Magic xpa 4.11.1

For Linux 64-bit

Compatibility Guide

&

Release Notes

Contents

We are delighted to announce the release of Magic xpa 4.11.1 Application Platform for Linux with new features and enhancements.

You can review the contents of Magic xpa 4.11.1 through the below link:

http://www2.magicsoftware.com/ver/docs/Downloads/Magicxpa/4.11.1ReleaseNotes.pdf

Compatibility

Application Servers

Red Hat Enterprise version 8.10 , 9.0,9.4.

IMM middleware - Red Hat Enterprise version 8.10, 9.0, 9.4.

Web Servers

Apache 2.4

Databases

- Oracle 12c
- Oracle 18
- Oracle 19
- Microsoft® SQL Server 2019 using ODBC Driver 17
- MySQL 5.x with ODBC Driver 5x
- MySQL 8.x with ODBC Driver 8x
- DB2 UDB 11
- DB2/400
- AS/400
- PostgreSQL 13
- PostgreSQL 16

Middleware, Frameworks, and Protocols

- IBM WebSphere® MQ 8 client/server – for messaging capabilities
- JMS – requires JMS API 1.3 Client

- JRE 8.0, JRE 11.0 – for Java integration capabilities
- LDAP V.2 and V.3 using simple bind
- TLS 1.2 for Java HTTP framework

Support for 64-bit Server Engine

Magic xpa for Linux comes with a 64-bit server engine supported in background mode in order to take utmost benefit of a 64-bit operating system.

Note: The 32-bit version of the product is no longer available.

Magic xpa Licensing

In addition to the new unitary development and deployment paradigm for RIA and SaaS, Magic xpa also supports any previous Magic Software editions and forms of development and deployment. However, in order to maintain your former development and deployment capabilities, you need to obtain new Magic xpa licenses that reflect your current licenses.

To obtain Magic xpa licenses, please contact your local Magic Software representative.

Installing Magic xpa

To install Magic xpa for Linux, you need to consider the following points:

Pre-Installation

Back up the following files before starting the installation process:

- .cshrc,
- .profile,
- .bash_profile

Installation Steps

1. Create a new user. (You must perform installation using a non-root user.)
2. Log in as the new user.

3. Uncompress the installation file (**magicxpa_<Version>.<platform>.tar.gz**) using the local uncompress utility or a compatible utility, such as gunzip.

4. Run the command from **$HOME directory:**

   ```
   Tar xvf <installation file>
   ```

   The installation file name is **magicxpa_<Version>.<platform>.tar**.

5. Run the following command:

```
./magicxpainstall
```

6. Enter the requested information. Note that the Java home directory is required.

7. Choose the required Messaging Infrastructure. Note The IMM (In Memory Middleware) is supported only for Red Hat 8.0 and 9.0.

   a. If you choose to use **IMM** as the messaging infrastructure, the installation procedure updates the relevant configuration files accordingly in **InMemoryMiddleware** directories.

      Note: On Linux systems, the Tomcat server is used to host the HTTP requester. Follow the Windows installation explanation regarding those products with one difference – the scripts used have the **.sh** and not the **.bat** suffix.

   b. If you choose **Broker** as the messaging infrastructure, follow the given steps:

      i. Run the **$HOME/sbin/mgroot.sh** file as a root- user after the successful installation. This script copies Magic xpa files that should be accessed by your Web server

      ii. Follow the steps for the Apache Web server configuration.

8. Set up an Apache Web server as follows:
   a. Append the $HOME/web_utils/magic.conf file to the Apache configuration file (httpd.conf)
   b. place the requester in the modules directory
   c. Restart the Apache Web server (see Apache Requester Installation and Configuration).
9. Log-out from the new user.

10. Log-in again to activate the new environment settings.

11. If you need to uninstall the product, delete the user home directory created for the installation.
    Note: For complete removal, delete the files copied by the **$MAGIC_HOME/sbin/mgroot.sh** script and remove the changes that were applied to the Web server.

Post Installation

It is necessary to define for Magic xpa that a specific gateway must be loaded by pointing to a variable that contains a DB number. The DB number points to a specific executable that is the relevant gateway.

In Linux operating systems, an environment variable points to the executable, which should be used for a specific gateway, located at **$HOME/etc/mgenv**.

For example, in Linux: **MAGIC_ DB_ 14_ DRIVER=$HOME/bin/mgoracle12**
where the number 14 refers to the DB number.

Note:	◦ If the installation fails, it is best to delete all files in **$MAGIC_HOME** and start a new installation from the beginning. ◦ A remarked entry (meaning that it's preceded by a semicolon) named **MGLOCAL**, which points to the Korean support library **mglocal.kor**, exists in the **$HOME/etc/mgenv** file. This file is required for proper functioning of browser-based applications when using Korean/Hangul characters.

Installation Components

1. Magic xpa Server (**bin/mgxparuntime**)
2. Magic xpa Broker (**broker/mgbroker**)
3. Magic xpa command line requester (**broker/mgrqcmdl**)
4. Magic xpa gateway 12c, 18, 19(**bin/mgoracle12, bin/mgoracle18, bin/mgoracle19**) – versions are platform specific
5. Magic xpa gateway for DB2 UDB Version 11.1 (**bin/mgdb2**)
6. Magic xpa gateway for ODBC (**bin/mgodbc**)
7. Magic xpa memory gateway (**bin/mgmemory**)
8. Magic xpa Postgres gateway (**bin\mgpostgresql**)
9. Magic xpa SQLite gateway (**bin/mgsqlite**)
10. Magic xpa DB2/400 gateway (**bin/mgdb400**)
11. Magic xpa ISM/400 gateway (**bin/mgeac**)
12. Magic xpa SQLServer gateway based on ODBC **(bin/mgsqlserver)**
13. In Memory Middleware

14. OpenJDK 8
15. Magic xpa CGI requester (**cgibin/mgrqcgi411**)
16. Magic xpa requesters (64-bit only) for Apache Web Server (**cgibin/mod_V2.2_mgrequest411.so, cgibin/mod_V2.4_mgrequest411.so**)
17. Magic xpa UDF/UDP examples (**userproc directory**)
18. Magic xpa Web utility files used for Browser Client support (**web_utils directory**)
19. Magic xpa Hangul support (**language/mglocal.kor**)
20. Magic xpa Messaging component (**messaging/messaging.mff**)
21. Web Service provider installations: Apache Ant
22. apache tomcat 9.0.20 (http://www.apache.org/).

Starting a Magic Project Using IMM

1. Install Snap and microk8s according to the following link: https://snapcraft.io/install/microk8s/rhel
2. Execute the following command to start Controller, Web Server, Redis, and IMM Monitor.

./deploy-imm.sh with the parameter ..//config//**projectsStartup.json**.

While starting, Controller reads the data from **projectsStartup.json**.
You can also start IMM without **projectsStarup.json** parameter and later start Magic xpa engines using **StartProjects.sh.**

3. Check the status of all the pods for running by executing the following command:

```
kubectl get pods -n magic-xpa-imm-ns
```

4. The running pods can be seen as given below:

```
e:\Magic\Magicxpa49\InMemoryMiddleware\deploy>kubectl get pods -n magic-xpa-imm-ns
NAME                                          READY   STATUS    RESTARTS   AGE
imm-cluster-controller-deployment-5fdbdbfbf4-wnxx9   1/1     Running   0          2m14s
imm-monitor-deployment-b47ccb48c-kkp6z        1/1     Running   0          2m14s
imm-ws-deployment-5ff84846bd-5hfcp            1/1     Running   0          2m14s
mongo-deployment-5dc8cd6749-8ncqf             1/1     Running   0          2m14s
redis-deployment-0                            1/1     Running   0          2m14s
```

5. Start IMM Agent by executing the /InMemoryMiddleware/agent/**mg-imm-agent**

Starting a Magic Project Using Broker

To run the Magic xpa Server follow the given steps:

1. Use the **mgxparuntime.sh** script or invoke the **mgxparuntime** executable file. By default, Magic xpa uses the INI file specified in the **MGENV** environment variable.

For the non-default INI file, use: **mgxparuntime –ini=<ini file> &**

For an additional INI file, use: **mgxparuntime –ini=<ini file> @<additional ini> &**

Several scripts exist in the **sbin** directory to simplify the Magic xpa server administration:

startb	Start the Magic xpa Broker
stopb	Stop the Magic xpa Broker (and Server engines)
stopm	Stop all Magic xpa Server engines connected to the Magic xpa Broker
checkm	Check which Magic xpa Server engines are connected to the Magic xpa Broker

The **stopb** and **stopm** scripts require supplying the broker password, as shown in the following example:

```
stopb -password=<broker supervisor password>
```

Note:	The Magic xpa log file is created for each server you start. Its name is determined by the **ExternalLogFileName** entry in the MAGIC.INI file.

Magic xpa Requesters

The **MGREQENV** environment variable points to the **MGREQ.INI** file used by the Magic xpa Server, the Magic xpa Broker, and the Magic xpa command line requester. (The installation sets Magic xpa Requesters for Linux: **MGREQENV = $HOME/etc/MGREQ.INI)**.

To send a request to a Magic xpa Server on Linux from an Internet Browser, there are two types of requesters:

1. The Magic xpa CGI requester (mgrqcgi411): http://<server_name>/cgi-bin/mgrqcgi411?appname=example1&prgname=prog1
2. The Magic xpa requester for Apache (mod_V2.2_mgrequest411.so): http://<server_name>/mgrequest411?appname=example1&prgname=prog1

Additional Settings

The following settings in the **MGREQ.INI** file affect the requester execution.

3. RetryMainTime
4. KeepAlive

Apache Requester Installation and Configuration

The installation includes the requester **mod_V2.4_mgrequest411.so** to be used with Apache 2.4.

Apache Module Requester Setup

Magic xpa 4.11.1 includes requesters for Apache Web Server version 2.4.

1. Make sure the requester module **mod_V2.4_mgrequest411.so** is placed in the **/modules** directory of Apache installation (default: **/usr/local/httpd/modules**) with execute permissions.
2. Add the following lines to the Apache configuration file, **httpd.conf**.

```
LoadModule mgrequest411_module   modules/mod_V2.4_mgrequest411.so
<Location /mgrequest411>
   SetHandler mgrequest411-handler
</Location>
SetEnv MGREQ_INI_PATH  <directory>
```

3. Restart the Apache Web server.

Apache requester is configured using the **MGREQ.INI** file. The directory location of the **MGREQ.INI** file is specified by the **MGREQ_INI_PATH** setting in the Apache configuration file, **httpd.conf**.

For example,

```
SetEnv MGREQ_INI_PATH /usr/local/httpd/conf
```

Apache requester uses the **/usr/local/httpd/conf/ MGREQ.INI** file.

To use this requester, perform the following steps:

1. Call Magic xpa using a URL, such as:

```
http://server/mgrequest411?appname=...
```

2. Modify the **MAGIC.INI** file as follows:

```
InternetDispatcherPath=/mgrequest411
```

Using Apache Web Server with a Non-Default Port

To use Apache with a non-default port (port number other than 80), change the setting given below in the **MAGIC.INI** file:

Set InternetDispatcherPath= http://server:port/cgi-bin/mgrqcgi411

instead of /cgi-bin/mgrqcgi411.

File Names

- In Windows platforms, files can be referred to by either a URL or by file name, relatively or by full path/URL.
- In non-Windows platforms such as Linux, files can be referred to by a full URL only. Any reference to a file name with a fore slash (/) is considered to be a path name, either full or relative.

For example,

- **/etc/home1/a.jpg** (full path)
- **http://myserver/myalias/a.jpg** (full URL)
- **myalias/a.jpg** – a relative path name; not a relative URL.

Colors

To use colors properly on Linux platforms, you must define all the colors that are used as non-system colors. The easiest way to do this is to access the color file in the Magic xpa Studio and define the colors accordingly.

External Code Pages

- While installing Magic xpa on Linux platforms, the **ExternalCodePage** ini setting is set to **1252** (Windows 1252 is the Western European code page). This setting must be modified for any-non Western European languages, such as Hebrew or Thai, since it affects Unicode to ANSI conversions.
- Further to this, the following special flag in the MAGIC.INI file , must also be set to the same code page: **[MAGIC_SPECIALS]SpecialInternalNonUnicodeCodepage**

Platform-specific Information

Linux

- For Intel processors only, Linux requires Kernel 3.10.0-957.1.3.el7.x86_64 and up with glibc-2.17-260 and up.
- The Magic xpa 4.11.1 Server for Linux should be used with Red Hat Enterprise Linux version 8.0 and 9.0.
- The IMM (In Memory Middleware) is supported only for Red Hat 8.0 and 9.0.
- The Oracle gateway should be used with the Oracle 12 client and above.
- The Magic xpa DB2 gateway should be used with the DB2 Version 11.1 client.
- The Websphere MQ 8.x client/server is required for working with the MQ messaging capabilities.
- JRE 1.8 is required for working with Java integration capabilities.
- Apache 2.4.20 (or a more recent version) is required in order to use the Apache 2.2 requester.

Gateway-specific Information

- To enable the use of a particular gateway, remove the # sign from the corresponding entry in the **$MAGIC_HOME/etc/mgenv** file.
- While using Oracle gateway, make sure that **ORACLE_HOME** and **ORACLE_SID** are set in the **$MAGIC_HOME/etc/mgenv** file, and the environment variable **LD_LIBRARY_PATH** (or **LIBPATH** for AIX) includes the **$ORACLE_HOME/lib** directory.
- While using DB2 gateway, make sure that **DB2INSTANCE** is set in the **$MAGIC_HOME/etc/mgenv** file.

ODBC Gateway

The following information pertains to ODBC gateway:

General Information

- **Gateway name:** mgodbc
- **Required software:** This gateway works with the UnixODBC ODBC manager.
- It was tested with the following database gateways:

  MySQL MyODBC driver (libmyodbc-<ver>.so) – to access MyODBC software.

  For more information on this particular driver refer to http://www.mysql.com.

Installation and Setup Instructions

1. Uncomment the entry **MAGIC_DB_20_DRIVER** in the **mgenv** file. (Uncommentting means removing the semicolon preceding the entry.)
2. Install **UnixODBC** ODBC manager. You can download this product from: http://www.unixodbc.org.
3. Follow the online instructions to generate an ODBC manager.
4. Locate the following two files (shared libraries): **libodbc.so.2.0.0** and **libodbcinst.so.2.0.0.**
5. Copy the files to the directory **$MAGIC_HOME/lib**.
3. Install ODBC driver by referring to the specific driver documentation for installation instructions.
4. Make sure that the libraries have Execute permission. Use the **chmod +x** command to set execute permission.
5. Create a hidden file named .**odbc.ini** in the user's **/home** directory.
   For example, **/usr/magicadm/.odbc.ini**. This file is used to configure ODBC DSNs. Refer to the ODBC manager documentation for more explanations regarding the setup of this file.
6. To help you set up quickly, we have included the following **.odbc.ini** file as an example:

```
[mysql]
Driver   = /usr/lib/libmyodbc5.so
Trace    = No
Tracefile= mysql.log
Database = samp_db
```

Each section defines a Data Source Name (DSN). In the above example, there is a DSN defined named **mysql**.

7. Set the driver entry in each section to the full path of the ODBC driver. For a list of valid entries and their meanings, refer to the ODBC driver documentation. Alternatively, a general **/etc/odbc.ini** file can be used.

Setting the Magic Configuration File (MAGIC.INI)

Follow the given steps to set Magic Configuration file:

1. Set a Magic xpa database using the Database repository.

2. Copy the database definition in the **MAGIC_DATABASES** section from the MAGIC.INI file on Windows to the MAGIC.INI file on Linux. It is highly recommended to back up the MAGIC.INI file before editing.

Limitations and Recommendations

JMS

Connectivity to messaging servers via JMS is not supported using the provided Messaging component. Before you use JMS with the Sun Reference application, find the environment variables listed below to run J2EE applications on Linux platforms:

Variable Name	Values
$JAVA_HOME	Directory where the Java 2 SDK, Standard Edition, is installed
$J2EE_HOME	Directory where the J2EE SDK is installed
$CLASSPATH	Include the following: .;$J2EE_HOME/lib/j2ee.jar; $J2EE_HOME/lib/locale
$PATH	Include $J2EE_HOME/bin

Backups

We highly recommend backing-up Magic xpa configuration files, such as **MAGIC.INI**, **MGRB.INI**, **MGREQ.INI**, and **license.dat**, before modifying them.

Compression

There is no compression when the server is a Linux platform.

Java Integration

- The Java CLASSPATH separator character on Linux platforms is a colon (:) as opposed to the Windows platform separator character, which is a semicolon (;).

  For example, **CLASSPATH = /java/MyClasses:/java/OtherClasses**

  For more information, please refer to the Java documentation (Java 2 SDK Tools and Utilities at: http://www.oracle.com/technetwork/java/javase/documentation/index.html).
- If Java is installed on your server, you should edit the scripts: .**cshrc** and **.profile**.
- The **LD_LIBRARY_PATH** environment variable should include the directory where **libjvm.so** resides within the existing Java 1.8 installation.

WebSphere MQ

- If you are using an **MQ client** software, you should set the following logical name in the **MAGIC.INI** file: **WMQ_ModuleName = C**.
- If you are using an **MQ server** software, meaning that the MQ Queue manager runs on the same machine as the Magic xpa Server, you should set the following logical name in the **MAGIC.INI** file: **WMQ_ModuleName = S**.

LDAP

- The LDAP library used with xpa 4.11.1 has been switched to **OpenLdap**.
- When using SSL with LDAP the search path for the location of the certificates has changed.
- One of these files needs to contain the information as to where the certificates are located.

```
/etc/openldap/ldap.conf
$HOME/ldaprc
$HOME/.ldaprc
```

- Place the information into any one of these files:
  SASL_NOCANON on
  TLS_CACERTDIR /tmp/LDAP (as an example)
  TLS_CACERT /tmp/LDAP/CA.LDAP.cer (as an example)

External Procedures

User-defined procedures should be compiled according to this platform specific list:

Platform	Compiler Version & Vendor	c++ Compiler	c Compiler
Linux	gcc version 4.8.5	g++	gcc

Fully Qualified Domain Name (FQDN)

An FQDN stands for fully qualified domain name, for example: **linuxdev.Magic**.

Broker and enterprise server should bind using a specific network adapter by specifying a FQDN (instead of IP address). The requester layer should translate the FQDN to IP and bind using IP on a specific adapter.

The **MGREQ.INI** file contains the following entry: **BindFirstIPAddress=Y/[N]**.

- Y – During binding to a port, the server resolves the host name and binds to the resolved IP address.
- N – The server binds to any IP address (*.port – for backwards compatibility)

To enable a Magic xpa engine and Broker to work with a specific network adapter (if there are multiple adapters present on a machine) follow the given steps:

1. Edit the **MGREQ.INI** file
2. Set **BindFirstIPAddress**=Y.
3. Set **MessagingServer** to **FQDN/port**.
4. Edit the **MGRB.INI** file.

5. Set **MessagingServer** to **FQDN/port**.
6. In the **MAGIC.INI** file, set **TCP/IP=2, 30, 1500-2000 /LocalHost=FQDN**.
7. In the **MAGIC.INI** file, set **Default Broker** to **FQDN/port**.

The table below shows the binding for the server module:

Port Number	Binding
Port-No	BindFirstIPAddress=N */Port-No BindFirstIPAddress=Y IP-address/Port-No
Ip Address/Port-No	IP-Address/Port-No

Deploying a Rich Client Application

To be able to deploy a Rich Client application on Linux platforms follow the below steps:

1. The following files and folders are created once you use the Rich Client Deployment Builder:

appname\appname.application

appname\appname.publish.html

appname\mgxpaRIA_x_y_z_www (x,y,z represent the Magic xpa version and www is a unique number representing the specific version)

appname\Images\

2. Place them in the **MagicRIAApplications/*appname*** alias on the Web server.
3. Users can access the application from the following URL: `http://appserver/MagicRIAApplications/appname/appname.publish.html`

4. Add the following into the **httpd.conf** Apache configuration file in this order:
   - AddType application/x-ms-application .application
   - AddType application/x-ms-application .manifest
   - AddType application/octet-stream deploy
   - AddType application/x-msdownload .dll
   - AddHandler default-handler .jpg .gif .js .txt .bat .msi

5. Manually change the **HTTPCompressionLevel** in the application's **publish.html** file to **None**, since there is no compression when the server is a non-Windows platform.
   For example,

```
<body onload="initialize()">
        <xml id="rcExecProps">
             <properties>
            <property key="protocol" val="http"/>
            <property key="server" val="linux73:2261"/>
            <property key="requester" val="/mgrequest411"/>
            <property key="appname" val="frame"/>
            <property key="prgname" val="START"/>
            <property key="arguments" val=""/>
            <property key="envvars" val=""/>
            <property key="UseWindowsXPThemes" val="Y"/>
            <property key="HTTPCompressionLevel" val="None"/>
            <property key="DisplayStatisticInformation" val="N"/>
            <property key="InternalLogLevel" val=""/>
            <property key="InternalLogFile" val=""/>
            <property key="InternalLogSync" val="Session"/>
<property key="LogClientSequenceForActivityMonitor" val="N"/>
             </properties>
        </xml>
    <table align="center">
```

Deploying a Web Client Application

To be able to deploy a Web Client application on Linux platforms follow the given steps:

1. Run “ng build” when the current directory is **..\output\src**. A **dist** folder will be created under **output\myWCapp** (where **myWCapp** is your Web Client project name)
2. Define the following aliases in httpd.conf located in /etc/https/conf:
   a. MagicWebCCache pointing to the Web_Client_Cache directory

b. MagicWCApplications pointing to a directory holding the web client applications.

c. Magicxpa pointing to the cgibin directory

3. Modify the HTML tag in index.html located under output\myWCproject\dist\myWCapp so that **<base href="/">** will now be **href=/MagicWCApplications/myWCapp/>**
4. Change the **server-config.json** file located under the assets folder to point to your protocol, server, requester and application name
5. Copy the directory under the **dist** folder to a location under web server under **MagicWCApplications**
6. Start the xpa server using an MGWEB license.
7. Access the application through a URL as : http://myserver:8022/MagicWCApplications/myWCapp

V4.11.1 features

OpenSSL library upgrade

The OpenSSL libraries were upgraded to OpenSSL 3.0 LTS, namely 3.0.15.

Note: xpa 4.11.1 was created on Red Hat 8 and uses some Red Hat 8 system libraries, which create nested dependencies. These lead to the need to add both libssl.so.1.1 and libcrypto.so.1.1 files to /usr/lib64/.

Studio Services

Command Line Utility for Executing background Operations

The utility named **mgxpastudioservices** now supports the “Checker” option.

Syntax: mgxpastudioservices -AutomaticProcessingSequenceFile=<ProcessingFilename> -AutomaticProcessingLogFile=<LogFilename>

Limitations for current version:

Font and color – references to fonts/colors by expression show up as unused. References to nonexistent colors are not reported.

.NET – Applications with .NET variables/snippets are not supported

Applications with OLE or ActiveX variables are not supported

Java - Applications with .Java variables are not supported

See the xpa 4.11 help for this utility’s syntax.

V4.11 features

Support for Security Enhanced Linux (SELinux)

See http://www2.magicsoftware.com/ver/docs/xpaproduct/SELinux/SELinux Configuration for Apache.pdf for more details on configuring SELinux.

IMM Enhancements

The following feature is enhanced with respect to In-memory Middleware.

Support of REST API to Execute Batch Programs

Magic xpa provides you a way to use **REST API** and enable your application to execute Batch programs. You can define a list of Batch programs that you want to expose along with their arguments in an XML file and deploy REST API as a pod either on-premise or within a cloud-native service. The deployed REST API then converts REST calls into Magic xpa standard requests.

Magic xpa supports:

1. two Authentication methods with REST API:
    - Basic
    - OAuth 2.0.

2. returning an HTTP Response Code using **RqHTTPStatusCode()** function.

For more details, please see **How to - Use REST API to Execute Batch Programs.pdf** in **\Support** folder provided via the Windows server installation**.** Also please refer to the sample program located at <Magic_xpa_Installation>\InMemoryMiddleware\test\RestAPISample.

DownscaleThresholdTime

A parameter named DownscaleThresholdTime is introduced for downscaling an idle Worker or a Server. It is the time (default 60 minutes) required for a Worker or a Server to be idle until it stops. You can set the desired time as required.

Studio Services

Command Line Utility for Executing background Operations

Magic xpa introduces a utility named **mgxpastudioservices**, which executes the following operations in background in a deployment environment:

- Creation of ECF
- Importing an existing project's XML

This utility executes independent of Magic xpa Studio executable.

Support for PostgreSQL 16

Magic xpa was tested and is compatible to work with PostgreSQL 16.

V4.9 features

New Middleware

Magic xpa supports In-memory Middleware. In-memory Middleware is an underlying messaging layer of Magic xpa implemented on In-memory Data Store. The new middleware supports:

- Scalability - Processing power can be scaled up easily by spawning new XPA engines with the help of IMM controllers and agents
- Robustness - Under the stress of heavy load of xpa requests the middleware system performs efficiently and ensures good response time for every request
- High Availability - High availability is achieved with Kubernetes cluster orchestration.

Note: Magic xpa discontinued the support of GigaSpaces middleware.

Support of PDF Printing

Magic xpa now enables PDF file printing with the following printable items:
Printing of the following controls is supported:

- Label
- Image with Distorted Scaling image style
- Edit
- Line
- Ellipse
- Table

Printing of the following controls is not supported:

- Group control
- Rich Text Label
- Rich Text Edit

Printing of the following items is not supported:

- Forms set with property ‘Form Units’ to Dialog units
- BMP images
- Text-based forms
- Image Styles Copied, Scaled to Fit, Scaled to Fill, and Tiled
- Windows system colors

Limitations

- Magic xpa colors which are Windows system colors need to be converted to non-system colors. The default color is Cyan.
- Microsoft-specific fonts are required to be installed on Linux into the /usr/share/fonts/ folder. These fonts appear slightly different in the PDF printed on Linux.

Note:
1. Magic xpa generates messages into mgerror.log file for the items that cannot be printed into the PDF.

2. The Form type used for defining the content to be printed as PDF should be GUI > 0 and PDF = 'Yes' in I/O properties.

http://www2.magicsoftware.com/ver/docs/downloads/magicxpa/4.9releasenotes.pdf for more information regarding Magic xpa 4.9 features.

V4.8.1 features

See http://www2.magicsoftware.com/ver/docs/downloads/magicxpa/4.8.1releasenotes.pdf for more information regarding Magic xpa 4.8.1 features.

V4.8 features

See http://www2.magicsoftware.com/ver/docs/downloads/magicxpa/4.8releasenotes.pdf for more information regarding Magic xpa 4.8 features.

V4.7 features

Log file naming

When setting an environment variable named MG_LOG_NO_PID to Y, the log filename doesn't include the process ID at the end. Special care needs to be taken so that two or more engines will not write into the same log file, as there is no sharing mechanism in place.

See http://www2.magicsoftware.com/ver/docs/downloads/magicxpa/4.7releasenotes.pdf for more information regarding other Magic xpa 4.7 features.

V4.6 features

RC5 encryption

RC5 encryption algorithm is re-introduced into the cipher and decipher functions.

See http://www2.magicsoftware.com/ver/docs/downloads/magicxpa/4.6releasenotes.pdf for more information regarding other Magic xpa 4.6 features.

V4.11.1 fixes

MXPA-34812	Application hanged when the logical name and its translation value were same.
MXPA-34736	PID was not removed from the suffix of the log file name, even though environment variable MG_LOG_NO_PID was set to Y
MXPA-34508	Requester tag was not configured correctly in "InMemoryMiddleware/test/execution.properties" file during installation
MXPA-34496	The web_utils/magic.conf aliases for Web_Client_Cache were set up incorrectly.

V4.11 fixes

MXPA-32424	When the Broker was started and stopped without using password, then it was found that each time the Broker was stopped, the core dump files were getting generated in the background.

V4.10 fixes

MXPA-37193	'Bitmap images are not supported for PDF printing' error message is displayed in mgerror.log file, even though the image control on PDF form works fine in runtime.

V4.9 fixes

MXPA-28214	On using Inner Join and updating the main table caused Magic xpa on Linux to crash with the following error: '[Error] - EndTaskRequest(): J999J Test Inner Join;Inner JoinERR-THREAD-ABORTED (-139)…'.
MXPA-29330	In a specific scenario, Magic xpa Runtime engine crashed and suspended its execution in Linux environment.


magic
OUTPERFORM THE FUTURE™

About Magic Software Enterprises

Magic Software Enterprises (NASDAQ: MGIC) empowers customers and partners around the globe with smarter technology that provides a multi-channel user experience of enterprise logic and data.

We draw on 30 years of experience, millions of installations worldwide, and strategic alliances with global IT leaders, including IBM, Microsoft, Oracle, Salesforce.com, and SAP, to enable our customers to seamlessly adopt new technologies and maximize business opportunities.

For more information, visit http://www.magicsoftware.com.

Disclaimer