Magic xpa 3.3i
UNIX
Release Notes

Introducing Magic Software's Magic xpa 3.3i for UNIX Platforms

We are delighted to announce the launch of a new release of Magic Software's Magic xpa Application Platform.

Magic xpa allows organizations to quickly and cost effectively enjoy all the benefits of Rich Internet Applications (RIA) applications, whether on-premise or on-demand.

Based upon a unique, unitary development paradigm, Magic xpa gives the power to quickly develop, enhance, and deploy business applications under multiple deployment models and at a fraction of the cost and time compared to conventional .NET or Java environments.

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

Pre-Installation

- During the installation, several user-environment files are overwritten. Therefore it is best to back up the following files before starting the installation process:
  .cshrc, .profile, .bash_profile (applies to Linux only).
- If you already have a previous Magic xpa server version installed, it is best to install the product using a different user name.

Installation Steps

1. Create a new user. (The installation should be performed 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 **$HOME directory: tar xvf <installation file>**. The installation file name is **magicxpa_<Version>.<platform>.tar**.
5. Run the **./magicxpainstall** command and enter the requested information.
6. When choosing to use GigaSpaces as the messaging infrastructure, the installation procedure will update the relevant configuration files accordingly in the **GigaSpaces** and **GigaSpace-xpa** directories. Note: On UNIX systems, the Tomcat server is used to host the HTTP requester. Follow the Windows installation explanation regarding those products with one difference – the scripts used have the **.sh** and not the **.bat** suffix.
7. When choosing the broker as the messaging infrastructure, run the **$HOME/sbin/mgroot.sh** file as a root user after the installation has been successfully completed. This script copies Magic xpa files that should be accessed by your Web server and follow the steps for the Apache Web server configuration.
8. To set up an Apache Web server, append the **$HOME/web_utils/magic.conf** file to the Apache configuration file (**httpd.conf**), place the requester in the **modules** directory and restart the Apache Web server (see Apache Requester Installation and Configuration).
9. Log out from the new user and log in again to enable the new environment settings.
10. If you need to uninstall the product, delete the user home directory created for the installation. For a complete removal, delete the files copied by the **$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:

install web_services/apache-tomcat-7.0.62

install web_services/apache-ant-1.9.11
install web_services/axis2-1.7.6

copy web_services/axis2.war to apache-tomcat-7.0.62/webapps

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

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.cmd

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 UNIX operating systems, an environment variable points to the executable, which should be used for a specific gateway, located at **$HOME/etc/mgenv**. For example, in UNIX: **MAGIC_ DB_ 14_ DRIVER=$HOME/bin/mgoracle10**
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

- Magic xpa Server (bin/mgxparuntime)
- Magic xpa Broker (broker/mgbroker)
- Magic xpa command line requester (broker/mgrqcmdl)
- Magic xpa gateway 11g and 12c (bin/mgoracle11, bin/mgoracle12) – versions are platform specific
- Magic xpa gateway for DB2 UDB Version 11.1 (bin/mgdb2) – for AIX and Linux platforms only
- Magic xpa gateway for ODBC (bin/mgodbc) – for Linux platform only
- Magic xpa memory gateway (bin/mgmemory)
- Magic xpa SQLite gateway (bin/mgsqlite)
- Gigaspaces
- Magic xpa CGI requester (cgibin/mgrqcgi033)
- Magic xpa requesters (32-bit and 64-bit) for Apache Web Server (cgibin/mod_V2_mgrequest033.so, cgibin/mod_V2.2_mgrequest033.so)
- Magic xpa UDF/UDP examples (userproc directory)
- Magic xpa Web utility files used for Browser Client support (web_utils directory)
- Magic xpa license server (FlexLM 7 in license directory)
- Magic xpa Hangul support (language/mglocal.kor)
- Magic xpa SNMP sub-agent (snmp/mgsnmp.so)
- Magic xpa Messaging component (messaging/messaging.mff)
- Web Service provider installations: Apache Ant , Apache Tomcat , Apache Axis2.

(http://www.apache.org/).

Starting the Magic Project Using GigaSpaces

1. Follow the post installation instructions in the **setenv.sh** file, which is located in the following location: **$MAGIC_HOME/GigaSpaces-xpa/apache-tomcat**.

2. Start the Tomcat web server via the Tomcat startup script.

3. Follow the instructions for **Configuring Servers and Projects to Start Automatically** in the **Running Your Magic xpa Projects** section of the **Deploying Applications on GigaSpaces** concept paper:

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

4. Start the space by executing the command:

   cd $MAGIC_HOME/GigaSpaces-xpa/bin

   ./gs-agent.sh

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

```
$JAVACMD -classpath
"../../support/*:../lib/xpa/*:../../GigaSpaces/lib/required/*:$CLASSPATH"  -
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/config/mgreq.i
ni -Dcom.magicsoftware.xpa.ServerLoadOnDemandTimeout=10
com.magicsoftware.xpa.requester.commandline.CommandLineUtility  $ARGS
```

Starting the Magic Project Using the Broker

To [1]run the Magic xpa Server:

1. Start the license server by running the **$HOME/license/mglmstart** script.
2. 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 UNIX: MGREQENV = $HOME/etc/MGREQ.INI).

1

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

1. The Magic xpa CGI requester (**mgrqcgi033**):

   http://<server_name>/cgi-bin/mgrqcgi033?appname=example1&prgname=prog1

2. The Magic xpa requester for Apache (**mod_V2.2_mgrequest033.so**):

   http://<server_name>/mgrequest033?appname=example1&prgname=prog1

Additional Settings

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

- RetryMainTime
- KeepAlive

Apache Requester Installation and Configuration

Apache Module Requester Setup

Magic xpa 3.3 includes requesters for the Apache Web Server version 2.2 and version 2.4. The requester module **mod_V2.2_mgrequest033.so** should be placed in the **modules** directory of the Apache installation (default: **/usr/local/httpd/modules**) with execute permissions.

The installation includes the following requesters:

mod_V2.2_mgrequest033.so To be used with Apache 2.2

mod_V2.4_mgrequest033.so To be used with Apache 2.4

1. Add the following lines to the Apache configuration file, **httpd.conf**.

```
LoadModule mgrequest032_module
  modules/mod_V2.2_mgrequest033.so

<Location /mgrequest033>
   SetHandler mgrequest033-handler
</Location>

SetEnv MGREQ_INI_PATH  <directory>
```

2. **For AIX:** Add the **$MAGIC_HOME/lib** to the **$LIBPATH** environment variable.
   For Linux: Add the **$MAGIC_HOME/lib** to the **$LD_LIBRARY_PATH** environment variable.
3. Restart the Apache Web server.

The 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**.

Example

SetEnv MGREQ_INI_PATH /usr/local/httpd/conf

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

To use this requester, call Magic xpa using a URL, such as:
http://server/mgrequest033?appname=...

You should also modify the MAGIC.INI file as follows:
InternetDispatcherPath=/mgrequest033

Using an Apache Web Server with a Non-Default Port

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

InternetDispatcherPath= http://server:port/cgi-bin/mgrqcgi033

instead of

/cgi-bin/mgrqcgi033

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 UNIX platforms, files can be referred to by a full URL only. Any reference to a file name with a slash (/) is considered to be a path name, either full or relative.

Examples:

/etc/home1/a.jpg (full path)

http://myserver/myalias/a.jpg (full URL)

myalias/a.jpg – is considered to be a relative path name, not a relative URL.

Colors

To use colors properly on UNIX 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

When installing Magic xpa on UNIX 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 codepage : SPECIAL_INTERNAL_NONUNICODE_CODEPAGE

Platform-specific Information

Linux

For Intel processors only, Linux requires Kernel 2.6.32-696 and up with glibc-2.12-1.209 and up.

The Magic xpa 3.3i Server for Linux should be used with Red Hat Enterprise Linux version 6.9 , 7.1 7.2 or 7.3, or with SLES 11 SP3.

The Oracle gateway should be used with the Oracle 11g client and above.

The Magic xpa DB2 gateway for AIX should be used with the DB2 Version 11.1 client.

The Websphere MQ 5.3 client/server is required for working with the MQ messaging capabilities.

JRE 1.7 or 1.8 is required for working with Java integration capabilities.

Apache 2.0.43 (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.

When using the Oracle gateway, make sure that **ORACLE_HOME** and **ORACLE_SID** are set in the **$MAGIC_HOME/etc/mgenv** file, and that the environment variable **LD_LIBRARY_PATH** (or **LIBPATH** for AIX) includes the **$ORACLE_HOME/lib** directory.

When using the DB2 gateway, make sure that **DB2INSTANCE** is set in the **$MAGIC_HOME/etc/mgenv** file.

ODBC Gateway on the Linux Platform

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 and 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. Uncomment means to remove the semicolon preceding the entry.
2. Install the **UnixODBC** ODBC manager, this product can be downloaded from: http://www.unixodbc.org. Follow the online instructions to generate the ODBC manager.

   Locate the following two files (shared libraries): **libodbc.so.1.0.0** and **libodbcinst.so.1.0.0**

   Copy the files to the directory **$MAGIC_HOME/lib**.
3. In the same directory create symbolic links for the two libraries:

   ln –s libodbc.so.1.0.0 libodbc.so.1

   ln –s libodbcinst.so.1.0.0 libodbcinst.so.1
4. Install the ODBC driver. Refer to the specific driver documentation for installation instructions.
5. Make sure that the libraries have Execute permission. Use the **chmod +x** command to set execute permission.
6. 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.

To help you set up quickly, we have included the following **.odbc.ini** file as an example:

```
[mysql]
Driver   = /usr/magicadm/mysql/libmyodbc-2.50.23.so
Trace    = No
Tracefile= mysql.log
Database = samp_db
```

Each section defines a DSN (Data Source Name). In the above example, there is one defined DSN named **mysql**. The driver entry in each section should be set 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)

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 can use JMS with the Sun Reference application, the environment variables listed below are needed to run J2EE applications on UNIX 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 recommended 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 UNIX platform.

Java Integration

The Java CLASSPATH separator character on UNIX 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).

AIX

The **JAVA_HOME** entry should be set in the MAGIC_JAVA section of the MAGIC.INI file.

For example: If **JAVA_HOME** is set to **/usr/java6**, Magic xpa adds **/jre/bin/classic/libjvm.a** as a prefix in order to find the **libjvm.a** library.

If you encounter problems locating this file you can use the environment variable: **MG_JAVALIB**, which should be set to the absolute path of the library file.

For example: **MG_JAVALIB = /usr/java6/jre/bin/classic/libjvm.a**

The **AIX LIBPATH** variable should include **/usr/java6/jre/bin:/usr/java6/jre/bin/classic**

Linux

If Java is installed on your server, you should edit the following scripts: **.cshrc** and **.profile**.

The **LD_LIBRARY_PATH** environment variable should include **$JAVA_HOME/jre/lib/i386/client** and **$JAVA_HOME/jre/lib/i386**

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 3.3e has been switched to OpenLdap .
When using SSL with LDAP the search path for the location of the certificates has changed.
Either 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 4.1.2	g++	gcc
AIX	IBM XL C/C++ for AIX, V12.1	xlC_r	cc_r

FQDN (Fully Qualified Domain Name)

The 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.

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

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

Y – During binding to a port, the server will resolve the host name and will bind to the resolved IP address.

N – The server will bind 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 on a machine):

1. Edit the MGREQ.INI file and enable **BindFirstIPAddress** (= Y) and set **MessagingServer** to **FQDN/port**.
2. Edit the MGRB.INI file and set **MessagingServer** to **FQDN/port**.
3. In the MAGIC.INI file, set **TCP/IP = 2, 30, 1500-2000 /LocalHost=FQDN**.
4. In the MAGIC.INI file, set the **Default Broker** to **FQDN/port**.

The table below shows the binding for the server module:

Port Number	
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 UNIX platforms:

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\`

   a. Place them in the **MagicRIAApplications/*appname*** alias on the Web server.

   b. Users can access the application from the following URL: http://*appserver*/MagicRIAApplications/*appname*/*appname*.publish.html

2. 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
```

3. 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="aix51:2261"/>
            <property key="requester" val="/mgrequest033"/>
            <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">
```

V3.3e Fixed Issues

MXPA-691 - Call remote from XPA client towards XPA engine under GS did not send Magic user to the server.

V3.3c features

Vulnerability scanning

In order to protect the broker from processing requests bombarded on it by vulnerability scanners, a new keyword named *StrictSignatureSize* is now introduced. This keyword to be added to MGREQ.INI, adds a binary value to each of the messages exchanged between the partitioning modules. In case

a received signature will differ from the expected signature, the message will be discarded.

Note: this entry needs to be added to MGREQ.INI files in both the engine and broker directories.

V3.3 features

Encryption of cached files in a RIA architecture

Cached files are encrypted by the server using the Data Encryption (DES) algorithm by default. You can choose instead to use the Advanced Encryption (AES) algorithm, which is more secure. When the below flag is set to Y, cached files are encrypted using AES.

[MAGIC_SPECIALS]SpecialRIAEncryptAES

V3.3 Fixed Issues

- 148074 - When setting the RequesterTimeoutSec to 0 in mgreq.ini and trying to execute a Rich Client program ,an error "Unable to find an available server: Maximum contexts reached.(-120) " was displayed.
- 148355 - The XmlValidate() function didn't work for xsd files with upper case characters

V3.2d Fixed Issues

- 147921 - ClientFileToServer did not work properly on Linux with Gigaspaces and Tomcat as the web server.
- 146039 - A requester crash was encountered when using the Apache 2.2 64 bit edition.
- 147732 - Gigaspaces did not start on Linux when setting the DISCOVERY_PORT in GigaSpaces-xpa/bin/setenv.sh

V3.2b Fixed Issues

- 144270 (Linux only) – Conversion of a null string from Unicode to multi-byte occurred while handling an XML file.

- If environment variable MG_REMOVE_ARGS exists then all arguments sent to the runtime engine will not show in the process list (ps-ef) – (Linux only)
- 143590 – A crash occurred while using the menu() function in a Rich client application.

V3.2a Fixed Issues

- 144270 - Conversion of a null string from Unicode to multi-byte occurred while handling an XML file. (AIX only)
- 143381 – A memory corruption would cause a "Failed to open, data source:" message to appear.

V3.2 Fixed Issues

- 138668 – ps command showed the engine with all of its arguments (Linux only)