

Magic xpa 4.6

For Linux

Release Notes

For 64-bit Platforms Only



OUTPERFORM THE FUTURE™

Contents

FOR 64-BIT PLATFORMS ONLY	1
CONTENTS.....	1
Support for 64-bit Server Engine.....	3
Magic xpa Licensing	3
Compatibility	3
Installing Magic xpa	3
Pre-Installation.....	3
Installation Steps.....	4
Post Installation	5
Deploying a Web Service	5
Installation Components.....	6
Starting a Magic Project Using GigaSpaces.....	6
Starting a Magic Project Using Broker.....	7
Magic xpa Requesters.....	8
Additional Settings	9
Apache Requester Installation and Configuration.....	9
Apache Module Requester Setup	9
Using Apache Web Server with a Non-Default Port	10
File Names.....	10
Colors	10
External Code Pages.....	10
Platform-specific Information	10
Linux.....	10
Gateway-specific Information.....	11
ODBC Gateway	11
General Information	11
Installation and Setup Instructions	11
Setting the Magic Configuration File (MAGIC.INI)	12

Limitations and Recommendations.....	12
JMS.....	12
Backups.....	13
Compression	13
Java Integration	13
WebSphere MQ	13
LDAP	13
External Procedures.....	14
Fully Qualified Domain Name (FQDN).....	14
Deploying a Rich Client Application	15
V4.6 features	16
ABOUT MAGIC SOFTWARE ENTERPRISES	17
Disclaimer.....	17

We are delighted to announce the release of Magic xpa 4.6 Application Platform for Linux with new features and enhancements.

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.

Compatibility

For more information about the various platforms on which Magic xpa has been checked for operation by Magic Software Enterprises, refer to the **Compatibility Guide.pdf** file provided with this installation.

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.
 - a. If you choose to use **GigaSpaces** as the messaging infrastructure, the installation procedure updates the relevant configuration files accordingly in **GigaSpaces** and **GigaSpace-xpa** 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

In order to provide web services using Magic xpa, follow the below instructions:

1. Install **web_services/apache-ant-1.9.11**.
2. Make sure the following environment variables point to the right installation:
 - **MAGIC_XPA_HOME** should point to the Magic xpa installation directory
 - **TOMCAT_HOME** should point to the apache-tomcat-7.0.62 installation directory
 - **ANT_HOME** should point to the apache-ant-1.9.11 installation directory
 - **AXIS2_HOME** should point to the axis2-1.7.6 installation directory

Deploying a Web Service

The below file is provided as an example for deploying a web service:

GigaSpaces-xpa/samples/Web Service using Apache Axis2 under Tomcat/1-Provider/Build &Deploy a POJO service to Axis2, under Tomcat, using ant.

The below file is provided as an example for consuming a web service through Apache Axis2:

GigaSpaces-xpa/samples/Web Service using Apache Axis2 under Tomcat/2-Consumer/Native Java/Build_and_Execute_a_client.sh

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 11g, 12c, 18, 19(**bin/mgoracle11, 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 SQLite gateway (**bin/mgsqllite**)
9. Gigaspaces
10. Magic xpa CGI requester (**cgibin/mgrqcgi046**)
11. Magic xpa requesters (64-bit only) for Apache Web Server (**cgibin/mod_V2_mgrequest046.so, cgibin/mod_V2.2_mgrequest046.so**)
12. Magic xpa UDF/UDP examples (**userproc directory**)
13. Magic xpa Web utility files used for Browser Client support (**web_utils directory**)
14. Magic xpa Hangul support (**language/mglocal.kor**)
15. Magic xpa Messaging component (**messaging/messaging.mff**)
16. Web Service provider installations: Apache Ant
17. apache tomcat 9.0.20 (<http://www.apache.org/>).

Starting a Magic Project Using GigaSpaces

1. Start the Tomcat web server via the Tomcat start-up script.
2. Follow the instructions for **Configuring Servers and Projects to Start Automatically** in the **Running Your Magic xpa Projects** section of the concept paper **Deploying Applications on GigaSpaces**.

Note: All scripts have a suffix of **.sh**, whereas the Windows installation uses a **.bat** suffix.

3. Start the space by executing the command:

```
cd $MAGIC_HOME/GigaSpaces-xpa/bin
```

```
./gs-agent.sh
```

3. You can decide to add the line given below in bold to the **Gigaspace-xpa/bin/CommandLine.sh** script in order to allow changing Gigaspaces log file location.

```
$JAVACMD -classpath  
"../../support/*:../../lib/xpa/*:../../GigaSpaces/lib/required/*:$CL  
ASSPATH" -Dcom.gs.multicast.enabled=false -
```

```
Djava.util.logging.config.file=$XAP_LOGS_CONFIG_FILE -  
Dcom.magicsoftware.xpa.home=$MAGIC_XPA_HOME -  
Dlog4j.configurationFile=$MAGIC_XPA_HOME/GigaSpaces-  
xpa/config/log4j2.xml $MAGIC_XPA_LOGS_OPTIONS -  
Dcom.magicsoftware.xpa.requester.conf=$MAGIC_XPA_HOME/Scripts/con  
fig/mgreq.ini -  
Dcom.magicsoftware.xpa.ServerLoadOnDemandTimeout=10  
com.magicsoftware.xpa.requester.commandline.CommandLineUtility  
$ARGS
```

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 (mgrqcgi046):
<http://<server name>/cgi-bin/mgrqcgi046?appname=example1&prgname=prog1>
2. The Magic xpa requester for Apache (mod_V2.2_mgrequest046.so):
<http://<server name>/mgrequest046?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_mgrequest046.so** to be used with Apache 2.4.

Apache Module Requester Setup

Magic xpa 4.6 includes requesters for Apache Web Server version 2.4.

1. Make sure the requester module **mod_V2.4_mgrequest046.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 mgrequest046_module modules/mod_V2.4_mgrequest046.so
<Location /mgrequest046>
    SetHandler mgrequest046-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/mgrequest046?appname=...
```

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

```
InternetDispatcherPath=/mgrequest046
```

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/mgrqcgi046`

instead of `/cgi-bin/mgrqcgi046`.

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 setting must also be set to the same code page: **SPECIAL_INTERNAL_NONUNICODE_CODEPAGE**

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 4.6 Server for Linux should be used with Red Hat Enterprise Linux version 7.3.
- The Oracle gateway should be used with the Oracle 11g 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 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. (Uncommenting 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 4.6 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:

- Edit the **MGREQ.INI** file
- Set **BindFirstIPAddress=Y**.
- Set **MessagingServer** to **FQDN/port**.
- Edit the **MGRB.INI** file.
- Set **MessagingServer** to **FQDN/port**.
- In the **MAGIC.INI** file, set **TCP/IP=2, 30, 1500-2000 /LocalHost=FQDN**.
- 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 given 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="/mgrequest046"/>
      <property key="appname" val="frame"/>
    </properties>
  </xml>
</body>
```

```

    <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">

```

V4.6 features

RC5 encryption

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

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

Magic® is the registered trademark of Magic Software Enterprises Ltd. All other product and company names mentioned herein are for identification purposes only and are the property of, and might be trademarks of, their respective owners.

Magic Software Enterprises has made every effort to ensure that the information contained in this document is accurate. However, there are no representations or warranties regarding this information, including warranties of merchantability or fitness for a particular purpose. In no event Magic Software Enterprises Ltd. will be liable for any loss of profit, business, or data, or any kind of indirect, incidental or consequential damages due to the use of this document. Magic Software Enterprises assumes no responsibility for errors or omissions that may occur in this document. The information in this document is subject to change without prior notice and does not represent a commitment by Magic Software Enterprises or its representatives.

© Magic Software Enterprises, 2013-2020