Magic xpa 2.5 Release Notes

New Features, Feature Enhancements and Behavior Changes

Call with Destination – Backward Compatibility Enhancements

In Online, the following scenarios are now possible:

- Calling a program to a destination frame when the Call operation is defined in a handler that is a parent task of the task that includes the frames form.
- Opening an Online task in a destination frame when the Online task is called from a Batch parent and the Batch parent was defined to be called into the destination frame.

These capabilities provide backward compatibility with the splitter capabilities. Note that the behavior of these scenarios has some limitations. Refer to the Destination property help topic for additional information.

Table Control – Scroll Bar Appearance – Change of Behavior

When setting the **Scroll bar property** to **Yes**, the vertical scroll bar will now appear only if there are more records than the ones that can be displayed on the form (as in V1.9).

Rich Client – Local Data Sources Support in Range, Locate and Sort Functions

The RangeAdd(), RangeReset(), LocateAdd(), LocateReset(), SortAdd() and SortReset() functions can now be used on local data sources.

Rich Client – Client-Server Compatibility

The internal communication protocol between the Client module to the Server was changed. Therefore, pre-Magic xpa 2.5 clients (such as 2.4) cannot be used with a Magic xpa 2.5 server, and Magic xpa 2.5 clients cannot be used with pre-Magic xpa 2.5 servers.

Rich Client – DataViewToDataSource() Function – Change of Behavior

The DataViewToDataSource() function was changed and it will now only copy data between columns that have the same attribute and storage types. This change was done to improve the performance of the function.

Retrieving the Magic Executables Version

You can now retrieve information about the Magic version (as seen in the Help->About screen) by running the Studio or Runtime engine with the **-ver** parameter. For example: 'mgxparuntime -ver'. The result will be saved in a file named **mgxpaver.txt** in the Magic installation folder.

Floating License Usage for Mobile Devices

Floating licenses can be used only for licenses that are not limited to mobile devices or desktop computers.

Color Definitions – Alpha Support

The color definition screen was enhanced to support the definition of the colors' Alpha value. The Alpha value of a color defines the opacity level and enables the creation of semi-transparent colors.

This definition affects only the Android and iOS clients.

Oracle 12c Compatibility

Magic xpa was tested and is compatible to work with Oracle 12c.

WebSphere MQ Server 7.5 Compatibility

Magic xpa was tested and is compatible to work with WebSphere MQ Server v7.5.

Android™ and iOS™

The Android and iOS clients were enhanced and now support the following features:

- **Mobile specific properties**

  The form and controls now have a new section containing all of the mobile-related properties. These properties replace and extend the current capabilities that were previously defined in the **Additional Information** property. The **Additional Information** property values are still supported to maintain backward compatibility.

- **Opacity level support**

  A semi-transparent color can be defined and used in the controls. The definition is done by setting the Alpha value in the color definition screen as defined above. Using semi-transparent colors will enhance the appearance of your application.

- **Suggestion text (Hint) support**

  You can now define a hint for Edit controls by using the new **Hint** property. A hint is the text shown on an Edit control that is removed automatically while typing.

  See also program RMO01 in the Rich Client Samples project.

- **Navigation drawer support**

  A native Navigation Drawer is now supported by assigning a menu entry in the **Navigation Drawer Menu** property of the form. A Navigation drawer is the menu that is opened by swiping from the left side of the screen.

  See also program RMO02 in the Rich Client Samples project.

- **Form Animation support**

  You can now control the form entrance and exit animation by using the new **Enter Animation** and **Exit Animation** properties.

  See also program RMO03 in the Rich Client Samples project.

- **Tab control support**

  A native tab bar is now supported by using a Magic xpa Tab control. Each one of Magic xpa's tabs is seen as a tab page on the mobile devices.

  See also program RMO04 in the Rich Client Samples project.

- **Two-state image support**

  A two-state image (pressed/un-pressed) is now available for Android and iOS.

  You can define this type of control by using either a Check Box control or a Radio control and setting their **Image List File Name** property to a 6-state image.

  See also program RMO06 in the Rich Client Samples project.

- **Controlling the keyboard type**

  You can now define the keyboard type (such as an email or a URL keyboard) that will be opened when editing a control by using the new **Keyboard Type** property.

  See also program RMO05 in the Rich Client Samples project.

- **Controlling the keyboard action button**

  You can now define the text that will appear on the carriage return button by using the new **Keyboard Return Key** property.

  This allows you to force the value (such as Done) or set a specific value (such as Search or Go).

  When the Done, Search or Go values are pressed, the Select Internal event is raised so you can execute logic when these keys are pressed.

  See also program RMO05 in the Rich Client Samples project.

- **Title bar color support**

  You can now change the color of the forms' title bar by using the new **Title bar color** property.

- **Colum visibility support**

  The Visible property of the Column control is now supported for mobile devices.

- **Executing operations while editing – Change of behavior**

  When editing data in an Edit control that is attached to a Virtual variable, if an operation is executed (for example, if the operation is defined in the Control Modify event handler), the busy indicator spinner will not be shown. This will allow you to continue typing characters while the operations are executed.

- **Native code execution – New function**

  A new function named **ClientNativeCodeExecute** was added to allow you to easily execute native code (Java for Android and Objective C for iOS), without having to use the UserDefinedFunction in the native code as a single entry point.

- **Native code execution – Raising User events**

  It is now possible to raise a User event directly from the native code (Java for Android and Objective C for iOS). This capability replaces the previous one in which only the built-in External Event could be raised.

- **Native code execution – Finding a Magic xpa control**

  It is now possible to find the reference of a Magic xpa control directly from the native code (Java for Android and Objective C for iOS). This allows you to add custom native controls as children of the Magic control.

  See also program RNC09 in the Rich Client Samples project, which adds a native label control inside Magic xpa's Group control.

- **Native code execution – Samples update**

  The native code samples (programs RNCxx in the Rich Client Samples project) were updated to use the new native code execution features described above.

Android Only

- **Combo box – Native appearance**

  The Combo Box control will now appear according to the Android operating system's theme (instead of as a push button with a rectangle).

- **Pop-up window – Support location, size, position**

  The startup position of the pop-up window can now be customized, so you can define the location, size and position of the window (as in iOS).

- **Default theme – Change of behavior**

  The default theme was changed and is now the light theme.

  You can revert this change and have the previous dark theme appearance by removing the word **.light** from the **themes.xml** file located under the **RIAModules\Android\Source\MgxpaRC\res\values** folder in the installation folder.

- **Push notification – Change of behavior**

  The project ID in the integrated push notification capabilities is now set using the new **ClientNativeCodeExecute** function.

  For additional information, refer to the **Sending and Receiving Push Notifications** help topic in the Magic xpa Reference Guide and to program RNC06 in the Rich Client Samples project.

- **Native code execution – Internal rename**

  The internal core library package name was changed from **com.pdac.myact** to **com.magicsoftware.core** and the main class in the core was changed from **GlobalClass** to **CoreApplication**.

  Due to this change, the Android application files, such as the **AndroidManifest.xml** and the java files were also changed.

  If you added native code to the Android application, you should also make the changes in the updated files or write your custom code in a separate file and use the new native code execution capabilities to execute your code.

iOS Only

- iOS 5.x is no longer supported. Magic xpa is compatible with iOS 6.0 and above.

OUTPERFORM THE FUTURE™

Fixed Issues

QCR #	Description
76580	On iOS devices, when the DataviewToDatasource() function was used on a local data source with a unique index of Alpha type, data was not copied to the local data source and the function returned True.
77024	The focus on a record was removed while navigating through the tabs of a Tab control using the keyboard.
77033	If focus moved from an Edit control with the Border set to Yes to an Edit control with the Border set to No or vice-versa, the Edit control values were truncated in a specific scenario.
77036	Dragging and dropping an option from a list box that was not first selected did not work.
77037	When there was a Locked Row event handler, an Error Behavior of Abort, and a Link Query operation that did not return any records, the handler was executed in an endless loop.
77060	After tying to insert or delete records to an MSSQL data source that had a TIMESTAMP column in an index, an error message was displayed.
77067	When setting the Tooltip Timeout environment setting to 60, the tooltip disappeared after 5 seconds.
77072	Fetching records took longer than in previous Magic versions.
77074	Using a user-defined function from a component in a .NET property expression, resulted in an Attribute Mismatch error.
77079	In the Hebrew version, a scrollbar appeared after minimizing and maximizing the window.
77080	When a Right was defined in a component, the EvalStr() function returned the default value instead of the Right number.
77082	A user-defined function name with the first three characters of 'Not', resulted in an 'Unidentified operator' Checker error.
77083	Performing Cut, Copy or Paste actions in a certain scenario caused a crash.
77085	In a Rich Client program, when a BLOB in a Link Join operation was updated with content shorter than the original content, Magic xpa Runtime crashed when leaving the task.
77087	The Locate expression did not work the same as the Range expression.
77090	Running the Checker on a program with a .NET control caused the "Ambiguous Signature Match" Checker error.
77091	The splash screen was displayed when a task with the Parallel Execution property set to Yes was called.
77092	When using the WinMinimize() function in an SDI task that was called from a non-interactive task, the window was not minimized.
77095	The Call Remote operation returned a Message ID with a null in it and a garbage character attached to the end.
77099	There were display issues with the tooltips and Remark operations when switching between tasks.

QCR #	Description
77108	The ISNULL() function was not evaluated if it was not defined as the first function in the expression.
77109	The Visited Color of a Hyperlink button was not displayed properly when changed during Runtime.
77110	If the StatusBarSetText() function was used in a Batch task and the Open Task Window task property was set to No, text was not displayed on the status bar.
77113	The Last Update value was changed in the Program repository after creating an .ecf file when the program contained .NET controls.
77117	Raising an event from an Offline program that was handled in the Main program and calling a Batch task on the server with processing longer than 30 seconds sometimes caused Magic xpa RIA to crash.
77124	A Window Hit event that was defined in a subtask with Propagate=No caused the child window to move to background when clicking on the parent form or title.
77138	Debugging a Studio with components when the Open Components Using environment setting was set to Project and a full path or a logical name to the path was used, caused the references to components to be lost.
77156	When the Force Record Suffix property was set to Yes, the Post Record Update User event did not commit the record to the database.
77162	When calculating midnight, the AddDateTime() function returned 24:00:00 instead of 00:00:00 .
77164	An LDAP user could not be authenticated if it did not belong to a group.
77167	The SQ10 logging sample program did not work properly.
77173	Initializing a variable in a Link Create operation with the CallProg() function took longer than in a previous version.
77187	In a Rich Client program with a specific combination of an empty data view, a subform, variables outside of the data view, Force Record Suffix and links caused the Magic xpa server to crash.
77201	There were issues with some email messages, including an empty message body, attachments without a name, or no attachments.
77224	The selected value in a combo box was not displayed if the Picture of the attached variable was different than the Linked field.
77246	Settings were not saved in the Print Data wizard.
77250	The Allow Column Resize arrow was shown but could not be used at Runtime when the mouse arrow hovered over the end of the column title even though the Allow Column Resize property was set to No.
77251	A Multiline Edit control was only fully printed when printing one record.
77254	After converting from V1.9, some of the blank column names with an attached Label control remained blank in V2.5 and the Label control went behind the Table control.
77259	When the StatusBarSetText() function was evaluated in a Batch program or in Background mode , an error was written to the log.
77263	Records were added to the data source if they were added and then deleted in the same transaction.

QCR #	Description
77265	Inputting the date 00.00.0000 in an Edit control with a picture of ##.##.#### was not possible.
77267	Locating a record (Ctrl+L) did not find the record on its second attempt when the task had a CndRange expression on the range, the LocateAdd function was used and the ViewRefresh event was raised.
77268	Clicking a Combo Box control opened the combo box even when the control was associated with a variable that was in a Block If operation with the condition set to No in an RM Compatible logic unit.
77269	Using the Conversion V9 utility for DSQL caused comments to be concatenated with the SQL statements, resulting in an incorrect DSQL execution.
77274	If an Edit control had a Prompt help and an invalid range was defined for it, the Valid range error prompt was not seen on the status bar for Online tasks.
77275	When there was a Verify Error operation in the Control Verification logic unit of an Online task and the Display property was set to Status, the error was not shown in the status bar when the control had a Help prompt with no text.
77277	The Offline Changed Objects list was not displayed when reconnecting to Version control and the changes made while offline did not take effect.
77282	When raising the Change Color or Change Font events from a menu or context menu, 15-20 seconds elapsed before the Color or Font dialog boxes opened or the dialog boxes did not open at all.
77289	The Create Records event handler was not executed when the Allow Empty Dataview property was set to No, a user-defined function in the Main program was used in the Init value of a Logical variable and a CallProg() function was used in the return value of the user-defined function.
77302	In the Hebrew version, if a subform was attached to a Group control, the scrollbars were displayed while executing the program.
77303	When adding a new record in an Online task in Modify mode, by pressing F4 or the down arrow key, a copy of the previous record was created instead of a new blank record.
77305	In the Hebrew version, the last letters of the Local Files Folder option in the Open From Server dialog box were cut off.
77308	The Print Preview window was opened on the main monitor even if the Runtime window was open on the second monitor.
77309	When passing an invalid expression in an Invoke .NET operation's Arguments field, a Checker error was not returned.
77310	When PDF or text files were created in a loop, a crash occurred after approximately 9500 iterations.
77314	Pressing F1 on a pulldown menu did not open the Help.
77318	The number of records that were scrolled when scrolling on a table using the mouse wheel was smaller compared with previous versions.
77321	After migrating from V9.4, an unlinked control placed on a Tab control was not linked to the Tab control.
77325	When the X1 and X2 values of a Line control that was linked to the column of a table were changed, residual lines remained in the original location.

QCR #	Description
77333	Global parameters set using the SetParam() function showed - 0.00 for Numeric values without a decimal and showed ********* for Numeric values with a decimal in the Debugger's Watch variables screen.
77334	The Vertical Alignment property was disabled when the Multiline property was set to Yes.
77336	When a range was defined in the WHERE clause on a DATETIME field, an error was displayed.
77337	Loading a large project with a DataSources.xml file and DataSoucesIndex.xml file of approximately 7 MB each, caused the Studio to hang.
77344	In Rich Client tasks, a color set in the Alternating BG Color property was seen in the first row of the table.
77348	When generating an Import program and attaching it to a folder, the program did not attach to the folder.
77349	The context-related program from the samples projects was not working properly.
77366	When a blank table entry was overwritten with a table that had two pairs of date/time fields and the project was closed and reopened, the Studio crashed when parking on the overwritten table entry.
77368	The connection string saved in the .edp file for a project using TFS version control was updated every time a user connected to the project.
77372	In an Oracle database, when the SQLUTF Database Information parameter was set to Y with a VARCHAR2 of 2000 characters, the resulting Magic Unicode value was corrupted.
77375	The tooltips for functions in the Expression Editor did not appear for Windows Server 2012 and Windows 8.
77377	The RangeAdd() function did not work when it was evaluated in the Task Prefix of a Batch program.
77378	The RM Compatible logic unit was executed even when a task had an empty data view.
77393	The Menu() function returned the wrong value when evaluated in a Main Program User event handler or in an Online task called from the User event handler.
77402	A check box could be modified when the Modifiable property was set to No, the task was in Query mode or a RM Compatible logic unit was defined.
77429	The Checker returned an Attribute mismatch error for a valid ExpCalc expression that used Hebrew.
77441	In Oracle, when the database character set was UTF8, the ErrDbmsMessage() function returned question marks (?) when the error contained a Unicode character.
77444	Searching the Help when the Help was called from a parallel program, caused Magic xpa to hang and crash.
77445	When an Online task was defined with sequential Block If operations with Verify Revert operations, the task went into an endless loop.

QCR #	Description
77457	Raising a Cancel event while the transaction was defined in the parent, resulted in the changes being written to the SQL server.
77465	The table scrollbar moved when a parallel program with a Window Type of Fit to MDI was called from a program with a Window Type of Fit to MDI.
77469	Scrolling was slow when using the mouse wheel in an Online table with the Alternating BG color.
77471	While using the Get Definition utility, a field was missing in a specific scenario.
77474	In an Offline program, an Initial mode with an expression was not evaluated when the expression was based on a parameter that was passed to the program.
77478	A .NET assembly failed to load when any German umlaut character was used in the path.
77479	When using DSQL, changing views and browsing records in a RIA task, the server erroneously closed the session.
77480	The DropMouseX() and ClickWX() functions returned different values.
77491	Text in the column title of a table overlapped when the text was evaluated by an expression at Runtime when the Use Windows XP Theme environment setting was set to No.
77492	When using the Esc key to exit a combo box whose options list was open and then selecting an option from the combo box, the value of the selected option was not updated in an attached data variable.
77493	Moving an MDI window in a called program that had a .NET window, caused an Unhandled Exception error to occur.
77497	In the Data Source repository (Ctrl+D), when creating a new row from row 0 of the Expression Editor, caused the expression ID to be lost.
77501	The Find Reference utility (Ctrl+F) gave two results for an expression used as a condition in an Event header line.
77504	The text in a menu was not translated after using the SetLang() function.
77506	When using the Mgreq.ini settings DefApp and DefProg and sending requests using a browser, arguments were not passed to the called program.
77507	When using the Open from Server option with TFS version control, if an object was created but not checked-in, the project did not open.
77510	The Output operation was executed even if the condition was set to No or the line was disabled.
77515	The Hebrew date in a Rich Client task did not work if the picture was defined as DDD MMMMM YYYYYYH or DDDH MMMMMH LYYYYYH.
77520	A scroll bar displayed in Table controls even when it was not necessary.
78229	When F10 was pressed in Runtime, the focus shifted to the pulldown menu even if a handler was defined for this key.

QCR #	Description
78236	Calling a subtask that wrote data into a BLOB variable via a Form Output operation and then calling a second subtask that read the data via a Form Input operation caused the server to go into an infinite loop while a dummy variable in the calling task caused the server to crash.
78252	On an Android device, the focus was not set properly on an Edit control in a scenario involving Subform and Push Button controls.
80089	On an Android device, the keyboard was closed when performing View Refresh while typing characters in an Edit control.
81498	The selected value was not displayed in a combo box if the empty data view was displayed for a given range using the CndRange() function.
81508	The keyboard shortcut (&) for a button did not work on multi-marked records.
81566	Deleting a model from an application caused a crash when the application contained a program with a Vector and the internal ID of the deleted model was smaller than one of the vector's cell models.
81582	When the comma was used as the decimal separator, the selected value from a combo box was not displayed with the decimal value.
81601	In the Hebrew version, a scrollbar appeared after restoring the window and maximizing it.
81605	Raising an event from a component that created a .NET object invoked the ".NET object not set to an instance of an object" error.
81619	In Online tasks, scrolling a combo box attached to a Table control using the mouse wheel, on any table record except the first one, scrolled the table records instead of the combo box values.
82957	Calling a Rich Client program that had a Virtual variable using Direct SQL with an Oracle database and a Memory result database, caused the program to crash when the Transaction mode was None and the subtask updated the parameter.
104437	The Profiler tool was not working.
104670	Records were loading slower than in previous versions.
109337	In an Oracle DSQL Online task, when using a specific table column in a WHERE condition and the destination database was the same as the source, the executed SQL query was truncated.
109469	There was very little space between the scrollbar slider and the scroll arrow when scrolling the records, making it difficult to click the mouse in the remaining space.
109475	When the first record in a task's range was deleted and the Screen Refresh event was activated twice, Magic xpa Runtime crashed.
109773	When the focus was on a button and the user clicked to move to another record in a table, the MgGui.dll error was created.
109777	Controls that were linked to different tabs of a Tab control were linked to one particular tab after migrating from uniPaaS to Magic xpa when one of the tabs had a Table control linked to it.
109831	Sending .NET arguments to a user-defined function caused a crash.

QCR #	Description
109844	When running an Online program in Debug mode that called a component program and that program called another program, the Studio crashed when zooming on the subprogram's Call operation arguments.
114944	A specific scenario with a Numeric field and a BLOB field caused Magic to crash.
115526	The position of a Group control attached to a Tab control was shifted down if an expression was used in the Items List property.
115557	When syncing data from the client to the server on Android devices, the DataViewToDataSource() function did not work when setting [MAGIC_LOGGING]LogClient to Y.
115623	The custom DPI mechanism for Android devices did not work when the device name included a hyphen (-).
115652	When syncing data from the client to the server on iOS devices, the DataViewToDataSource() function did not work when setting [MAGIC_LOGGING]LogClient to Y.
115672	When syncing data from the client to the server, the DataViewToDataSource() function did not work when setting [MAGIC_LOGGING]LogClient to Y.
115746	The KbPut function did not work in an RTF control.
115812	A Direct SQL statement with any DBMS log level except None, caused Magic xpa to crash.
115843	The image on an image button appeared faded.
115844	The appearance of an image on a button was not sharp.
116139	When clicking a hyperlink that pointed to a file on a Browser control, the Download option did not appear for Android devices.
116265	In a specific scenario, clicking on combo boxes in a Group control caused the Runtime to crash.

magic

OUTPERFORM THE FUTURE™

Magic xpa 2.4c – New Features, Feature Enhancements and Behavior Changes

Subforms – Behavior Change for Unsupported Task Mode

Starting with this version, if an Online task that is running in a Subform control cannot be executed, the subform will close (and not the entire program, as in previous versions).

Requester Log – Number of Lines Threshold

The log key in the **mgreq.ini** file now supports an additional optional parameter named **Number of Lines Threshold**. This parameter allows you to change the total number of lines that will be stored in the log file. This setting applies only to the server-side log and not to the RIA log.

Android™ and iOS™

The Push Notification sample code is now integrated in the client.

Magic 2.4b – New Features, Feature Enhancements and Behavior Changes

Allow Empty Dataview – Support Expression

The Allow Empty Dataview property now supports an expression, so you can condition whether a task will go into Create mode or remain in Query mode when no data exists.

Logging() Function – Single Context Logging

The Logging() function now supports an additional optional parameter to enable logging only on a specific context.

Rich Client – Additional Log Level

A new log level with the code of 'B' was added to the Broker, Requester, Client module and the server.

This level logs only HTTP request arrivals and departures. This level is meant to be used for troubleshooting of production (and also testing) systems, especially in the areas of communication and component handshaking

Rich Client – ClientDBDiscnt() Function

A new function was added in order to close the connection to the local database.

This is mostly required when copying the local database from the server using the ServerFileToClient() function.

Rich Client – ClientDBDel() Function

A new function was added in order to delete a data source from a local database.

This is mostly required when copying all the data of the data source from the server to the client.

Rich Client – DataviewToDataSource() Function

A new function was added in order to improve the performance of copying data between a server data source and a local data source and vice versa.

Note that this function is currently in beta status.

Refer to the Magic xpa Help for information on how to efficiently use the function.

Rich Client – Folder and Wildcard Support in the ServerFileToClient() Function

The ServerFileToClient() function now supports transferring file names with wildcards (such as 'c:\images\image*.jpg') and folders (such as 'c:\images').

Implementing the ServerFileToClient() function on a collection of files or on a specific folder improves the flexibility of the code and the performance at runtime.

Rich Client – Run in Offline Even When Network Is Available

A new option was added to provide a way to start an application on the client without connecting to the server even if the network connection is available. This option enhances the startup performance since no connection to the server is made.

This new mode is enabled by setting the value of **N** for the **ConnectOnStartup** property available in the RIA execution properties.

Compatibility with Pervasive 11

Magic xpa was tested and is confirmed to be compatible with the Pervasive 11 database.

Next Window Event – Shortcut Change

The Ctrl+F6 shortcut for the Next Window event must now be defined in the Runtime Keyboard Mapping file in order for the shortcut to work. This definition is done automatically by the installation. However, if you use your own Runtime Keyboard Mapping file, then to switch between windows using the Ctrl+F6 shortcut, you need to add an additional entry in the keyboard mapping file for the Next Window event with the shortcut of Ctrl+F6.

Android™ and iOS™

The Android and iOS clients were enhanced and now support the following features:

- Performance improvements – The performance of different aspects of the clients was improved.

- Package resource files – It is now possible to package resource files (such as images and local database) in the mobile installation packages (APK and IPA). In order to include files in the package, you need to place them in the **Assets** subfolder in your customized application. These files will be copied to the cache folder after installing and starting the application on your device. Refer to the **Developing Mobile Applications** concept paper for additional information.

- Improved native code samples – The native code samples (such as push notification and image resize) are now part of the Rich Internet Samples project. These programs have a public name of RNCxx.

A PDF file explaining the required native code changes is available for each of these samples in the **Android and** iOS subfolders under the **Rich Internet Samples** project folder.

iOS Only

- Xcode 5 and iOS 7 support – The iOS client now support iOS 7.

  When compiling the client, you should use Xcode version 5.0.

Android Only

- Device DPI override mechanism – The Magic client relies on the DPI information defined in the device in order to display the controls in the same size on all the devices.

  There are some devices, such as the HP Slate 21, that return a wrong DPI value and, therefore, the controls do not look good on them.

  You can now override the device DPI values and define the correct value. This can be done by changing the **custom_dpi.xml** file in the **RIAModules\Android\Source\Res\Values** folder.

- Native code samples
    - New sample for Barcode scanning using the ZBar SDK
    - New sample for Image compression and resize
    - Updated sample for Push Notification

Magic xpa 2.4 – New Features, Feature Enhancements and Behavior Changes

Rich Client – Offline and Client-Side Storage

Magic xpa now enables Rich Internet/Client applications (RIA) to work while not connected to a server, and to store database information locally, on the client. You can now build applications with both connected and offline capabilities for Windows desktop, iOS and Android devices. Offline applications allow users to continue and be productive in areas with intermittent, limited or unavailable internet connectivity. While working offline, data is stored locally on a local database, and periodically, when internet connectivity resumes, you can synchronize it back to the server.

Using Offline programs and local storage can also improve performance by locally caching commonly used data, such as the one used in selection programs, so the server does not need to be accessed in order to fetch the data.

To benefit fully from this new functionality, we recommend that you read the **Developing Offline Applications** concept paper available in the *Magic xpa Help*.

Note: For iOS devices, the offline and client-side storage capabilities are in beta status. All other functionality offered by the iOS client is production ready.

The desktop and Android clients are fully supported for use with the new offline and client-side storage capabilities.

Rich Client – Application Startup – Behavior Change

Since Magic xpa now supports offline capabilities, when executing a Rich Client application when the server is unavailable, the application will remain open and will not be automatically closed. If you want to close the application in such a scenario, you can add a handler in the Main Program on the Unavailable Server event and in this handler, raise the Exit System event.

Rich Client – Delete Task Mode in Non-Interactive Tasks

Non-interactive Rich Client tasks now support the Delete task mode. This mode is especially useful to delete records from a client-side storage data source.

Rich Client – Disable the Retry Dialog Box on Network Error

A new function named **ClientSessionSet** was added to provide control of the current session parameters.

Currently this function supports the **EnableCommunicationDialogs** key with the values of True and False. Setting the value to False will not show the retry dialog box upon network error.

This behavior is useful when the application is written to handle a network error, such as using Offline programs and client-side storage.

Rich Client – HTTP Timeout – Behavior Change

A new Magic.ini setting named **HTTPRequestTimeout** was added to define the time that the RIA client will retry sending the request to the server.

This setting exists under the **MAGIC_RIA** section in the Magic.ini file and has a default value of 5 seconds.

The existing **HTTP Timeout** environment setting only controls the execution of the **HTTPGet**, **HTTPPost** and **HTTPCall** functions.

Due to this change, the **ClientNetworkRecoveryInterval** setting is now deprecated.

Rich Client – First HTTP Request Timeout

A new **execution.properties** setting named **FirstHTTPRequestTimeout** was added to define the time that the RIA client will try connecting to the server at startup (first request).

The default value is 2 seconds to allow fast loading of the application. For non-offline enabled applications running on slow networks, it is advised to set this key with a higher value, such as 5 seconds in order to minimize the effect of communication failures.

Rich Client – Links on Virtual Variables – Behavior Change

In Rich Client tasks, when a Virtual variable without an init expression is changed, its value is seen on all of the records. However, links based on the new value were not refreshed in all of the records. This has now been changed, so, when changing a Virtual variable, if it appears in a link, and the link is based only on Virtual variables that do not have an init expression, then the link fields and the success indication variable will be updated in all of the records as well.

Rich Client – Startup Splash Screen

The Rich Client application will now show a splash screen immediately when it is launched.

For the Studio – You can change the splash screen by changing the **startup.png** file in the **RIAModules\Desktop\Resources** folder.

For Runtime – A new setting was added to the Rich Client Deployment Builder in which you can select the splash image.

Note that only PNG files are supported.

Rich Client – Keyboard Buffering

A new special setting named **SpecialKeyboardBuffering** buffers keystrokes that cannot be processed immediately and sends them to the control when available. This is relevant for Rich Client tasks and only when you are parked on a modifiable Edit control.

Rich Client – Compatibility with Previous Versions' Clients

Due to the changes made to support Offline programs, desktop and mobile clients from a previous version of Magic xpa will not work with the Magic xpa 2.4 server.

To use this version's server, you must upgrade your clients to the same version.

Online – Migration from Version 1.9 Enhancements

The migration process from version 1.9 will now do the following for Online tasks:

- Attach controls to the column that they are placed on if the controls are not attached to that column.
- Place the Label control text as the column title text and remove the Label control for Label controls that are placed below a table header section.

Font Orientation Support

Font orientation is now supported for Label controls and the column headers in Online and RIA programs running on a Windows desktop.

End User Functionality Component Enhancement

The End User Functionality component used for user range, locate and sort now supports multiple columns with the same name.

Coordinated Universal Time (UTC) Support

The following functions were added to return the current UTC date and time: UTCDate, UTCTime and UTCmTime.

Calling Magic xpi 4.x Programs

It is possible to call Magic xpi 4.x programs using a Call Remote operation to a Magic xpi server.

The Magic xpi server should be defined with type = **Magic xpi** and server address = **SpaceName=MAGIC_SPACE&LookupGroupName=<name>&LookupLocators=<host: [port]>**.

Rich Client – Running a Program/Project from the Studio Directly on Android

A new option named **Execution on Android** was added to the **Debug** menu. When this option is toggled, the execution of a Rich Client program or the project will be done on an Android device or emulator.

To use this feature, you need to follow the instructions detailed in the **Execution on Android** topic available in the *Magic xpa Help.*

Android™ and iOS™

The Android and iOS clients were enhanced and now support the following features:

- **Connection Settings dialog box in the Generic Client – behavior change**

  In previous versions, the generic client opened the Connection Settings dialog box whenever the value entered was not accessible.

  This was changed and now whenever the value entered is not accessible or the server is unavailable, the client will try to run the application without accessing the server (Offline).

  If you want to change the value, you can do the following:
    - On iOS – Change the value in the iOS device's built-in Settings screen.
    - On Android – Change the value in the Android device's built-in Accounts screen. Note that the account detail screen also shows syncing capabilities. These capabilities are not supported and will not affect the application.

- **URL Retrieval Timeout**

  A new **execution.properties** setting named **URLRetrievalTimeout** was added to define the time that the RIA client will try retrieving the file defined in the URL key in the **execution.properties** file.

  The default value is 2 seconds to allow fast loading of the application. For applications running on slow networks, it is advised to set this key with a higher value, such as 5 seconds in order to minimize the effect of communication failures.

- **Tab Order**

  The **Tab Order** property is now supported.

- **Push Notification**

  A sample program and a step-by-step guide for how to implement push notification in your application was added to the *Magic xpa Help.*

  To implement push notification in your application follow the instructions detailed in the **Android Push Notification Sample** and **iOS Push Notification Sample** help topics*.*

Android Only

- **Minimum Android platform version for compilation**

  The minimum Android platform version for compilation (defined in the **Target** property of the **settings.properties** file) was changed from 11 to 15.

- **Theme support**

  The Android client will now appear according to the Android OS theme. This means that applications running on Android 3 will appear with the Holo theme and applications running on Android 4 will use the OS default theme.

  You can reverse this change and have the old appearance by deleting the **values-v11** and **values-v14** folders from the **RIAModules\Android\Source\MgxpaRC\res** folder in the installation folder.

You can change the themes by modifying the **themes.xml** file in the **values**, **values-v11** and **values-v14** folders from the **RIAModules\Android\Source\MgxpaRC\res** folder in the installation folder.

- **Scale to Fill image style**

  Images set with the **Scale to Fill** image style will now be shown centered to the form or control.

- **Default alignment in Edit controls**

  You can now define that the content of an Edit control will be aligned using Android default alignment rules. For example, you can set that English text will be aligned to the left and Hebrew text to the right.

  This can be done by setting the **DefaultAlignment** value to **Y** in the Edit control's **Additional Information** property.

- **PDF viewer**

  You can launch a third-party PDF viewer application installed on your device by using a client-side **Invoke OS** operation with the OS command **pdf:filename** (where **filename** is the PDF file in a public folder on your device, such as the /SDCARD folder). You can use the **ServerFileToClient** and **ClientFileCopy** functions to copy the file to that folder.

- **Modifying Edit controls in a Table control**

  The Android operating system does not behave as expected when it needs to open a keyboard for an Edit control that is placed in a Table control. To solve this problem, by default, when clicking on a modifiable Edit control (which is not a Date or Time control) in a table, an Edit dialog box will be shown and users will enter their data in this dialog box.

  If you want to maintain the previous behavior and enter your data in the Edit control inside the table, you can define the value **editdialog=0** in the **Additional Information** property of the Edit control.

- **Scrolling mechanism**

  The scrolling mechanism was improved and now supports scrolling on both parent and nested objects. The values of **ScrollHorizontal** and **ScrollVertical** in the Additional Information property are therefore no longer required and were removed.

- **Manifest and Source files**

  The **AndroidManifest.xml** file and the java source files were changed. If you changed these files in your application, you will need to make these changes in the new files.

BlackBerry Support

The BlackBerry client is no longer supported.

The Magic Compare

Magic Software is happy to introduce Ocean Solutions's complementary utility – the Magic Compare. The Magic Compare provides visual comparison of two projects or within a single project and locates all changes between repositories, objects, properties, and so on.

This version of the Magic Compare is a limited evaluation version displaying a partial compare results list.

To obtain a fully operational license of the Magic Compare, please contact Ocean Solutions at info@magic-optimizer.com.

Magic xpa Step-by-Step Video Clips

Magic now offers a collection of video clips that demonstrate how to use various Magic xpa features. New video clips are being added to our YouTube channel on an ongoing basis. To see the video clips, click here.

Magic xpa 2.3a – New Features, Feature Enhancements and Behavior Changes

Android™ and iOS™ – Spinner Display

The Android and iOS clients now display a spinner when the application is busy.

Android™ – Cache Folder

The Android client supports the 'temp' keyword in the ClientOSEnvGet() function to retrieve the cache folder (as in iOS).

Controls' SWFName Property

The value of the **SWFName** property of the controls can now be set with the value of Magic xpa's Control Name property if the **Control Name** property has a value and when the **SpecialSwfControlNameProperty** special setting is set to **Y**.

Magic xpa 2.3 – New Features, Feature Enhancements and Behavior Changes

Table Control – Multi-Mark Support in Online Tasks

Multi-marking is now supported for Table controls in Online tasks.

A row is marked by clicking on it. The **Marking Column** property is, therefore, obsolete and no longer supported.

Window Pulldown Menu Support for Online Applications

The Window pulldown menu is now supported and behaves similar to the V1.9x menu.

This menu will show all the windows defined with **Show in Window Menu = Yes**. The **More Windows** entry is, therefore, obsolete and no longer supported.

Label Control – Expression Support in Online Tasks

Expressions are now supported for the Label control in both Online and Rich Client tasks.

.NET Data View Control Binding in Online Tasks

It is now easier than ever to use .NET grid controls in Magic xpa.

A new property named **Dataview Control** was added to both the .NET control and the .NET control model so that you can define a .NET control as the dataview control. This definition means that the entire task data view will be assigned to the **DataSource** property of the .NET control.

You can now use any .NET control grid instead of the Magic xpa built-in Table control. Any modification made to the Magic xpa variables will be automatically reflected in the .NET control grid and vice versa.

Note that at this point this functionality is supported in Online tasks only and when the **Preload View** = **Yes**.

For more information on how to use .NET dataview control binding, see the .NET Tutorial sample installed with Magic xpa.

.NET Choice Controls' Data

It is now easier than ever to use .NET choice controls in Magic xpa.

A few new properties were added to the .NET control model so that you can define the .NET control properties for the data source, display member, and value member values.

After these properties are defined, the .NET control placed on the Form Editor will have the same properties as the built-in choice controls, so you can use the .NET control in the same way as you use any Magic xpa built-in choice control.

For more information on how to use .NET data binding, see the .NET Tutorial sample installed with Magic xpa.

.NET Data Binding – Behavior Change

When using a .NET control defined with data binding and updating the .NET control, Magic xpa automatically updates the Magic xpa variable with the .NET control value.

If after the update, the values in the Magic xpa variable and the .NET control are different, the control will be updated with the Magic xpa variable value (instead of keeping the typed value as in previous versions).

.NET in Batch Tasks Called from Rich Client Tasks

It is now possible to run .NET code in a Batch task called from a Rich Client task.

Note that:

- A client side .NET object (defined in a Rich Client task) will not be available on the server side (Batch task) and vice versa.
- It is not possible to send .NET variables as a parameter to a Batch task.

Generic Server Error Messages in Rich Client Applications – Behavior Change

At runtime, when there is an error serving a request (such as Server not found), a generic message containing only the error code will be shown (instead of showing a full message exposing the server address).

You can still have the previous functionality by specifying the 'N' value in the **DisplayGenericError** property in the execution properties.

List Box Control – Multi-Mark Support in Rich Client Tasks

Multi-marking is now supported for List Box controls in both Online and Rich Client tasks.

Edit Control – Vertical Alignment Support in Rich Client Tasks

Vertical Alignment is now supported for Edit controls in both Online and Rich Client tasks.

Choice Controls – Data Source Properties

The data source properties (such as **Item list** and **Display list**) of choice controls were separated into a new group named **Data Source**.

Form – Startup Position Property

The **Windows default bounds** and **Windows default location** values were renamed to **OS default bounds** and **OS default location** respectively.

Tab Control – Hebrew Version

The Hebrew version is now aligned with the English version and supports the following properties for the Tab control:

- Tabs width
- Hot Track
- Multiline
- Image list fie name
- Image list indexes

These properties affect runtime only. They do not affect the Form Editor of the Hebrew version.

Functions – Data View Variables' Indexes

A new function named DataViewVarsIndex() was added to retrieve the data view variables' indexes.

Functions – Names and Types of Global Parameters and Shared Values

The following functions were added to retrieve the names and the type of the global parameters and shared values:

- GetParamNames()
- GetParamAttr()
- SharedValGetNames()
- SharedValGetAttr()

Background Mode – Showing the Verify Message Box

A new special setting named **SpecialAppServerAllowVerifyInBox** was added to allow opening the message box of the Verify operation when the engine is running as an application server in Online deployment mode.

Application Installation Utility – Govern Web Service Installations

A new option was added to the application installation utility to define whether or not to install Systinet.

Enable Usage of 3-GB Address Space

The Magic xpa engine can now access a 3-GB address space.

For this to work you must modify the **Boot.ini** file to enable application memory tuning. To do this, add the **/3GB** parameter to the **ARC** path in the **Boot.ini** file. Refer to:

http://msdn.microsoft.com/en-us/library/windows/hardware/ff556232(v=vs.85).aspx.

Using Magic xpa from .NET Applications

It is now possible for .NET applications, such as ASP.NET or a .NET web service, to interact with Magic xpa.

For more information, please see the **Managed Requesters** Help page.

Broker Execution

When launching the broker, it will now be launched as one process (without a watchdog process as implemented in previous versions). If you need to load the watchdog process, you can start the broker with the **-StartAsWatchdog** parameter.

Note: This is only applicable if the broker is started as an executable (not as a service).

Rich Client Deployment Builders – Mobile Device Support

The Rich Client Deployment Builder now creates Android, iOS and BlackBerry packages.

For iOS, since the compilation is done on a Mac machine, the builder will generate the required files. Therefore, you should simply copy them to the Mac and run the build script. You may also need to copy the **iTunesArtwork** and **iTunesArtwork@2x** files to the Mac.

Note that the deployment files for each operating system will be created in a subfolder with the same name as the operating system. If you want to maintain the previous folder structure for Windows desktop deployment, use the builder metadata file from the previous version.

Rich Client Form Editor – Combo Box Height

The Combo Box control's Height property is now modifiable on Rich Client forms so it can be used to define the controls' height on Android and iOS clients.

On Windows platforms, the value of this property has no effect since the Windows combo box control ignores the Height property and calculates the height according to the font used.

Rich Client Form Editor – Z-order Limitation

The limitation of setting a manual z-order for some controls to be above other controls (such as to define an Edit control with a z-order value higher than a Button control z-order) was removed, so that it can be used to define the controls' z-order on Android and iOS clients.

On Windows platforms, combinations that are not allowed are still not supported, so they will be ignored.

Android™ and iOS™

The Android and iOS clients were enhanced and now support the following features:

- Edit control – behavior change
  - Alpha and Unicode – picture mask

    The picture mask of Alpha and Unicode Edit controls is now calculated when leaving the control (instead of after typing a character). This change provides support for the built-in auto-complete feature of the mobile devices.

  - Date and Time – pickers

    When the focus moves to an editable Date or Time Edit control, the native OS date or time picker will be opened.

    If you want to update the Edit control manually without a picker, you can use the **'picker=0'** value in the **Additional Information** property of the control.

- Resources per platform and device characteristics

  The Android and iOS RIA mobile client now support alternative resources to support specific device configurations.

  You can define multiple copies for the same resource and place them in specific folders on the server. The RIA client will automatically retrieve the matching resource according to the device platform, resolution and DPI.

  This can be used, for example, to automatically support:
  - Images for different resolutions.
  - Font files and color files for different platforms.

  For example, if the runtime font file is **Support\fnt_rnt.eng**, then you can put the Android font file at **Support\Android\fnt_rnt.eng** and the iOS font file at **Support\iOS\fnt_rnt.eng**.

  At runtime, the appropriate font file will be used automatically, if it exists. There is no need to define this file implicitly.

  This is also the case for images. For example, you can place an HDPI image at **c:\images\Android\hdpi** and an LDPI image **at c:\images\Android\ldpi**. At runtime, the appropriate image file will be retrieved automatically (if such a file exists).

  For more information about how to use alternative images, refer to the **Alternative images (Android and iOS)** section in the **RIA for Mobile Devices** concept paper.

- The controls' border can be fully customized for controls with a non-system color by using the following values in the **Additional Information** property:
  - 'BorderWidth=xxx' – the border width
  - 'BorderColor=xxx' – the border color from the color table
  - 'BorderFocusWidth=xxx' – the border width when the Edit control is in focus
  - 'BorderFocusColor=xxx' – the border color when the Edit control is in focus
- The corner radius of the Text, Edit, Button, Group and Image controls can be fully customized for controls with a non-system color by using the following values in the **Additional Information** property:.
  - 'CornerRadius=xxx' – defines the controls' corner radius

  Note: For iOS devices, the Subform and Table controls also support this functionality.
- The internal **Press** event is raised when performing a long-press gesture on a control or form on the mobile device.
- The application area size in inches can be retrieved using the **device_physical-width** and **device_physical-height** values in the **ClientOSEnvGet()** function. These values can be used, for example, to determine whether the device is a phone or a tablet.
- The **Row Highlight Style** property is now supported.
- Gradient Support – The form and the Group and Push Button controls support the following gradient styles: Horizontal, Vertical, Diagonal Left and Diagonal Right.

  Note: On Android devices, the Label control also supports the above gradient styles.
- Launching the application from another application

  You can launch your application from another application or from an email using a URI convention such as **myapp://?parameters** on iOS and **http://com.mycompany.myapp?parameters** on Android.

  The query parameters sent to the application can be retrieved using the ClientOSEnvGet function with the value of **'device_udf|getargs'**.

  For more information about how to use this feature, refer to the **Launching the application from another application** section in the **RIA for Mobile Devices** concept paper.

iOS Only

- Fixed orientation form, so the form will not be rotated upon screen rotate.

  This is done by defining the value of the **Additional Information** form property to **orientation=portrait** or **orientation=landscape** (as in Android).
- OS commands are now supported when defined as a menu entry (as in Android).
- Popup screens are supported by defining the **popup=Y** value in the **Additional Information** form property (as in Android).

On iOS devices, unlike Android, the window size and location can be defined using the navigation properties and the **Startup Position** property.

- Printing a PDF using the AirPrint protocol.

  Printing a PDF can be done via the Invoke OS command with the command of **Print:filename**, where **filename** is either:

  - A path to a local file (as received from the ServerFileToClient() function)
  - A URL to a file on a web server (such as 'http://1.1.1.1/myfile.pdf')

Android and BlackBerry Only

- Popup screen support – behavior change

  Popup screens are now defined using the **popup=Y** value in the **Additional Information** form property.

  The previous way (using a Floating window type) is no longer supported.

  If you used the Floating window type in previous versions, you need to manually update your application,

Prerequisites Update

The minimum .NET Framework version required to run Magic xpa is V2.0 SP2 (instead of SP1). When using this version, you also need to install the Microsoft Visual C++ 2008 Redistributable.

OUTPERFORM THE FUTURE™

Magic xpa 2.2a – New Features, Feature Enhancements and Behavior Changes

As Parent Initial Mode in Rich Client Tasks

The behavior of the **As Parent** initial mode in Rich Client tasks was changed and it will now be recomputed when there is a change in the parent task mode (instead of being set only once when the subform task is opened).

Database Connection Pooling

Magic xpa uses a database pooling mechanism that allows for sharing of connections between different contexts.

This behavior improves the efficiency and the performance of the server.

However, there are scenarios in which you need a separate connection per RIA client, so when the context is closed, so does the connection.

This can now be done by setting the new **SpecialDatabaseConnectionPooling** flag to **N**.

Android™ and iOS™

The Android and iOS clients were enhanced and now support the following features:

- Subform control
- Client side images
- Spinner display – This is done using the SetCrsr() function.
- Image picker from the gallery – This is done by sending the value of 'images' in the ClientFileOpenDlg() function's second argument.
- Call to a native OS code – This is done by evaluating the ClientOSEnvGet() function with the value of **'device_udf|my_string'**.

  Performing this action will send the value **'my_string'** to the function named **userDefinedFunction** in the mobile application code.

- Call from the native OS code to the application – This is done by running the **invokeExternalEvent("my_string")** command in the native OS code.

  Performing this action will raise the internal event named 'External Event' in the Magic xpa application and send the value **'my_string'** to it.

iOS Only

- Simpler customization methodology – The iOS application customization settings (such as defining the package name and the provision file) were changed and are now defined in a single file named **settings.properties** as in Android.

OUTPERFORM THE FUTURE™

Android Only

- Fixed orientation form, so the form will not be rotated upon screen rotate – This is done by defining the value of the **Additional Information** form property to **orientation=portrait** or **orientation=landscape**.

- Define scrolling on the form or its controls – The Android interface allows only one scroll object inside a hierarchy. This means that, for example, if the form has a scrollable object, such as a Table, Browser or Subform control, then the scroll can be done in either the form or the subobject, but not in both.

  You can set the scrolling for the form or its controls by defining the value of the **Additional Information** property of the form or its controls to **scrollhorizontal**=X or **scrollvertical**=X where X can be Y or N.

Magic xpa 2.2 – New Features, Feature Enhancements and Behavior Changes

Dependency in Mshtml.dll in Rich Client Tasks

The Mshtml.dll file is no longer required to run Rich Client tasks.

If your RIA application needs to use the BrowserScriptExecute() function, then it should include a reference in the CRR to this dll file.

Additional Functions Supported in Rich Client Tasks

The SubformExecMode, MnuRemove and MnuReset functions are now also supported in Rich Client tasks.

Status Bar Panes in Rich Client Tasks

The task mode, wide and zoom indications can be seen in the status bar of a RIA application using a new special setting named **SpecialShowStatusBarPanes**.

Spanish Language Support

Magic xpa now supports a Spanish Language Studio.

The new language was added to the Language list in the installation wizard.

Android™ Support

The Android client was enhanced with the following features:

- Simpler customization methodology
- Repackaging script
- Support device location (GPS)
- Accessing the mobile devices' capabilities (telephone, text messages, emails)
- Support of the Row Highlight Color property for the Table control

iOS™ Support

The iOS client was enhanced with the following features:

- Support of the Row Highlight Color property for the Table control
- Support of the Subform control (Beta feature)

Refer to the **RIA for Mobile Devices PDF** or to the **RIA for Mobile Devices** concept paper available in the Magic xpa Help.

BlackBerry™ Support

The BlackBerry client is now available as part of the installation.

In addition, the BlackBerry client was enhanced with the following features:

- Support of the Line and Group controls
- Support of the Wallpaper form property
- Support encryption of messages between the client and the server (The SpecialClientSecureMessages=N is no longer required.)

Change of behavior:

- Forms with a Floating window type will now open as a popup window. The Title Bar property, which previously defined if the window opened as a popup window, is now supported for full and popup windows and defines whether a title bar will be shown.
- The forms will be automatically scaled according to the device's DPI.

Refer to the **RIA for Mobile Devices PDF** or to the **RIA for Mobile Devices** concept paper available in the Magic xpa Help.

BlackBerry PlayBook™ Support

Magic xpa RIA client can run on the BlackBerry PlayBook tablet using the PlayBook's **BlackBerry Runtime for Android apps**' capabilities.

The script for repackaging an Android 2.3.3 RIA application to BAR file format, which is the compatible file format required for an application to run on the BlackBerry Tablet OS, is available as part of the installation.

Network Installation

The .NET runtime security policy (by default) disables code from running if it exists on a network drive. To run Magic xpa from a network drive you need to adjust your security policy.

This can be done via the Microsoft .NET Configuration tool or by running the following command:

```
c:\Windows\Microsoft.NET\Framework\v2.0.50727\caspol.exe -machine
-addgroup All_Code -strong -file <path>MgxpaRuntime.exe -noname -
noversion FullTrust -name Magicxpa_Assemblies_Access -description
"Code group granting trust to Magic xpa assemblies"
```

This command needs to be executed only once in each of the computers.

You can also add a **–silent** key for a silent installation.

Mobile Application Samples

A sample project named **Mobile Demo** was added to demonstrate how you can write mobile applications using the mobile device capabilities of Magic xpa's RIA technology.

Magic's Company Rebranding: A New Look for a New Age of Magic

Magic is launching its rebranding as a proactive strategic move to reflect the exciting changes that we are undergoing as a company and the major developments in our product offering.

Magic's rebranding highlights our company values, the set of beliefs that guide and inspire us at every level throughout our organization.

For our brand architecture—the framework that defines the relationship between the corporate brand and the product portfolio—we have chosen to use the 'master brand' model, focusing more on our company brand (Magic) rather than on the branding assigned to our different products and services. This emphasizes that Magic's different products are all instances of the same unified technology stack and the same fresh approach to enterprise software.

Accordingly, we have renamed our products as follows:

- **uniPaaS** is now named **Magic xpa Application Platform**
- **iBOLT** is now named **Magic xpi Integration Platform**

Magic's rebranding aims to illustrate that all our corporate activities and communications are infused with our core values, differentiating us from competitors, strengthening our relationships with customers, and increasing the popularity of our products and services.

Renaming of Product Executables

As part of the rebranding process, some of the product files were renamed, including:

- **uniStudio.exe**, which is now called **MgxpaStudio.exe**
- **uniRTE.exe**, which is now called **MgxpaRuntime.exe**
- **uniRQBroker.exe**, which is now called **MgBroker.exe**
- **uniRQMonitor.exe**, which is now called **MgBrokerMonitor.exe**

uniPaaS 2.1 – New Features, Feature Enhancements and Behavior Changes

Android and iOS Support

The uniPaaS RIA client is now capable of running on Android™ and iOS™ devices.

As for the Windows Mobile and BlackBerry, the Magic RIA client for Android and iOS is a native operating system (OS) application implementing the Magic RIA client protocol. Using the Magic RIA client for the different mobile devices, developers can deploy enterprise connected, highly interactive RIA applications on the different mobile devices.

Note that the RIA client for Android is a Beta version.

Refer to the **RIA for Mobile Devices PDF** or to the **RIA for Mobile Devices** concept paper available in the Magic xpa Help.

Call to a Destination Subform or Frame in Online Tasks

The Call operation in Online programs was enhanced with a new property called 'Destination'. This property provides the ability to dynamically call a program or a task and run it in a subform or a frame (similar to Rich Client).

Retain Focus in Online Tasks

The Call operation in Online programs was enhanced with a new property called 'Retain Focus'. This property defines whether the focus will remain on the current control or be moved to the first control of the called program or task after executing the Call operation.

This property is enabled when performing the call from a logic unit other than the Task, Record, or Control logic units to a destination subform or frame using the Destination property.

Parallel Execution in Online Tasks

Parallel execution is now supported for Online tasks.

The Application Modal window type was removed and the Modal window type will behave as the old Application Modal.

The Window menu and window-related events are still not supported.

View Refresh in Rich Client Tasks

The View Refresh behavior in Rich Client tasks was enhanced. After performing a view refresh with 'Relocate Mode=0', the position of the current record now remains as-is (and is not changed to be the first record).

Row Placement Property in a Table Control

A new property was added to support the fixed number of table records.

Setting this property value to Yes means that the row's height will be resized according to the table's height, thus keeping the same number of records as designed in the Studio.

Default Location of Forms

A new option named Windows Default Location was added to the form's Startup Position property. When using this option, the form is positioned at the Windows default location and has the dimensions specified in the form's size.

The Default option in which the form is positioned at the Windows default location and has the bounds determined by the Windows default was renamed to Windows Default Bounds.

.NET Data Binding

It is now easier than ever to use uniPaaS data variables with .NET controls.

Two new properties were added to the .NET control model so that you can define the .NET control property to which you want to bind the data and the event that will be the trigger for the data update.

After these properties are defined, you can use the .NET control in the same way as you use any uniPaaS built-in control. You simply need to attach your data variable to the .NET control's Data property.

The previous functionality of assigning a .NET variable to the .NET control is still supported and can be done by using the .NET Object property in the .NET control.

For more information on how to use .NET data binding, see the .NET Tutorial sample installed with uniPaaS.

.NET – DataViewToDNDataTable() Function

A new function was added to create a .NET DataTable object out of the task's data view.

The .NET DataTable objects can then be used as the data source of other .NET objects.

.NET Controls – Change of Behavior

When clicking on a .NET control defined with Allow Parking = No, the focus will not leave the current control or record.

Requester and Broker Logs – Default File Names

The default value for the requester and broker logs were changed as follows:

mrb_event.log => BrokerActivity.log, mrb.log => Broker.log, req.log => Requester.log

Mobile Devices – Additional Information

A new property was added to allow maximum flexibility with mobile device properties.

This property will be used to send specific pre-defined information to the mobile devices.

Mobile Devices – Press Event

A new event was added to handle the long press on a control in mobile devices. At this stage, this event is supported for BlackBerry only.

.NET 3rd Party Samples

A new sample project was added to demonstrate how you can use 3rd party .NET objects and controls in your application to improve the functionality and user interface.

The 3rd party assemblies are not included with this package and should be downloaded from the 3rd party vendor site as described in the sample programs.

Note: If your 3rd party package version is different than the one used in the samples, you will have to replace the assemblies defined in the CRR with the ones from your version.

uniPaaS 2.0a – New Features, Feature Enhancements and Behavior Changes

MDI Frame Color

A new color was added to the default color file with the value corresponding to an MDI Frame background.

This color is used as the MDI Form color in newly created projects (and can be changed in the proper ties of the Main Program form).

End-User Functionality Component

The End-User Functionality component is now added by default to newly created projects.

A new option was added in the migration wizard, which lets you select whether to also add this component to the migrated project.

Note that if you want to use both this component and the Report Generator, then the Report Generator component should be defined above the End-User Functionality component.

Subform Behavior

A new property named Refresh When Hidden was added to the Subform control in Online tasks.

This property provides the ability to load and refresh the task running in the Subform control the same way it was done in uniPaaS 1.9.

A value of Yes means that the subform tasks will be called for the first time after the Record Prefix of the host and again after each refresh of the subform. (This is similar to the behavior in uniPaaS 1.9 when the subform was attached to a container control and the container control was hidden.)

A value of No means that the subform tasks will be called only if the subform is visible. (This is similar to the behavior in uniPaaS V1.9 when the subform had a Visible expression.)

For nested subforms defined with a value of No, if the parent subform has a value of Yes at runtime, then the nested subform wi ll also be executed with a value of Yes.

The migration from uniPaaS 1.x will set the value of this property as follows:

- No – When the subform is attached to a container control.

- Yes – When the subform has a Visible expression.

- Exp – When the subform is attached to a container control and also has a Visible expression. The expression is the expression of the Visible property.

In addition, up until this version, when clicking on a Subform control, the Control Prefix and Control Suffix of the first control were always executed. This is now fixed, so only the Control Prefix of the clicked control will be executed.

EDP File Structure

The EDP file was changed a bit to prevent opening a uniPaaS 2.0 project in previous versions, in order to maintain the integrity of the sources.

uniPaaS 2.0 – New Features, Feature Enhancements and Behavior Changes

Prerequisites

uniPaaS is a native .NET application, so in order to run uniPaaS on your machine, you must have .NET framework installed on your machine with one of the following configurations:

- Both .NET Framework V2.0 SP1 and Microsoft Visual C++ 2008 Redistributable.

  You can install them by running the NetFx20SP1_x86.exe and VCRedist_x86.exe files from the Scripts\RIA folder.

- .NET Framework V3.5 (or above)

Network Installation

The .NET runtime security policy (by default) disables code from running if it exists on a network drive. To run uniPaaS from a network drive you need to adjust your security policy.

This can be done via the Microsoft .NET Configuration tool or by running the following command:

c:\Windows\Microsoft.NET\Framework\v2.0.50727\caspol.exe -machine -addgroup All_Code -strong -file <path>uniRTE.exe -noname -noversion FullTrust -name uniPaaS_Assemblies_Access –description "Code group granting trust to uniPaaS assemblies"

This command needs to be executed only once in each of the computers.

You can also add a –silent key for a silent installation.

Online Changes

In uniPaaS 2.0, the Online GUI is based on the .NET framework instead of Win32.

The .NET framework does not support all the appearances and control behaviors that were supported in previous versions, so there is a change in the uniPaaS Runtime appearance as well.

For a full list of changes, refer to the What's Different in uniPaaS 2.0 help topic.

Some of the major enhancements are described below.

.NET Integration in Online and Batch Tasks

.NET Integration was introduced to RIA applications in uniPaaS V1.8.

With this release you can enhance your Client/Server application offering by easily embedding and integrating any .NET control or assembly in your Online and Batch tasks.

For more information on how to add and manipulate .NET modules, see the **.NET Tutorial** sample installed with uniPaaS.

Online MDI

The Online Runtime MDI Frame is now defined as any other form in the Main Program (similar to the Rich Client MDI Frame).

The benefit of this change is the centralizing of all the form properties from the different locations they were once defined in, into a single location.

The MDI can be disabled by defining Open Task Window = No in the Main Program properties.

Online Frameset

Splitter forms are now defined using a frameset form for an Online task (as in Rich Client tasks) instead of using a Splitter form.

User State Persistency in Online Tasks

The Online task's form state persistency is now defined as a simple Yes/No property.

This change makes it simpler to use the form state persistency since you now do not need to define and maintain a list of unique identifier names for your forms.

Rich Client – Hebrew Support

The Logical() and Visual() functions are supported.

Rich Client – Forms

All UOM types (dialog units, centimeters, inches) are supported.

Rich Client – Drag and Drop

1. Drag & drop is supported for all of the controls.
2. User-defined format is supported.
3. DragSetCrsr() function is supported.

Rich Client – Post Refresh by Parent Event

A new internal event was added to execute logic every time the subform is refreshed.

This event is raised before the Record Prefix in the subform task when the subform is refreshed by the parent. The event is raised when one of the following occurs:

1. The subform is defined with Auto Refresh=Y and a variable sent to the subform as a parameter is changed.
2. The Subform Refresh event is raised.
3. The subform is executed for the first time.

The event is not raised when a 'View Refresh' is raised in the subform task.

Sample Projects

Additional sample programs were added to the Online Samples and Rich Internet Samples projects. The new sample programs are for some of the uniPaaS functions and for connectivity to Google Calendar, Google Blogger, Facebook, LinkedIn and Twitter.

uniPaaS 1.x Release Notes

For information about uniPaaS 1.x releases, see the PastReleaseNotes.pdf file.