a new password.
 - c. Function RDFAppSessionUtil.userOptOverride() checks option RDF_ProdOverride and uses this table name if it is present. In addition, it has code examples and suggestions that you may wish to implement instead.
 - d. Function RDFAppSessionUtil.RDFuserStartup() is for you to add your own startup code if desired. This is called at the end of RDFAppStart() processing.
 - e. Function RDFAppSessionUtil.RDFenvironmentMode() contains the default code to determine whether the present session is Live production or Development. You may wish to substitute your own processing logic.
 - f. Function RDFAppSessionUtil.RDFpasswordIsValid() contains the default new-password validation code that you may wish to modify.
6. HAL Library updates
 - a. New HAL_TickCounter Class Module allows you to establish multiple timer instances, each with 10-millisecond resolution.
 - b. HAL_ConnectODBC Module – minor fixes.
 - c. HALform.MoveSize – fixes, only set options which are specified, remove “frm” parameter.
 - d. HALform.setInputStatus – added callback function set by optional parameter.
 - e. New HALutil methods to manage Access Db properties:
 - setDbProperty: Sets or deletes a Db property.
 - getDbProperty: Returns a Db property value or indication if missing.
 - ObjectPropertyExists: This existing method can be used to just test for the presence of a Db property.
7. Option Table Changes
 - a. RDF_ValidateMode moved from AppData to AppEnv category
 - b. RDF_CustApplicationName in AppEnv category is replaced by RDF_AppTitle in DbProperty category
 - c. RDF_DevMode moved from DevTools to AppEnv category
 - d. New options added to replace the former property manipulation by function SetUserMode() which is now deprecated:
 - RDF_AllowBreakIntoCode
 - RDF_AllowBuiltinToolbars
 - RDF_AllowBypassKey
 - RDF_AllowFullMenus
 - RDF_AllowShortcutMenus
 - RDF_AllowSpecialKeys
 - RDF_AllowToolbarChanges
 - RDF_StartupShowDbWindow
 - RDF_StartupShowStatusBar
 - RDF_UseMDIMode

- e. New option RDF_ProdOverride is added to the DevTools category.

Version 1.2.77x

1. NOTE: *If you copied frmRDFAppMenu or frmRDFAppMenu_BB to make a new form of your own, then you MAY want to make these changes:*
 - a. Copy the [cmdLogOut] button to your form
 - b. Add this code to your form:

```
Private Sub cmdLogout_Click()  
    Call RDFmenuMgr.RDFmenu_Logout_Click(Me)  
End Sub
```
 - c. All other enhancements and fixes to the Application Menu are in Library code.
2. NOTE: There are startup code changes. Move your inserted application startup code from RDFstartup() to RDFAppSessionUtil.RDFUserStartup().
3. New RDF.LogOut and RDF.ShutDown functions can be called from application code to LogOut the current user or ShutDown the application, respectively.
4. HAL updates
 - a. HALfile.CreatePath – added
 - b. HALform.ArrowScroll – added
 - c. HALform – new functions added, primarily for ingesting data from external text files: CtlIsValidNumeric, CtlIsValidText, CtlValueIsBlank, CtlValueIsInTable, CtlIsValidDate
 - d. HALform.MoveSize – bug fix
 - e. HALform – new functions added to display a message on a form for a specified length of time: MsgDisplay, MsgClear
 - f. HALform – functions enhanced to search for Word values instead of just text substrings in Tag fields: setInputStatus, setVisibility
 - g. HALutil.MsgBox – bug fix
 - h. HALutil.ResetWorkspace – bug fix
 - i. HALutil.IsArrayAllocated – added
 - j. HALutil.IsLoaded – moved to HALform.IsFormLoaded
 - k. HALutil.IsWordInStr – added
5. RDF Library updates
 - a. RDF.CustApplicationName and RDF.CustomerName – Deprecated
 - b. RDF.LogOut – added
 - c. RDF.ResetWorkspace – bug fix
 - d. RDF.ResolveFilePath – enhancement and bug fix
 - e. RDF.Shutdown – added
 - f. Run-Time Version Checking can now copy support files down to each user's PC along with the Front End database file.
6. RDF Template updates
 - a. RDFAppSessionUtil.RDFUserStartup – added (new user code insertion point)
 - b. RDFAppStart.RDFReLogin – added
 - c. RDFAppStart.RDFstartup – updated (user code insertion point moved)
 - d. RDF_APPversioning – enhanced to auto generate required .bat and .tmp files
 - e. frmRDFAppMenu, frmRDFAppMenu_BB – Menus enhanced and bug fixed:
 - i. All menus now have "LogOut" and "Shutdown" buttons
 - ii. Role processing for each menu line now uses Word searching rather than substring searching. So, for example, selecting role [Admin] will not also select for role [SuperAdmin], etc.
 - f. [RDF_vbWinst] configuration option bug is fixed

7. RDF How-To Documentation is greatly expanded.
 - a. How to determine the version of RDF you are running has been added.
 - b. A new section describes How to locate the RDF Product Documentation.
 - c. The How-to set up Production Run-Time Version Checking has been updated to describe the new capability that copies multiple files, along with the Application Front End database file, down to the specified Local location for each user.
 - d. How to add your own Application Options has been updated.
 - e. Instructions to Configure RDF to use source code versioning have been updated.
 - f. How to set up a form for Read Only Modes has been updated.
 - g. How to set form control accessibility based upon User Role has been updated.
 - h. Appendix A: Sample OASIS setup parameters has been added.

Version 1.1.72x

1. Several new HAL library functions. See file "HCI-Access-Library-(HAL)-Reference.pdf" for full information.
2. Two new RDF options.
 - a. [RDF_MenuButtonTextColor] specifies the Text color on the application menu buttons. (The RDF Developer Tools Menu colors are not set by any of the RDF options).
 - b. [RDF_RibbonDisplay] specifies whether or not the Command Ribbon will display. It is normally set in module [RDFAppSetupUtil] and so it is recommended that you NOT set it via the Developer "Edit RDF Options" dialog.
 - c. To update your Options file:
 - i. Copy the new [tblRDFoptions] table into your project as [tblRDFoptions1]
 - ii. Run query [qryRDFoptions1_update] which will push your options over to the new table
 - iii. Rename your [tblRDFoptions] to [tblRDFoptions_bak]
 - iv. Rename [tblRDFoptions1] to [tblRDFoptions]
3. Changed the way that "Option Getter" routines are coded in module [APPopt] so that you can specify what value should be used if the option name is not found in the table.

Example:

```
Property Get MY_Option() As String
    MY_Option = Nz(RDI_getOpt("MY_Option"), "Value when entry not in table")
End Property
```

4. Modified both "Help About" screens to include [User Login] in addition to Windows User Name.
5. Modified the RDF "Help About" screen to include RDF Library release identifier.
6. Changed all "Return" buttons (with Exit Door icon) to "Go Back" with a Left Arrow icon.
7. Added "Restart" flag on left of Menu Footer. When in Development mode and an option is changed which requires MS Access restart for activation, this flag will be displayed. This is activated in the following situations and is displayed immediately:
 - a. Production / Development mode is changed.
 - b. Any option is changed via the "Edit RDF Options" dialog.
 - c. Any changes are made and committed via the "Edit Menus" dialog.
 - d. Any Application Colors are changed via the "Manage Color Settings" dialog.
8. File path resolution is enhanced to parse the Double Dot notation:
Say you want session logs to be in a directory named "MyLogs" placed as sibling to the Project directory. You would then enter the following [RDF_LogDir] option value:
"%ProjectPath%\..\MyLogs"
9. New Template module [RDFAppSessionUtil] contains functions RDF_EnvironmentMode() and RDF_PasswordIsValid(). You might want to customize these functions for your applications.

10. The code "Connect Secondary Back End DB files" in the "HCI-Access-RDF-How-To" document is corrected.
11. Changed function name RDF_SetStartupOpts() to RDF_SetDbProperty() for clarity.
 - a. Added [DbObject] as a parameter that can be "CurrentDb", "CodeDb" or any Access database object that you specify.
 - b. Added parameter [DeleteIfEmpty] so that you can delete or "un-specify" a user-defined property if desired.
 - c. In a future release this Sub will be moved to Function HALutil.SetDbProperty(). See also HALutil.ObjectPropertyExists(), a related method.
12. Fixed bugs in icon file processing. When you change the icon file specification it will display immediately on the next startup. Also, you can now delete the value for option [RDF_AppIcon] and RDF will revert to using the standard Microsoft Access icon.
13. Fixed bug in option [RDF_CustApplicationName]. It can now be set to a Null or empty string and RDF will revert to the standard MS Access window name.
14. Improved formatting of Menu Edit form. Now uses HALform.Change_Height_Of_Subform() to size the subform according to actual content.
15. Each Menu form may now include any number of command buttons. The standard RDF menus include 14 buttons in a single column. When you make a custom menu form by copying the standard template, you can place command buttons in any arrangement on the form as long as the naming convention [Option#] and [OptionLabel#] is maintained. Where "#" is an integer ranging from 1 to the total number of buttons. The range must be continuous (i.e. no missing numbers are allowed in the range).

Version 1.1.681

1. The main change in this release is that most of the RDF objects have been moved to an external Reference Library named RDF_Library.accdb. There are 27 objects in the Front End database and 6 objects in the Back End database that you need to deal with. All the rest of RDF is in hidden objects in the Front End or in the RDF_Library reference library.
2. New options (RDF_LibPath, RDF_LibName) define the path and full name of the RDF Library.
3. There are now 48 RDF Option Settings, divided into 8 logical groups so it is easier to find what you're looking for.
4. A new Session Authorization mode (option RDF_SessionAuth=RDF_LocalAuth) prompts the user for ID and Password at start up. The passwords are stored as a one-way encryption for security. Users may reset their own password and the administrator may reset any user's password if the user forgets it.
5. A new function (RDF.reAuthUser) can be called to force the user to reenter his password. This is useful for applications which require an "electronic signature" for specific transactions.
6. A new table (tblRDFuserPwdChgs) is used to log every user password change.
7. Removed option [RDF_Default_UserInactiveFalse_AskForFullname]. New / unknown user logins will not be prompted for their name. The menu name field will show "UID: UserLogin" until the Administrator updates the user's record to associate contact information.
8. Enhanced option (RDF_DevRoleTest=True) prompts Developers at startup to select the Role to be assumed for the current session. This allows much easier testing of menu and application behaviors for each of the Roles.
9. A new Developer Tools Menu option (Error Process Testing) allows for easy testing of 4 major types of program exceptions. This validates that new option RDF_DebugON is set and behaving as expected, and that, if installed, vbWatchdog behavior is as expected.
10. A new submenu (Reset Debugging Option) of the [Change Operating Environment] menu allows the Developer to change the setting of RDF_DebugON dynamically without requiring a

system restart. (If changed via the regular RDF Options dialog then a system restart is required to activate the change).

11. The Checklists document is updated.
12. A new "How To" document has been added.
13. A new "Bug List" document has been added.

Version 1.0.453

1. New Reference document added for HCI Access Library (HAL) modules. The HAL may be used independently of RDF if you like. RDF does include it and does require it.
(All RDF-specific functions have been moved to module RDF and some general utilities that were in RDF have been moved to the corresponding HAL... modules: HALfile, HALform, HALsys or HALutil. For example,
 - RDF.WinUserPublic is now HALsys.WinUserPublic and
 - RDFform.setFormTheme is now RDF.setFormTheme).
2. Updated RDF Reference document.
3. Fixes to form color processing in RDF.setFormTheme(): Text color is changed only for controls which have a transparent background.
 - If you have a control that is a special color, like Red, you can keep that control from being modified by adding a Tag value "NoColor" to it.
 - Also, when you call RDF.setFormTheme in your form Open Event code, you can skip text color processing entirely by adding parameter "FormText:=False" to the call. However, that form will not be as pretty as others...
4. The Navigation Option "Show Hidden Objects" is disabled by default so you will only see the RDF and HAL Modules that expose RDF and Library interfaces to your code, and a few key Tables and Forms.
5. New wildcard "%BackEndDBPath%" for FilePathName type options refers to the value of option RDFopt.RDF_BackEndDB_Path. If all the other FilePathName type values for shared files are made relative to this, then it may be simpler to change configurations between Production and Development environments.
6. New option named "RDF_AppIcon" lets you specify an optional icon file to be used in place of the standard Access icon.
7. If the vbWatchdog warning message irritates you, then comment out the line 'RDF.MsgBox "Warning: vbWatchdog is not installed" ...' in RDFAppStart.RDFstartup().
8. If you want to test Role processing of the Menu, then set option RDF_DevRoleTest = TRUE and you will be prompted to enter the Role you would like to test at Session startup time.

Version 1.0.415

1. The Overview document is deprecated in this release. Content moved to other documents.
2. The Developer Tools Menu is now totally separate from the Application menu so it doesn't tie up one of the Main Menu buttons.
 - a. RDF Product menu pages have the HCI logo and a distinctive color scheme.
 - b. Application menu pages have the end-user logo and a color palette that you select. There are 7 pre-defined choices.
3. New form to edit RDF Options, with contextual help.
4. New forms to edit RDF Configuration, Application Options, Menus and select/modify Menu color palettes
5. New forms to Import multiple RDF log files and display a consolidated view (Suitable for use by your customer Admin users if you wish).
6. Splash form displayed at startup and when "Help About" button pressed

7. New file pathname wildcards added for Option settings. The full list is now: %ProjectPath%, %HOMEDRIVE%, %HOMEPATH%, %LOCALAPPDATA%, %TEMP%, %Public%, %USERPROFILE%
8. Ten new options added:
 - a. RDF_AppFormLogo: Name of image file for form logos
 - b. RDF_AppOptMod: Name of module with Getter Properties for developer-defined options
 - c. RDF_AppRelease: Release identifier for application
 - d. RDF_AppVersion: Version number when automated version control is disabled
 - e. RDF_AppVersionDate: Version date when automated version control is disabled
 - f. RDF_AppVersionDisplay: Display mode for RDF.AppVersionString() method.
A number from 1 – 4 as follows:
 - 1."Release rr.rr",
 - 2."Release rr.rr Build vvvvv",
 - 3."Release rr.rr.vvvvv",
 - 4."Version: # yy.q.vvvvv (Compiled: m/d/yy)"
 - g. RDF_MenuLogON: Should the menu manager log all menu activity?
 - h. RDF_DevRoleTest: Shall a DEV role user be prompted to test alternate Roles?
 - i. RDF_SplashForm: Name of optional Splash Form to be opened at Startup
 - j. RDF_SplashTime: Minimum time the Splash Form must be displayed, in seconds
9. Eight new utilities added:
 - a. RDF.UserRoleKey: Returns the RoleKey of the defined user
 - b. RDF.WinUserHomedrive: Gets Windows HOMEDRIVE variable for current user
 - c. RDF.WinUserPublic: Gets Windows Public variable for current user
 - d. RDF.WinUserTemp: Gets Windows TEMP variable for current user
 - e. RDFform.setFormTheme: Set form colors and Logo image
 - f. RDFutil.CalcWorkdays: Calculate the number of workdays between two dates
 - g. RDFutil.IsValidFileName: Validate a FileName per Windows standards
 - h. RDFmenuMgr.RDF_MenuMsg: Display a message from a menu item