

Magic xpa 2.2b Release Notes



OUTPERFORM THE FUTURE™

Fixed Issues

- 166025 – The Runtime engine crashed when calling a component program that had a variable updated with the result of the Menu() function.
- 274946 – The task's allowed modes condition was not refreshed upon recompute.
- 278301 – The menu was not refreshed according to the user's rights when the Logon function was used in a program that was called from the Task Prefix of the Main Program.
- 287710 – Expressions returned negative numbers instead of Null values in a specific scenario.
- 288050 – A Call by Name operation to a cabinet file with a long path failed.
- 297438 – The Runtime engine crashed when hiding the menu and immediately showing it using the MnuShow() function.
- 305892 – A variable assigned with a Null value showed the default value and not the Null one.
- 438486 – The Runtime engine crashed after migrating a uniPaaS application that had a program with a Table control placed on a Tab control.
- 983489 – The placement value of controls was set to more than 100% after migrating a uniPaaS application that had controls linked to a Group control with placement attached to both the Group and the linked controls.

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.

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 **SpecialShowStatusBarPanels**.

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 -  
noverision 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 **MgxpStudio.exe**
- **uniRTE.exe**, which is now called **MgxpRuntime.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 will 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.

Magic Software Enterprises Ltd provides the information in this document as is and without any warranties, including merchantability and fitness for a particular purpose. In no event will Magic Software Enterprises Ltd be liable for any loss of profit, business, use, or data or for indirect, special, incidental or consequential damages of any kind whether based in contract, negligence, or other tort. Magic Software Enterprises Ltd may make changes to this document and the product information at any time without notice and without obligation to update the materials contained in this document.

Magic is a trademark of Magic Software Enterprises Ltd.

Copyright © Magic Software Enterprises, November, 2012

