



Lifting Off into Space-based Architecture with Magic xpa 3.x

Self-Paced Tutorial

Book ID: UTLSBAMA3x

Edition: 1.0, December 2016

Course ID: UCLOSBMA32

Magic University Official Courseware

The information in this manual/document is subject to change without prior notice and does not represent a commitment on the part of Magic Software Enterprises Ltd.

Magic Software Enterprises Ltd. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms and conditions of the license agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

No part of this manual and/or databases may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or information recording and retrieval systems, for any purpose other than the purchaser's personal use, without the prior express written permission of Magic Software Enterprises Ltd.

All references made to third-party trademarks are for informational purposes only regarding compatibility with the products of Magic Software Enterprises Ltd.

Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are part of a completely fictitious scenario or scenarios and are designed solely to document the use of Magic xpa.

Magic® is a registered trademark of Magic Software Enterprises Ltd.

Btrieve® and Pervasive.SQL® are registered trademarks of Pervasive Software, Inc.

IBM®, Topview™, System i5/iSeries™, System i™, IBM i™, pSeries®, xSeries®, RISC System/6000®, DB2®, and WebSphere® are trademarks or registered trademarks of IBM Corporation.

Microsoft®, FrontPage®, Windows™, WindowsNT™, ActiveX™, and Windows 10 Mobile are trademarks or registered trademarks of Microsoft Corporation.

Oracle® and OC4J® are registered trademarks of the Oracle Corporation and/or its affiliates.

Linux® is a registered trademark of Linus Torvalds.

UNIX® is a registered trademark of UNIX System Laboratories.

GLOBEtrrotter® and FLEXlm® are registered trademarks of Macrovision Corporation.

Solaris™ and Sun ONE™ are trademarks of Sun Microsystems, Inc.

HP-UX® is a registered trademark of the Hewlett-Packard Company.

Red Hat® is a registered trademark of Red Hat, Inc.

WebLogic® is a registered trademark of BEA Systems.

Interstage® is a registered trademark of the Fujitsu Software Corporation.

JBoss™ is a trademark of JBoss Inc.

GigaSpaces, GigaSpaces eXtreme Application Platform (XAP), GigaSpaces eXtreme Application Platform Enterprise Data Grid (XAP EDG), GigaSpaces Enterprise Application Grid, GigaSpaces Platform, and GigaSpaces, are trademarks or registered trademarks of GigaSpaces Technologies.

Systinet™ is a trademark of Systinet Corporation.

Android is a trademark of Google Inc.

iPod, iPad, iPhone, iTunes, and Mac are registered trademarks of Apple Inc.

Portions Copyright © 2002 James W. Newkirk, Michael C. Two, Alexei A. Vorontsov or Copyright © 2000-2002 Philip A. Craig

Clip art images copyright by Presentation Task Force®, a registered trademark of New Vision Technologies Inc.

This product uses the FreeImage open source image library. See <http://freeimage.sourceforge.net> for details.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by Computing Services at Carnegie Mellon University (<http://www.cmu.edu/computing/>). Copyright © 1989, 1991, 1992, 2001 Carnegie Mellon University. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes software that is Copyright © 1998, 1999, 2000 of the Thai Open Source Software Center Ltd. and Clark Cooper.

This product includes software that is Copyright © 2001-2002 of Networks Associates Technology, Inc All rights reserved.

This product includes software that is Copyright © 2001-2002 of Cambridge Broadband Ltd. All rights reserved.

This product includes software that is Copyright © 1999-2001 of The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved.

All other product names are trademarks or registered trademarks of their respective holders.

Lifting Off into Space-based Architecture with Magic xpa 3.x

Copyright © 2016 by Magic Software Enterprises Ltd. All rights reserved.

Table of Contents

Introduction.....	5
About Magic xpa 3.x's Space-based Architecture.....	5
About the Seminar	6
How to Use This Guide.....	7
Seminar Materials	7
Benefits of Space-based Architecture	8
Key Terms.....	9
Broker-based Architecture.....	10
Space-based Architecture at a Glance	10
Push vs. Pull Mechanism.....	13
Recovery.....	13
IMDG Database Gateway.....	13
Summary	14
Preparing Your Machine	15
Installing Magic xpa	16
Firewall Settings	17
Starting the Magic xpa GSA Service	18
Running the GigaSpaces Infrastructure	20
Summary	22
Starting and Stopping Your Magic xpa Projects	23
Space Startup Sequence	24
ProjectsStartup.xml file.....	25
Starting and Stopping the Servers and Projects from the Command Line	27
Loading Engines Dynamically.....	28
Licensing	31
Reserving License Threads or Users for a Project.....	32
Monitoring.....	33
Starting the Monitor	34
Verifying that the Applications Started Properly.....	34
Testing that an Application Can Connect to the Space.....	35

Space Deployment	37
Grid Components – Memory Allocation	39
Configuring Event Containers' Concurrency	39
Accessing Magic xpa Servers over the Space from an EJB.....	41
Space Clustering	43
Recommendations for Working with a Cluster.....	46
Server Installation	46
Installation Prerequisites	46
Installation Settings.....	46
Post Installation Configuration Steps.....	46
Deploying on Multiple Servers.....	48
Deploying Web Requesters on Different Servers	48
Troubleshooting.....	51
Logging	52
Space Is Not Deploying	52
Memory-related Issues	54
Waiting for GS Agent Status.....	54
About Magic Software Enterprises.....	57

Introduction

Welcome to Magic Software University's **Lifting Off into Space-based Architecture with Magic xpa 3.x** self-paced tutorial. We, at Magic Software University, hope that you will find this tutorial informative and that it will assist you in getting started with this exciting product.

About Magic xpa 3.x's Space-based Architecture

Magic xpa 3.x uses an In-Memory Data Grid (IMDG) as its underlying messaging layer.

You can easily prepare to handle new demands created by enterprise mobility and cloud services and take advantage of the immediate benefits and capabilities of

Magic xpa 3.x, including:

- Built-in clustering and fail-over capabilities
- Unlimited linear and elastic scalability
- High availability
- Automated recovery
- Improved management and monitoring capabilities

Magic xpa also provides an IMDG gateway that can be used for storing and accessing data in the space. This gateway is similar to the other Magic xpa database gateways and enables complete access to information stored in the space. For more information about the gateway, please see the *Magic xpa Help*.

About the Seminar

Magic University's **Lifting Off into Space-based Architecture with Magic xpa 3.x** seminar is intended for people with experience in working with Magic xpa 3.x and who want to take full advantage of the Space-based architecture.

In this seminar, you will learn about:

- The benefits of Space-based architecture
- The architecture of Magic xpa and its new Space-based infrastructure
- Magic xpa 3.x's startup mechanism and licensing
- The monitoring capabilities
- What to consider when deploying Magic xpa 3.x
- Troubleshooting Magic xpa 3.x when using the Space-based architecture

Seminar Prerequisites

Before you start with the seminar there is basic knowledge that you need to have:

Development knowledge	Familiar with Magic xpa 3.x
-----------------------	-----------------------------

Your computer must also meet some basic requirements:

Hardware	<ul style="list-style-type: none"> • Windows 7 and above. The seminar was tested on Windows 7. • Pentium processor 1.8GHz and upwards (recommended: Dual-core, 2.66-GHz or faster). • 4Gb RAM or greater. • At least 2Gb of free space. • Screen resolution of at least 1024x768 pixels.
Java	The Space middleware requires Java 1.7 to be installed on the server.
.NET Framework	.NET Framework 4.0.
License	The seminar uses the standard license. Please obtain a Magic xpa 3.x evaluation license from your local Magic Software Enterprises representative.

How to Use This Guide

To get the most out of this guide, follow the classroom lesson and ask any questions that you have. You can then review the self-paced guide for that lesson, and if you have further questions you can ask the instructor before the next lesson.

The self-paced guide provides detailed step-by-step instructions. If you are learning using this self-paced tutorial, feel free to contact your Magic Software Enterprises representative or the Support department for further assistance.

Seminar Materials

For the purpose of this seminar, we'll use the test application that is included with the Magic xpa installation. The application is located at: **GigaSpaces-xpa\test\GSTest.ecf**.

Lesson

1

Benefits of Space-based Architecture

Magic xpa 3.x's In-Memory Data Grid (IMDG) middleware ensures that new demands can easily be handled by creating enterprise mobility and cloud services, allowing the end users to take advantage of the immediate benefits and capabilities of Magic xpa 3.x including:

- Built-in clustering and fail-over capabilities
- Unlimited linear and elastic scalability
- High availability
- Automated recovery
- Improved management and monitoring capabilities

Key Terms

Throughout this seminar you will come across some new terms related to GigaSpaces, as seen below:

- **Data Grid** – The in-memory data grid is middleware software composed of multiple server processes running on multiple machine instances (physical or virtual). These processes work together to store large amounts of data in memory, thereby achieving high performance, elastic scalability, and fail-safe redundancy.
- **Space** – A Space is a data and business logic container that stores data objects in memory (similar to a database instance) and runs in the data grid. A data grid can contain multiple Spaces. However, Magic xpa uses a single Space for running multiple projects. For redundancy and scalability, data and business logic in the Space are replicated and partitioned across all machines participating in the data grid, ensuring continuous service even during machine or software failure.
- **Processing Unit** – The Magic processing unit is a software module that runs in the data grid and has direct access to all data in the Space. Magic xpa includes a processing unit component that handles administrative and housekeeping operations such as startup of engines, monitoring, and recovery tasks.
- **Grid Service Agent (GSA)** – This is a GigaSpaces entity that acts as a process manager that can spawn and manage Service Grid processes such as the Grid Service Manager (GSM), the Grid Service Container (GSC), and the Lookup Service (LUS). This agent can be installed as an operating system process. It runs on a specific machine and is responsible for adding that machine to the grid, monitoring grid processes on its own machine, and restarting them in case of process failure.
- **Grid Service Container (GSC)** – A container that hosts processing units or Space partitions.
- **Grid Service Manager (GSM)** – The service that manages the Grid Service Containers. The GSM deploys and undeploys processing units to the grid. The GSM monitors the processing units running within each GSC.
- **Lookup Service (LUS)** – A registry that contains information as to where each loaded component is actually situated.

Broker-based Architecture

In general, the broker middleware is used for development and the deployment of small and medium-sized applications that have a small number of engines.

Space-based Architecture at a Glance

GigaSpaces' XAP in-memory computing technology is the middleware that implements Magic xpa's functionality on the In-Memory Data Grid. The GigaSpaces middleware is the new underlying messaging layer of Magic xpa.

The Space middleware is used for the deployment of medium and large applications that have a large number of engines. Servers within a cluster automatically distribute the workload, which ensures that there is no single point-of-failure. Each partition can have a backup, which improves data consistency and provides high availability of the data. The Space-based architecture also enhances your applications' scalability and load balancing.

Configurations

With Magic xpa 3.x, using the Space middleware, you can set up any of the following configurations:



- Single host + Single engine
- Single host + Multiple engines
- Multiple hosts + Multiple engines

All of these configurations are available thanks to the In-Memory Data Grid.

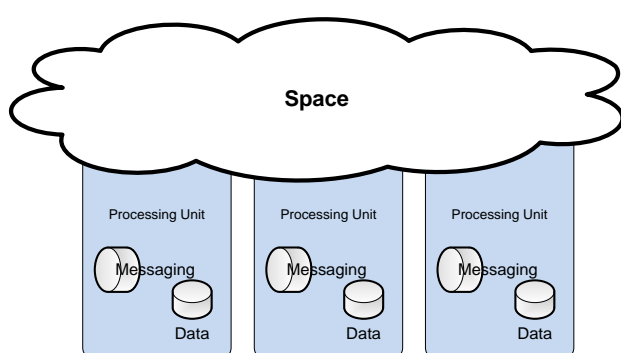
In-Memory Data Grid

One of the biggest benefits of the new Space-based architecture is that the In-Memory Data Grid (IMDG) does away with a single point-of-failure by spreading out the logic across multiple servers running on multiple machine instances (physical or virtual).

Each partition handles all of the logic in a **processing unit**.

Each processing unit holds a subset of the data. Together, the partitions make up the **Magic Space**, which for all intents and purposes, works like a database.

The processing units are independent of one another, so that the application can scale upwards by adding more processing units.

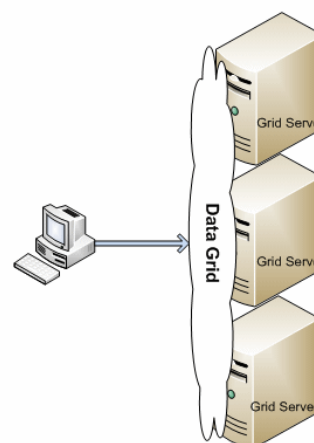


The capabilities of the Space-based architecture provide built-in active/active clustering and fail-over capabilities for Magic xpa projects. It also enables unlimited linear scalability and improves performance.

As you can see in the image on the right, you can add servers to the grid to form a logical grid. You can add servers without disrupting the other servers, thereby extending the data grid.

The IMDG is comprised of the following elements, which are described at the beginning of this lesson:

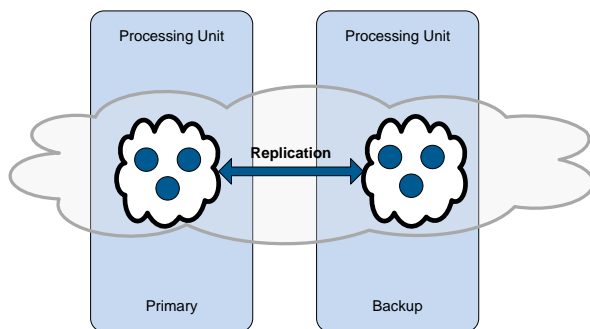
- Grid Service Agent (GSA)
- Grid Service Container (GSC)
- Grid Service Manager (GSM)
- Lookup Service (LUS)



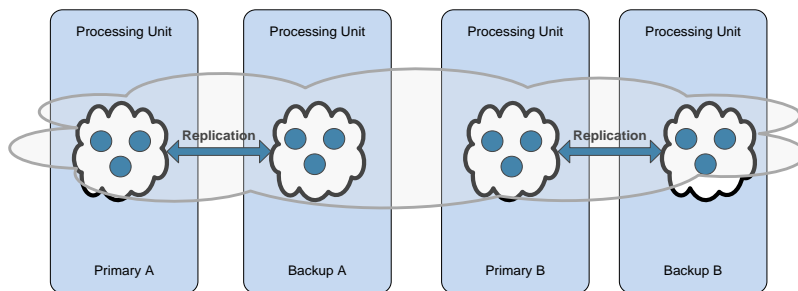
Replication

In the Magic xpa IMDG architecture, each partition can have a backup, which ensures high availability of the information. This improves data consistency and provides high availability of the data.

In replication, you have a partition that is active, known as the primary partition, and another partition that is the backup, often residing on a different server or on a different process on the same server. The so-called primary partition replicates the data with the backup partition.



If the primary partition fails, the backup partition immediately becomes active as the primary partition and already contains the entire information set. The system will then try to automatically load another processing unit that will behave as the new backup unit. This mechanism ensures that the system is self-healing and has zero downtime.



Clustering

In Space-based architecture, the Magic Space can reside on several computers (physical or virtual) that are viewed as a single logical unit, also known as a cluster. A cluster can be seen as a single "large" space and enhances your applications' scalability, high availability and

load balancing. You can have a combination of both clustering and partitioning to ensure that the Magic Space and the data are always available.

Push vs. Pull Mechanism

With the broker, requests are sent to the broker and it is the broker's duty to find a server (engine) that could handle the request. This is known as a *push mechanism*. If there are no available threads to handle the request, the broker puts them in a queue until there is an available thread or a timeout is reached. If there is a problem with the broker, no requests are handled.

The Space middleware uses a *pull mechanism*. The server has one or more **workers** (a pool of workers) that are simultaneously actively searching the Magic Space for requests to handle.

Recovery

The Space-based architecture provides Magic xpa with an enhanced recovery mechanism, enabling you to develop projects that recover automatically from many disaster and failure scenarios. The Grid Service Agent (GSA) is able to automatically reload components that have crashed. When a primary partition crashes in the Magic Space, the backup immediately takes over.

IMDG Database Gateway

Magic xpa 3.x provides a new database gateway, the IMDG gateway, for storing and accessing data in the space. This gateway is similar to the other Magic xpa database gateways and enables complete access to information stored in the space.

Summary

In this lesson, you learned about the benefits of Space-based architecture. In this architecture, you have a processing unit that is made up of two layers: data and messaging. One or more processing units form what is known as the Magic Space. An incoming request will only see the Magic Space, and not each individual processing unit. As a result, it is easy to add processing units to keep up with demand.

The Magic Space itself is an entity on the In-Memory Data Grid. Data is kept in memory, thereby providing faster access to the data. This makes each process faster.

To develop a fail-safe system, you can use a primary-backup implementation and clustering, which ensure zero downtime of your system.

Lesson 2

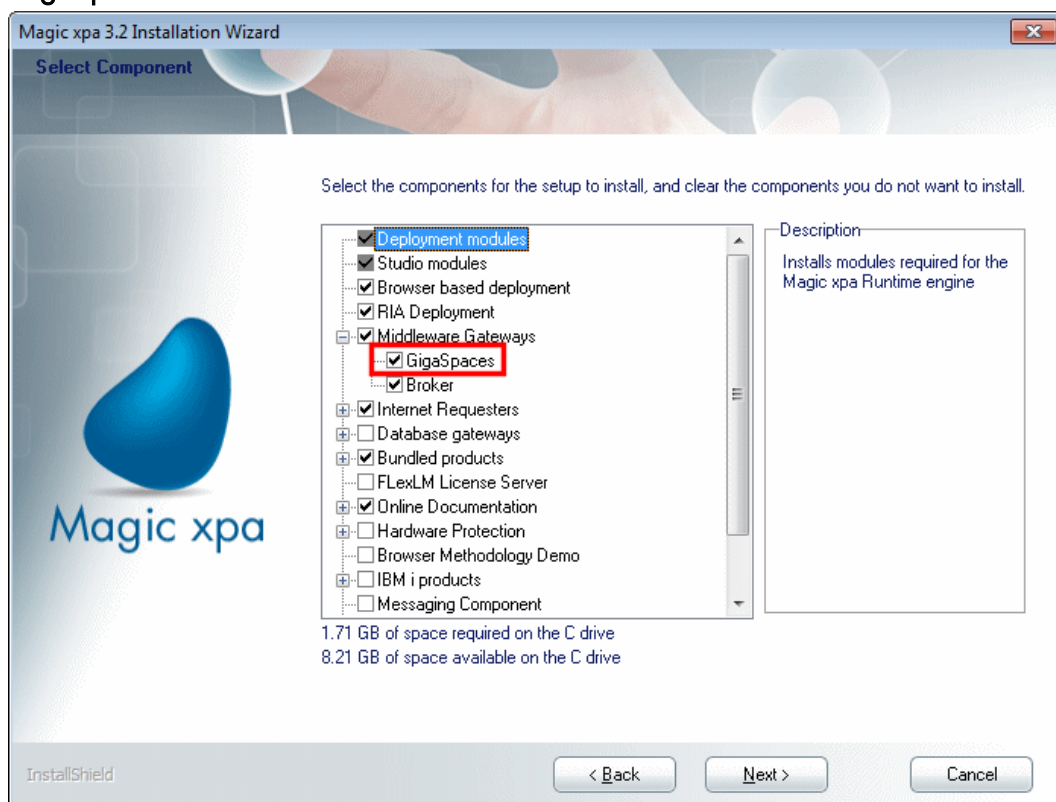
Preparing Your Machine

There are a number of steps you need to take to set up your computer to work with Magic xpa and GigaSpaces:

- Installing Magic xpa
- Setting the fixed ports for the GigaSpaces infrastructure
- Starting the Magic xpa GSA service, which starts the GigaSpaces Grid and deploys the Magic Space

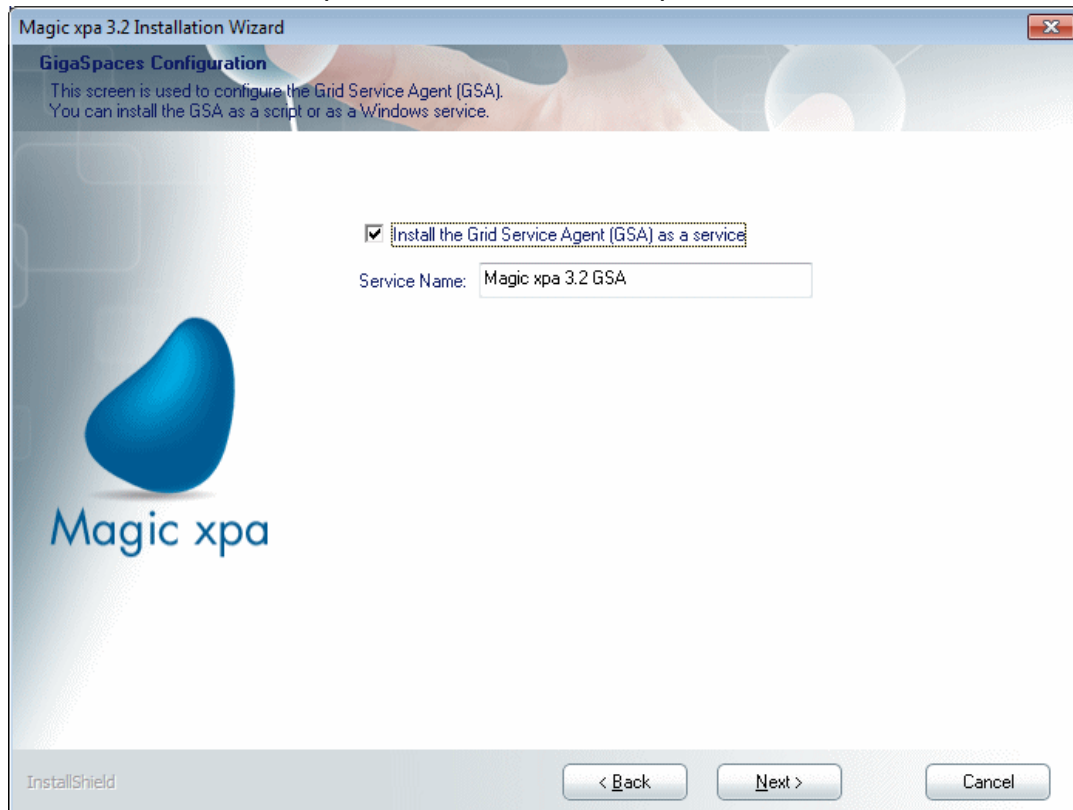
Installing Magic xpa

1. Ensure that your computer meets the hardware requirements listed in the **Seminar Prerequisites** section on page 6.
2. Install the Magic xpa 3.2 Application Platform.
3. Run through the installation screens. In the **Select Product** screen, select **Magic xpa Studio**.
4. In the **Setup Type** screen, select **Custom**.
5. In the **Select Components** screen, select the database of your choice and uncheck the **Bundled products** option.
6. Notice by default that the both middleware gateways are selected, **Broker** and **GigaSpaces**. Leave this as-is.



7. Since you selected to install **GigaSpaces** in the **Product Components** screen, the **GigaSpaces Configuration** screen will appear. On this screen, you select whether to install GigaSpaces to be started manually or as a service.

8. Select the **Install the Grid Service Agent (GSA) as a service** check box if you want to start the GSA service on your machine automatically.



9. Go through the rest of the screens leaving the defaults as-is.

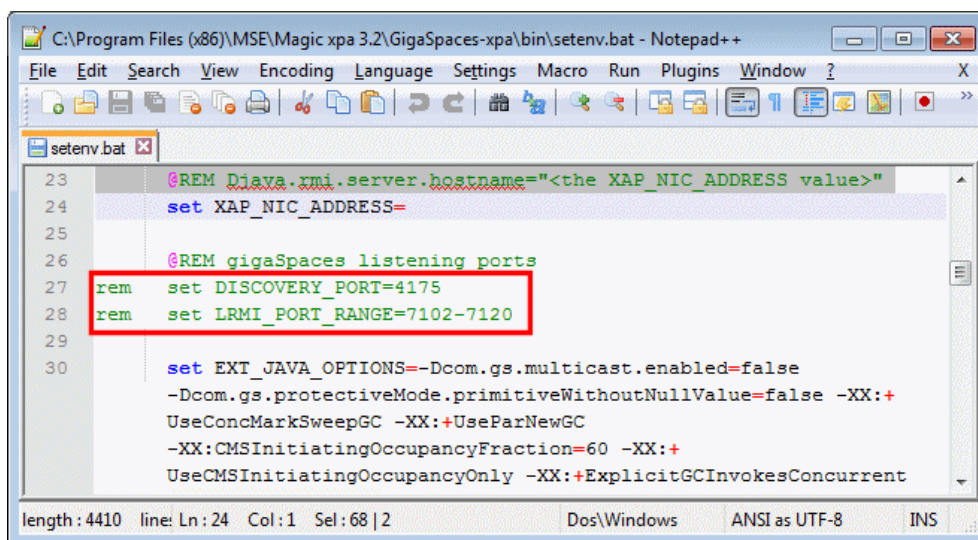
Firewall Settings

There are two settings that you need to enable to set the fixed ports for the GigaSpaces infrastructure:

- The discovery listening port.
- The range of ports for the LRMI (the protocol for internal communication between grid components).

You make these changes by doing the following:

1. From the **GigaSpaces-xpa\bin** folder, open the **setenv.bat** file in a text editor.
2. Find the line with **DISCOVERY_PORT** and remove the text **rem** from the beginning of the line.
3. If the discovery port is set to a port other than the GigaSpaces default port (**4174** in XAP 11.0), you must also change the value of the locators to use the port that you defined.
4. Do the same for the line with **LRMI_PORT_RANGE**.



```

C:\Program Files (x86)\MSE\Magic xpa 3.2\GigaSpaces-xpa\bin\setenv.bat - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ? X
setenv.bat
23 @REM Diava.rmi.server.hostname="<the XAP_NIC_ADDRESS value>"
24 set XAP_NIC_ADDRESS=
25
26 @REM gigaSpaces listening ports
27 rem set DISCOVERY_PORT=4175
28 rem set LRMI_PORT_RANGE=7102-7120
29
30 set EXT_JAVA_OPTIONS=-Dcom.gs.multicast.enabled=false
-Dcom.gs.protectiveMode.primitiveWithoutNullValue=false -XX:+
UseConcMarkSweepGC -XX:+UseParNewGC
-XX:CMSInitiatingOccupancyFraction=60 -XX:+
UseCMSInitiatingOccupancyOnly -XX:+ExplicitGCInvokesConcurrent
length: 4410 line: Ln: 24 Col: 1 Sel: 68 | 2 Dos\Windows ANSI as UTF-8 INS
  
```

Both of these settings can be enabled separately. This means that if you want to keep the default discovery port, and only configure the range of ports for the LRMI, you can enable only the LRMI settings.

Additional information about the ports can be found in the **Deploying Applications on GigaSpaces.pdf** file in the Magic xpa Support folder.

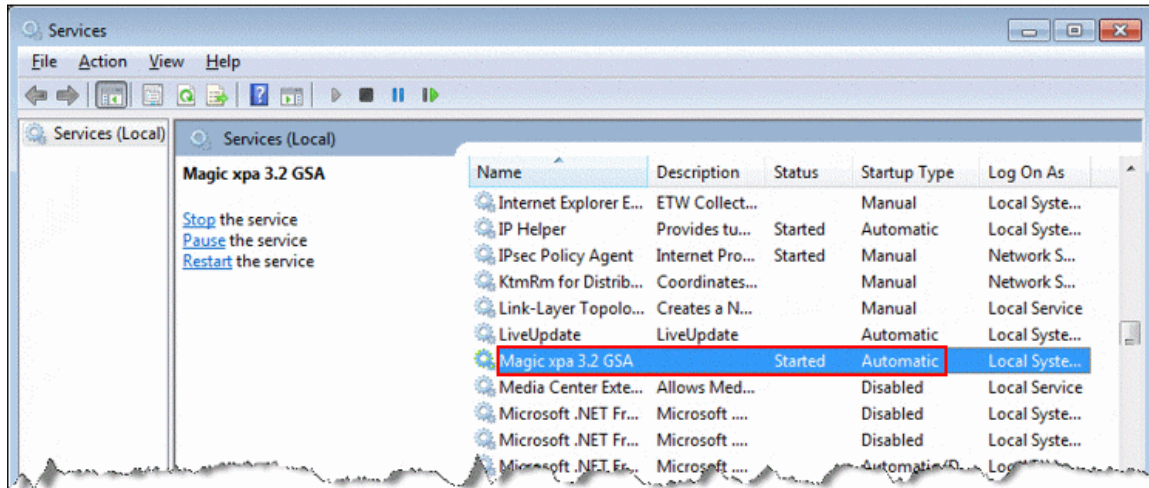
Starting the Magic xpa GSA Service

If you selected the **Install the Grid Service Agent (GSA) as a service** check box when installing Magic xpa, the **Magic xpa GSA** service is installed, but it is not set to start automatically.

For deployment purposes, you should set this to start automatically, which you do as follows:

1. On your machine, from the **Start** menu, click **Run**.
2. In the **Run** dialog box, enter **services.msc**.
3. In the **Services** dialog box, look for **Magic xpa 3.2 GSA**. If it has a **Started** status and an **Automatic** startup type, then it's running.

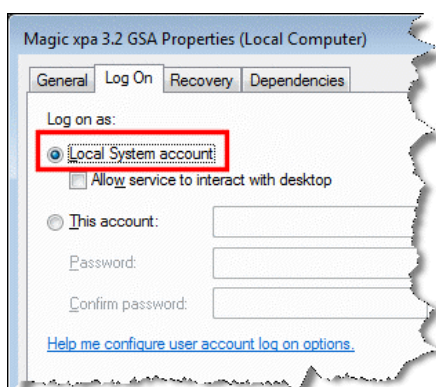
4. If not, double-click on it and in the **Magic xpa 3.2 GSA Properties (Local Computer)** dialog box's **Startup type** parameter, select **Automatic**.
5. Click **OK** to finish.



When the **Magic xpa GSA** service starts on a specific machine, it loads the grid locally and searches the LAN for other grid components with the same Lookup Locator names. If any such components are found, the local grid is considered to be part of that Lookup Locator. In this way, a single unified grid is established in the network.



It's important to remember that the Magic xpa engines run under the user defined for the GSA service and not by the logged in user. By default, the user defined for the service is the **Local System account** as shown in the image below. For running Magic xpa on a single machine, this is usually fine. However, on a clustered environment, the service should run as a user who has privileges to access network resources.



Running the GigaSpaces Infrastructure

Before you start a Magic xpa project, the GigaSpaces infrastructure should be running. The GigaSpaces infrastructure startup is constructed from two parts:

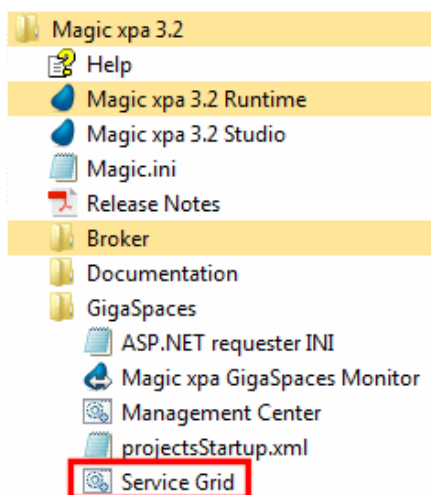
- Starting the GigaSpaces Grid
- Deploying the Magic Space



Running the GigaSpaces infrastructure is a one-time operation. There is no need to restart the operating system service once it has been started.

It is also possible to manually start the Grid and deploy the Space, but it is recommended that you use the dedicated operating system service to do this.

1. To start the grid and deploy the space manually, in the **Start** menu, open the **GigaSpaces** folder and select the **Service Grid**. You can also start the grid and space by running the **gs-agent.bat** file from the **GigaSpaces-xpa\bin** folder.

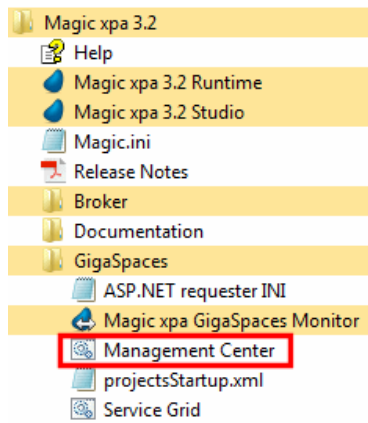


2. You need to start the grid as an Administrator user. If you start the grid manually via the batch file, make sure that you start the **Command Prompt** app with the **Run as Administrator** option set.
3. If you do not want to deploy the Space database, open the **gs-agent.bat** file in a text editor and set the **gsa.deployDatabaseSpace** entry to **0**.

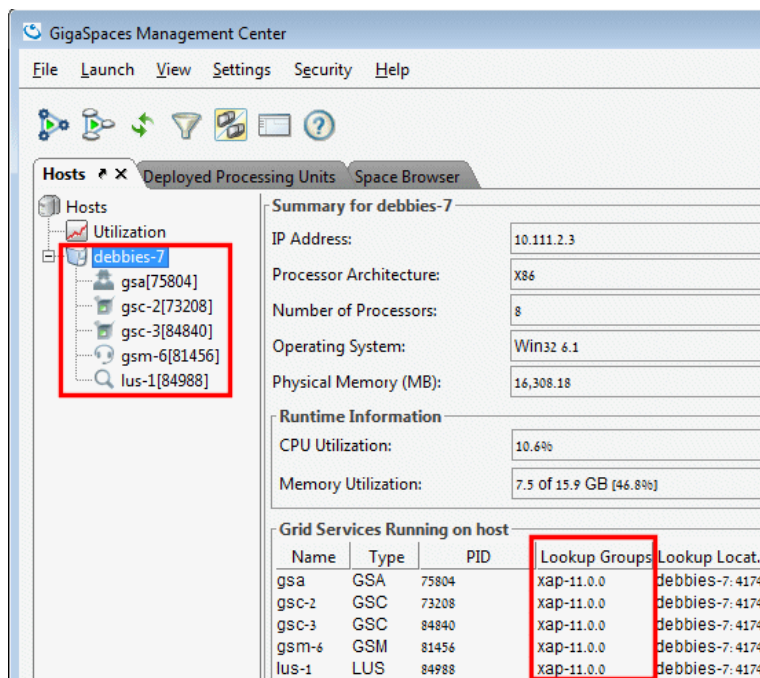
Verifying that the Grid Started Properly

To verify that the Grid is running:

1. From the installation shortcuts in the **Start** menu, click **Management Center**, which is located in the **GigaSpaces** folder.



2. The **GigaSpaces Management Center** dialog box will open.
3. If the Grid has started properly, you should see your machine listed in the **Hosts** tab. Under the machine name, for the default configuration, you should have one **GSA**, two **GSCs**, one **GSM**, and one **LUS**.
4. In addition, all those GigaSpaces entities should be listed with the **LookupGroup** defined during the installation.



Summary for debbies-7

IP Address: 10.111.2.3

Processor Architecture: X86

Number of Processors: 8

Operating System: Win32 6.1

Physical Memory (MB): 16,308.18

Runtime Information

CPU Utilization: 10.6%

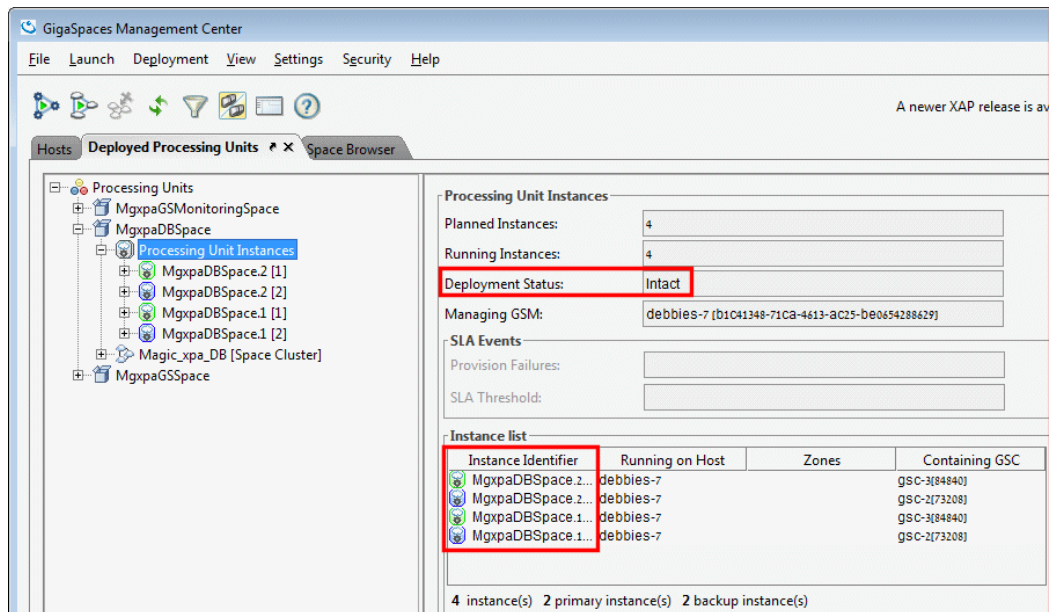
Memory Utilization: 7.5 of 15.9 GB [46.8%]

Grid Services Running on host

Name	Type	PID	Lookup Groups	Lookup Locat...
gsa	GSA	75804	xap-11.0.0	debbies-7: 4174
gsc-2	GSC	73208	xap-11.0.0	debbies-7: 4174
gsc-3	GSC	84840	xap-11.0.0	debbies-7: 4174
gsm-6	GSM	81456	xap-11.0.0	debbies-7: 4174
lus-1	LUS	84988	xap-11.0.0	debbies-7: 4174

To verify that the Magic Space has been deployed:

1. From the **GigaSpaces Management Center**, click the **Deployed Processing Units** tab.



2. You should see four instances running, and the **Deployment Status** should be **Intact**.

Summary

You learned how to set up your machine so that it is ready to work with Magic xpa and GigaSpaces. You installed Magic xpa and set the Magic xpa GSA service to start automatically. You started the GigaSpaces Grid and deployed the Magic Space using the **gs-agent.bat** file.

You learned how to verify that the grid and space were set up properly using the **GigaSpaces Management Center**. You also learned about a few changes to make to the firewall settings.

Lesson 3

Starting and Stopping Your Magic xpa Projects

There are a number of steps to make sure that your Magic xpa works smoothly and effortlessly when using the Space-based architecture.

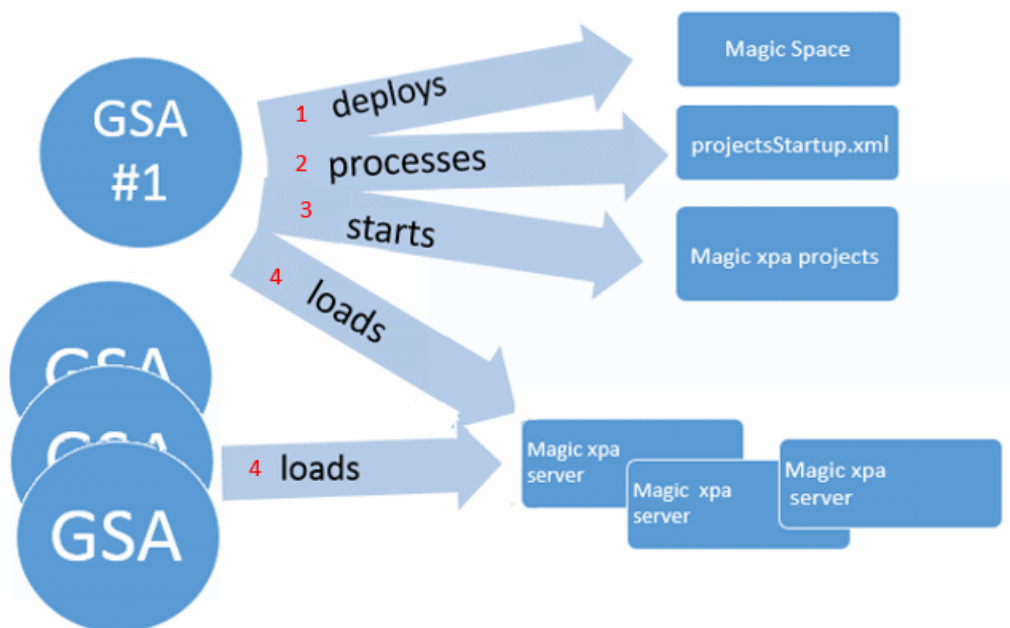
This lesson covers various topics including:

- Startup sequence
- ProjectsStartup.xml file
- Automatic provisioning
- Starting and stopping the servers and projects from the command Line
- Stopping projects with on-demand servers

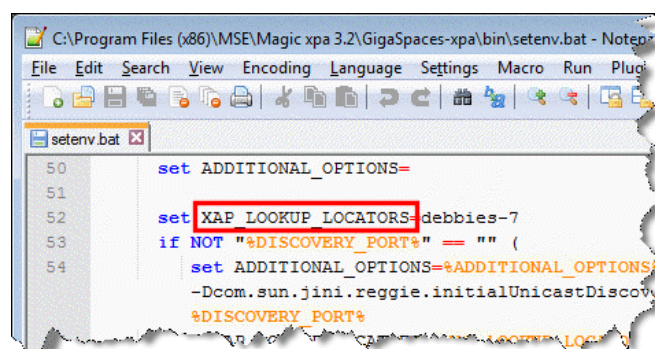
Space Startup Sequence

The startup sequence of the grid is as follows:

1. The GSA is started (usually as a service) on each application server that is part of the grid. The GSA starts the core grid infrastructure to connect all application servers together.
2. The first GSA deploys the Magic Space and starts the projects. This Space contains all Magic-related objects that are shared across the grid.
3. The same GSA also deploys the **projectsStartup.xml** information into the Magic Space to enable projects to start automatically.
4. All GSAs in the grid load the Magic xpa servers as defined.



When a Magic xpa server is started by a GSA, it is automatically directed to the grid according to the **XAP_LOOKUP_LOCATORS** environment variable set in the **GigaSpaces-xpa\bin\setenv.bat** file of that GSA.



```

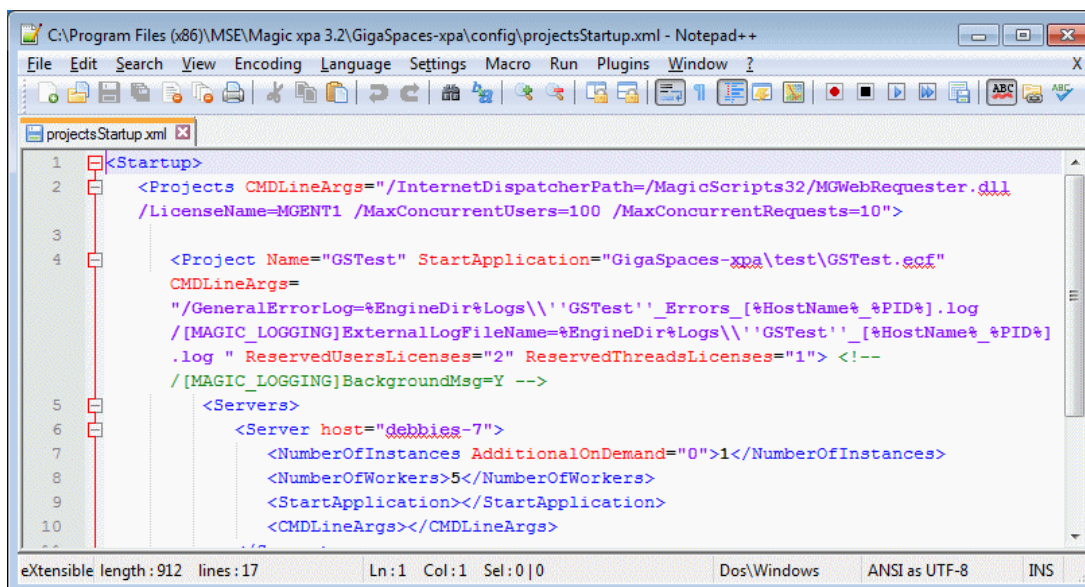
50 set ADDITIONAL_OPTIONS=
51
52 set XAP_LOOKUP_LOCATORS=debbies-7
53 if NOT "%DISCOVERY_PORT%" == "" (
54     set ADDITIONAL_OPTIONS=%ADDITIONAL_OPTIONS%
    -Dcom.sun.jini.reggie.initialUnicastDiscov
    %DISCOVERY_PORT%

```

In the following sections, you will read more about the steps described above. First though, it's important that you become familiar with the **projectsStartup.xml** file.

ProjectsStartup.xml file

The Grid Service Agent (GSA) is responsible for starting, monitoring, and terminating the Magic xpa processes. The startup instructions to the GSA are based on configurations defined in the `projectsStartup.xml` file. This file is located in the `GigaSpaces-xpa\config` folder.



```

1 <Startup>
2   <Projects CMDLineArgs="/InternetDispatcherPath=/MagicScripts32/MGWebRequester.dll
3     /LicenseName=MGENT1 /MaxConcurrentUsers=100 /MaxConcurrentRequests=10">
4     <Project Name="GSTest" StartApplication="GigaSpaces-xpa\test\GSTest.ecf"
5       CMDLineArgs=
6         "/GeneralErrorLog=%EngineDir%Logs\''GSTest''_Errors_[*HostName%_*PID%].log
7         /[MAGIC_LOGGING]ExternalLogFileName=%EngineDir%Logs\''GSTest''_[*HostName%_*PID%]
8         .log " ReservedUsersLicenses="2" ReservedThreadsLicenses="1"> <!--
9         /[MAGIC_LOGGING]BackgroundMsg=Y -->
10      <Servers>
11        <Server host="debbies-7">
12          <NumberOfInstances AdditionalOnDemand="0">1</NumberOfInstances>
13          <NumberOfWorkers>5</NumberOfWorkers>
14          <StartApplication></StartApplication>
15          <CMDLineArgs></CMDLineArgs>
16        </Server>
17      </Servers>

```

Two of the elements that you should be aware of in this file are:

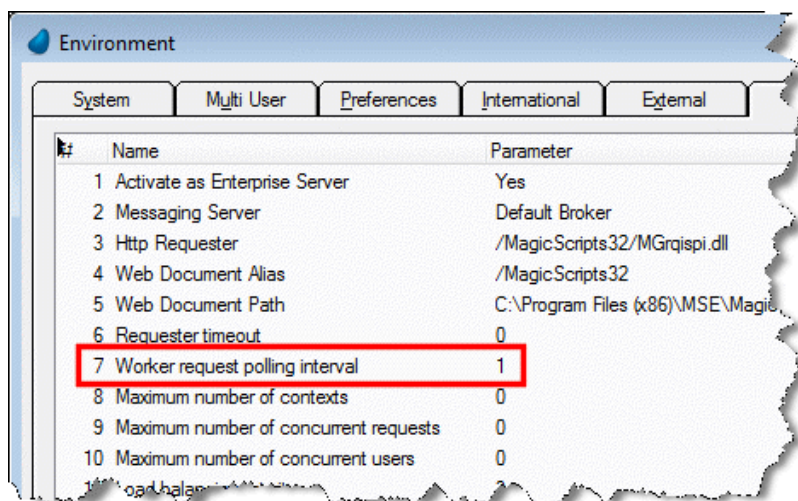
- **NumberOfInstances** – If you want several servers to run with exactly the same configuration, you can define a single server section in the `projectsStartup.xml` and define in this element how many of those sections will run. The default value is 1. A value of 0 means that the server will not be loaded on startup. In this element, you can use the **AdditionalOnDemand** attribute to set additional servers that will be available on a need-to-have basis.
- **NumberOfWorkers** – This element determines how many worker threads the server will start upon startup. You can set different numbers of worker threads for different Magic xpa servers. The number of workers that will execute a project at any given time is bound by the license. It is recommended to keep this value as low as possible, since each worker polls the space for requests.

```

<Servers>
  <Server host="debbies-7">
    <NumberOfInstances AdditionalOnDemand="0">1</NumberOfInstances>
    <NumberOfWorkers>5</NumberOfWorkers>
    <StartApplication></StartApplication>
    <CMDLineArgs></CMDLineArgs>
  </Server>
</Servers>

```

The interval between polling attempts can be configured using the **Worker Request Polling Interval** environment setting.



Automatically Starting Your Servers and Projects (Automatic Provisioning)

As you saw above, you do not manually start the **MgxpRuntime.exe** processes. When the Magic Space is deployed, the Grid Service Agent (GSA) automatically loads Magic servers to serve projects defined in the **projectsStartup.xml** file. You will need to update this file with the Magic projects and servers' details. In this file, any declaration at the Server level will override the declaration at the project level.

Note:

- Different **projectsStartup.xml** files can be used across the grid – during startup by the **StartProjects.bat** file and by the GigaSpaces Monitor.
- You can define repetitive values once in the **CMDLineArgs** attribute (optional) in the **Projects** element.
- When starting projects using the GSA (from the **projectsStartup.xml** file), the project name is automatically used as the value of the **ApplicationPublicName** environment setting.

Example ProjectsStartup.xml File

The following example of a **projectsStartup.xml** file defines a start-up configuration of the test application **GigaSpaces-xpa\test\GSTest.ecf**:

```
<Startup>
<Projects CMDLineArgs="/InternetDispatcherPath=/MagicScripts/MGWebRequester.dll /LicenseName=MGENT1 /MaxConcurrentUsers=100
/MaxConcurrentRequests=10">

  <Project Name="GSTest" StartApplication="\\Server1\Magic\gpa\GigaSpaces-xpa\test\GSTest.ecf"
  CMDLineArgs="/GeneralErrorLog=\\\\Server1\Magic\gpa\Logs\ 'GSTest' '_Errors_ [%HostName% %PID%].log
  @ '\\Server1\Magic\gpa\GigaSpaces-xpa\test\My test.ini'"
  ReservedUsersLicenses="2" ReservedThreadsLicenses="1">

    <Servers>
      <Server host="Server1">
        <NumberOfInstances>2</NumberOfInstances>
        <NumberOfWorkers>10</NumberOfWorkers>
      </Server>
      <Server host="Server2">
        <NumberOfInstances AdditionalOnDemand="1">1</NumberOfInstances>
        <NumberOfWorkers>5</NumberOfWorkers>
        <CMDLineArgs>/MaxContexts=200</CMDLineArgs>
        <StartApplication></StartApplication>
      </Server>
    </Servers>

  </Project>
</Projects>
</Startup>
```

This example file configures the following settings:

1. The application will be started on two server machines: **Server1** and **Server2**.
2. **Server1** will start two Magic xpa servers (engines), each with ten worker threads.
3. **Server2** will start one Magic xpa server (engine) immediately and another on demand (if all other engines will be busy), each with five worker threads. Each engine will be limited to 200 contexts.

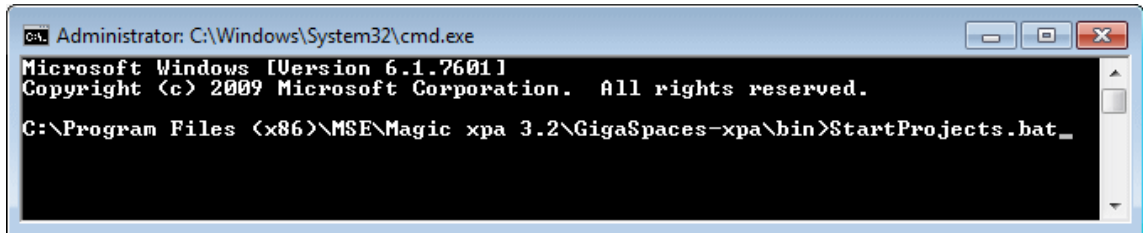
Note:

- You can have each server start a different application (<StartApplication> in the <Server level). However, it is recommended to use the same application for all of the servers, from a shared location, as shown in the <Project element.
- To use an @ini file with a space in its names, wrap the file name with single quotes.

Starting and Stopping the Servers and Projects from the Command Line

Although the GSA is responsible for starting the projects, you can also do it manually from the command line.

- To start the projects defined in the projectsStartup.xml from the command line, run the **StartProjects.bat** found in the **GigaSpaces-xpa/bin** folder.



```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\MSE\Magic xpa 3.2\GigaSpaces-xpa\bin>StartProjects.bat_
  
```

- To stop the projects running in the space, run the **ShutdownProjects.bat** found in the **GigaSpaces-xpa\bin** folder.
- To stop the servers running in the space, use the following syntax in the command line: **CommandLine.bat [-ShutdownProjects=%*] [-ShutdownServer=server name:process ID] [-Timeout=10]**

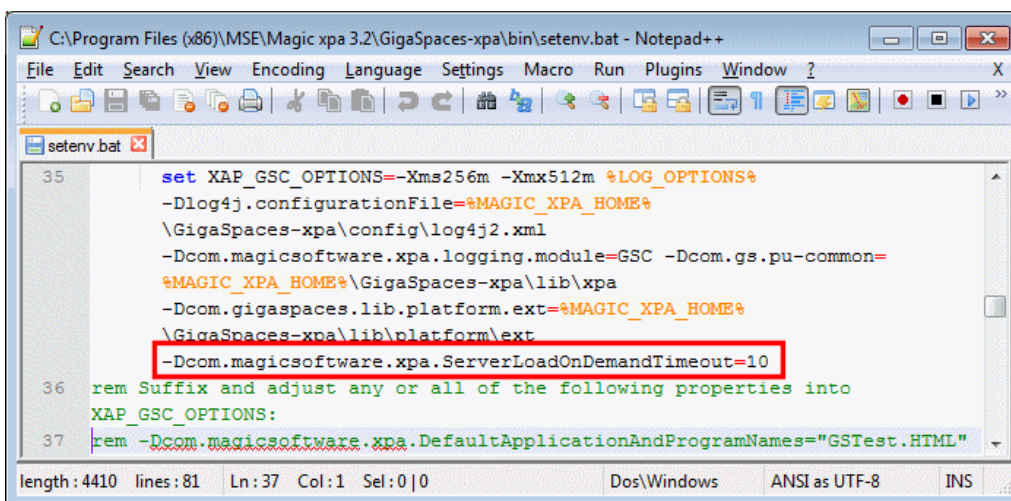
Some of examples of this are:

- **-ShutdownProjects=** will stop all servers (regardless of the applications they serve).
- **-ShutdownProjects=GSTest** will stop all the servers that serve the application GSTest.
- **-ShutdownServer=ABC:12345** will stop process ID 12345 in the server ABC. This lets you stop a specific instance of a server.

Loading Engines Dynamically

The Space middleware can automatically load a new Magic xpa server when receiving a request and there are no available servers to serve that request. If a request is not served within a certain timeout (by default 10 seconds:

-Dcom.magicsoftware.xpa.ServerLoadOnDemandTimeout=10), an additional Magic xpa server will be started. This timeout is set in the **setenv.bat** file.



```

C:\Program Files (x86)\MSE\Magic xpa 3.2\GigaSpaces-xpa\bin\setenv.bat - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
setenv.bat
35     set XAP_GSC_OPTIONS=-Xms256m -Xmx512m %LOG_OPTIONS%
      -Dlog4j.configurationFile=%MAGIC_XPA_HOME%
      \GigaSpaces-xpa\config\log4j2.xml
      -Dcom.magicsoftware.xpa.logging.module=GSC -Dcom.gs.pu-common=
      %MAGIC_XPA_HOME%\GigaSpaces-xpa\lib\xpa
      -Dcom.gigaspace.lib.platform.ext=%MAGIC_XPA_HOME%
      \GigaSpaces-xpa\lib\platform\ext
      -Dcom.magicsoftware.xpa.ServerLoadOnDemandTimeout=10
36 rem Suffix and adjust any or all of the following properties into
      XAP_GSC_OPTIONS:
37 rem -Dcom.magicsoftware.xpa.DefaultApplicationAndProgramNames="GSTest.HTML"
length: 4410 lines: 81 Ln:37 Col:1 Sel:0|0 Dos\Windows ANSI as UTF-8 INS
  
```

This Magic xpa server will be started on one of the servers defined in the **projectsStartup.xml** if the server was defined with **AdditionalOnDemand > 0** and the number of Magic xpa servers currently running on this server did not exceed the sum of **NumberOfInstances+AdditionalOnDemand**.

Stopping Projects with On-Demand Servers

When stopping a project (**GigaSpaces-xpa\bin\ShutdownProjects.bat**) or stopping a specific server, the servers that were already started will be stopped. However, servers waiting to be started on demand will still be started if requests are inserted into the space before restarting the project (**GigaSpaces-xpa\bin\StartProjects.bat**).

Lesson 4

Licensing

The Magic xpa license mechanism is floating by nature (both for threads and users), with an option to reserve a fixed (minimum) number of license threads or users for each project.

The licenses are available for use by all of the projects. The Space acts as the license server. Before executing a flow, the worker will attempt to check-out the license from the pool. When it has finished executing the flow, the worker will check the license back into the pool. This lets you make optimum use of licenses in a multi-project environment, by sharing licenses between servers. Therefore, when a particular project is idle, other projects can utilize its licenses.

When a Magic xpa server is started by the Space middleware:

- For each license feature (ENT and/or RIA), the Magic xpa server will check out (from the license file/server) and will write license tokens to the space only if that specific license feature (that is, a license feature with the same serial number) was not already written to the space. In practice, only one Magic xpa server will write license tokens to the space, regardless of the number of Magic xpa servers started simultaneously.
- If the host ID in the license feature is wrong, the Magic xpa server will stay alive, but it will not check out (from the license file/server) and will not write license tokens to the space.



Multiple RIA clients from the same machine consume one (RIA) license token.

All Magic xpa installations should be configured to point to the same license file (you can put it in a shared location).

The first engine that starts writes the number of licenses to the space. Then, other engines can also run. It is not required that the Magic xpa engine that allocated the license continues to run. Once it sets the thread and user count in the Space, it is not needed anymore to maintain the license. It can therefore be stopped and started like any other Magic xpa engine on the grid.



When working with the Space middleware, if you updated the license, you need to restart the space in order to use the updated license.

Reserving License Threads or Users for a Project

If you are running multiple projects, and some projects may consume all threads or users under stress conditions (for example, there are many Web services and/or HTTP requests), you should consider reserving a minimum fixed amount of license threads or users for critical projects that need to run continually.

You can use the **ReservedUsersLicenses** and **ReservedThreadsLicenses** attributes in the **projectsStartup.xml** file, to reserve the number of users and/or thread licenses per project.

License tokens reserved for a given project are acquired by workers in Magic xpa servers that serve that project before trying to acquire non-reserved license tokens.

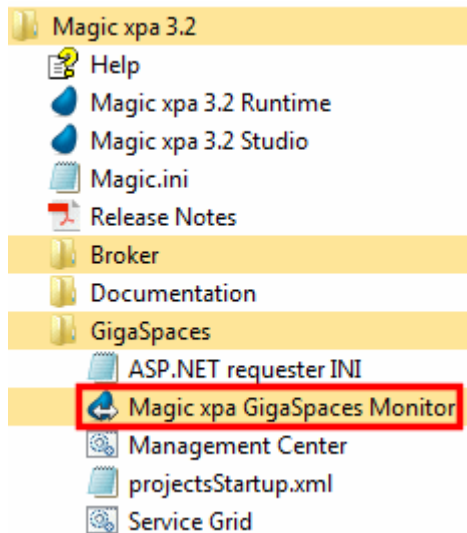
Lesson 5

Monitoring

The Magic Monitor is a tool that enables you to track the execution of your project by giving you accurate information about your projects in a single intuitive and easy-to-use dashboard. You can view the information for whole projects or you can select different levels within projects, and you can use filters to display information per engine or per context. In the Monitor, you can see the available servers, contexts, requests and license usage. The information displayed is updated regularly. The status of each project is taken from the Space.

Starting the Monitor

1. From the **Start** menu, click on the **Magic xpa 3.2** folder.
2. From the **GigaSpaces** folder, click the **Magic xpa GigaSpaces Monitor** option.



3. The Magic xpa GigaSpaces Monitor will open.
4. By default, it will connect to the current space, but you can change the connection from the **Connection Details** option in the toolbar.



The monitor is started by default from the **GigaSpaces-xpa\Monitor** folder. If the **\bin\setenv.bat** file exists and has the following setting:
set PROJECTS_STARTUP_XML=projectsStartup.xml, the full path of the **projectsStartup.xml** will be used by the monitor. Otherwise, the following default path will be used: **GigaSpaces-xpa\config\projectsStartup.xml**.

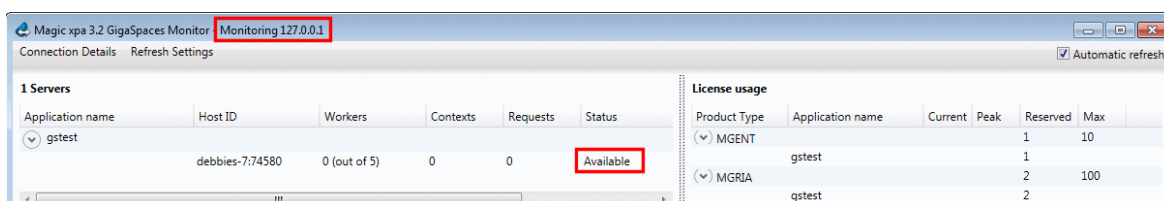
Verifying that the Applications Started Properly

By default, the **projectsStartup.xml** file is defined with a test application named **GSTest**. The cabinet file of this application is located in the **GigaSpaces-xpa\Test** folder.

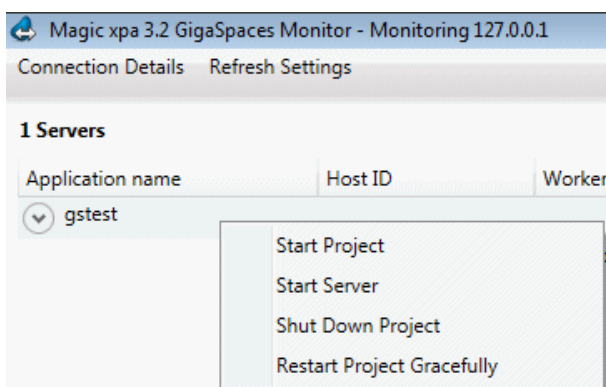
As mentioned earlier, when the grid is started, a Magic xpa server will be started with this application.

To verify that a Runtime engine was started and that the application is ready for use, you can look at the Monitor for the following:

1. If the connection to the space was successful, the monitor's window title will display the server being monitored. For example: **Monitoring 127.0.0.1**.
2. The monitor should also show the entries defined at the **projectsStartup.xml** file in the Servers list. The **Status** of the application entry should be **Available**.



3. Right-click on the server and you will see various options available to you.



Testing that an Application Can Connect to the Space

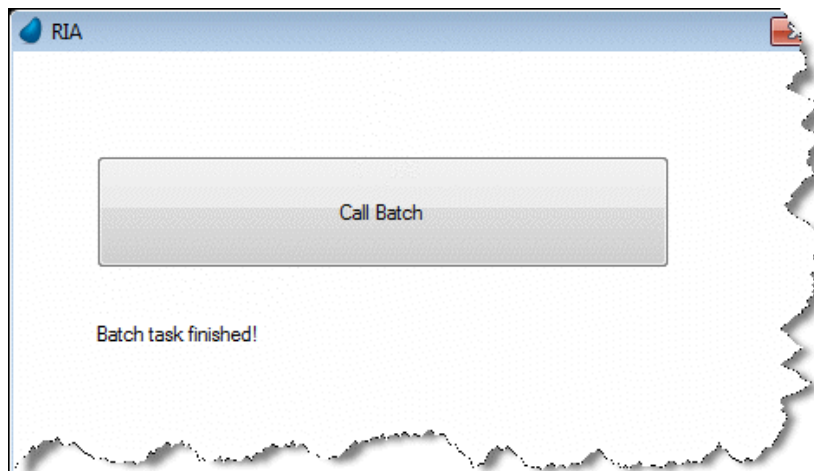
You can test that the application was properly deployed by executing any of the following batch files from the **GigaSpaces-xpa\test** folder:

- **RIATest.bat** – This file starts a RIA client that connects to the GSTest application.
- **BatchTest.bat** – This file starts a Batch program from the GSTest application.
- **HTMLTest.url** – This shortcut starts an HTML-based program from the GSTest application.

For the purposes of this seminar, let's run the **RIATest.bat** file.

1. Go to the **GigaSpaces-xpa\test** folder.
2. Click the **RIATest.bat** file.

- Click the **Call Batch** button. You should get the following screen.



- Go to the Monitor and see that the execution of the test files added requests to the **Requests log** section.

Requests log			
ID	Context ID	Application name	Prog...
12:13:29.9878_18446744073172615167	3387000000000007	gstest	RIA
12:13:29.9348_18446744072834458319	3387000000000007	gstest	RIA
12:13:29.8708_18446744072903536123	3387000000000007	gstest	RIA
12:13:19.9858_18446744073304604391	3387000000000006	gstest	RIA
12:13:19.8598_18446744073172679804	3387000000000006	gstest	RIA
12:11:58.2076_18446744072320155487	3387000000000006	gstest	RIA
12:11:51.9630_18446744073171841535	3387000000000006	gstest	RIA [Do
12:11:51.8570_18446744073688449021	3387000000000006	gstest	RIA [Dow
12:11:51.7490_00000000001668497350	3387000000000006	gstest	RIA [Dow
12:11:51.7080_18446744072534822783	3387000000000006	gstest	RIA [Dow
12:11:51.6790_18446744072534822783	3387000000000006	gstest	RIA [Dow

If this does not work, see the **ASP .NET Web Requester Installation** section in the Troubleshooting lesson for information about how to solve this.

Lesson 6

Space Deployment

As you saw in an earlier lesson, by default, the Magic xpa operating system service starts the Grid Service Agent (GSA), which in turn runs an application called **deployAndStartProjects** (located in the **gs-agent.bat** file) that is responsible for the Magic Space deployment. The Magic Space can be deployed only once, so it will only be deployed by one of the started operating system services.

The deployment process uses SLA settings defined in the **MgxpGSSpace_sla.xml** file, which is found under the **GigaSpaces-xpa\config** folder. This file will be explained in more detail in the [Space Clustering](#) section.

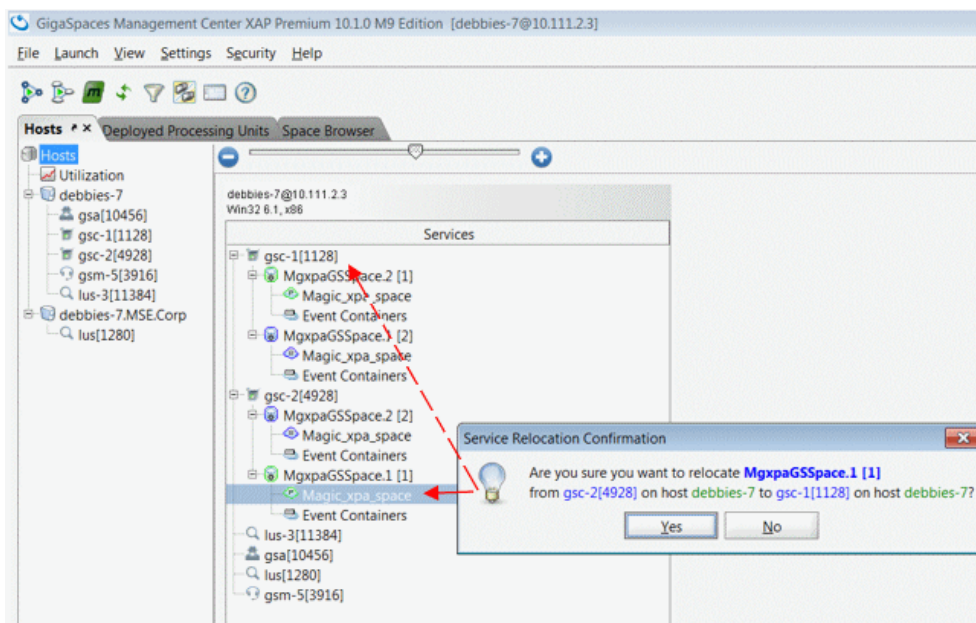
The deployment process will attempt to spread the partitions on the available containers (GSCs) in such a way that a single server failure will not affect the Magic Space operation and will not cause any data loss. This provisioning process is automatic, but once complete it will not rearrange itself.

If only one machine was running during the Magic Space deployment process, and there was no restriction in the SLA definition related to a single machine (**max-instances-per-machine**), this machine will hold all the partitions. Containers starting on other machines after the deployment was complete will not hold any Magic Space partitions, and the single machine that is currently running the Magic Space is now considered a single point-of-failure.

When you have more than one machine that is part of the grid, you will want to have control over when the Magic Space is deployed. When the Grid Service Agent (GSA) loads, and the machine becomes a part of the grid, that machine will not host a part of the Magic Space if there is already a Magic Space deployed on the grid.

To spread the partitions over multiple machines when one machine holds all of the partitions, you have the following options:

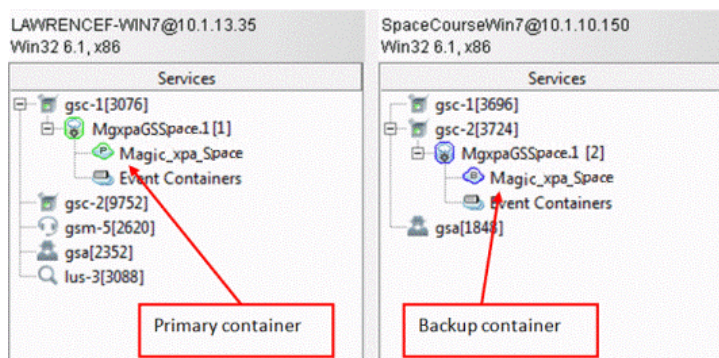
1. You can manually rearrange the partitions from the GigaSpaces UI. To do this, open the GigaSpaces UI **Hosts** tab, and stand on the **Hosts** entry at the top of the hierarchy tree on the left. In the **Services** pane, on the right side of the GigaSpaces UI screen, you will see a tree of containers and partitions. You can now select a partition (either primary or backup) and drag it to a different container, as shown in the following image.



2. You can restart the backup GSC and GigaSpaces will provision the grid. You do this as follows:
 - a. Park on the GSC node of the backup partition.
 - b. From the context menu, select **Restart**.

GigaSpaces will attempt to place the backup container on the second computer, as you can see from the image below. This provides redundancy for your application. If the secondary machine is not available, GigaSpaces will create the backup partition on the current machine.

When the secondary machine becomes available again, GigaSpaces may not automatically reposition the backup on the secondary computer. You may need to perform the operation manually.



3. You can use the **max-instances-per-machine** restriction in the SLA. This method should be restricted to a cluster of at least three machines, and it ensures that at least two machines in the grid will run the Space partitions.
 - a. In the **MgxpGSSpace_sla.xml** file, define the **max-instances-per-machine = "1"** entry as explained in the [Space Clustering](#) section.
 - b. When the automatic deployment process starts, it will not be completed until at least two machines are hosting the Space partitions.

Grid Components – Memory Allocation

Memory allocation for the various GigaSpaces entities is determined in the **setenv.bat** file (**GigaSpaces-xpa\bin**). In this batch file, you'll find settings for the GSA, GSM, and LUS entities:

- set XAP_GSA_OPTIONS=-Xmx64m
- set XAP_GSM_OPTIONS=-Xmx64m
- set XAP_LUS_OPTIONS=-Xmx64m

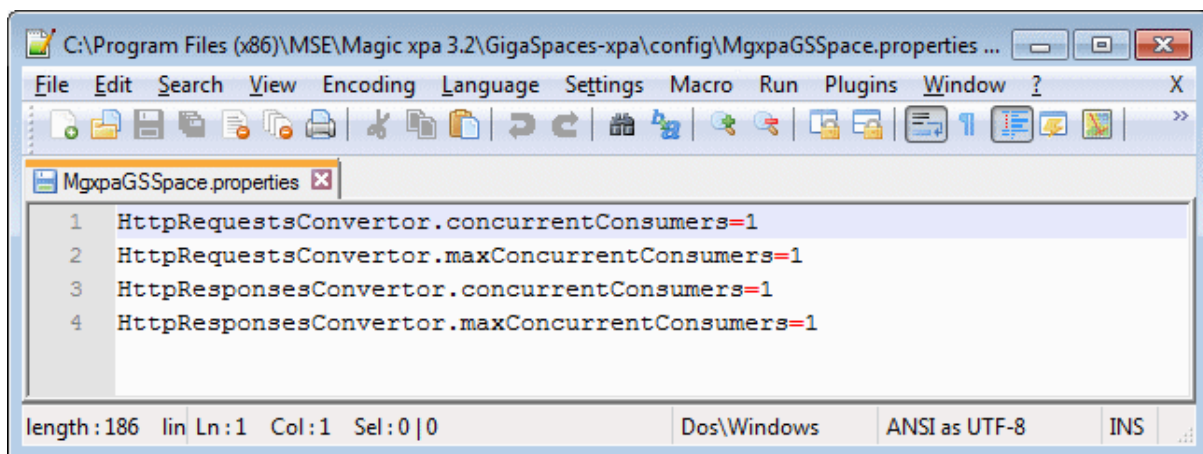
These entities have quite a small memory footprint, so you can leave these settings as is.

The GSC is the container that runs the Space partitions and holds all of the data that flows through the projects. The GSC entity in the batch file is:

- set XAP_GSC_OPTIONS=-Xms256m -Xmx512m

Configuring Event Containers' Concurrency

Magic xpa can control concurrency externally by changing the **HttpRequestsConvertor** and **HttpResponsesConvertor** event containers in the **MgxpGSSpace.properties** file, which is located at **GigaSpaces-xpa\config**. If the file is omitted, or certain properties within the file are omitted, the default concurrency (1) will remain.

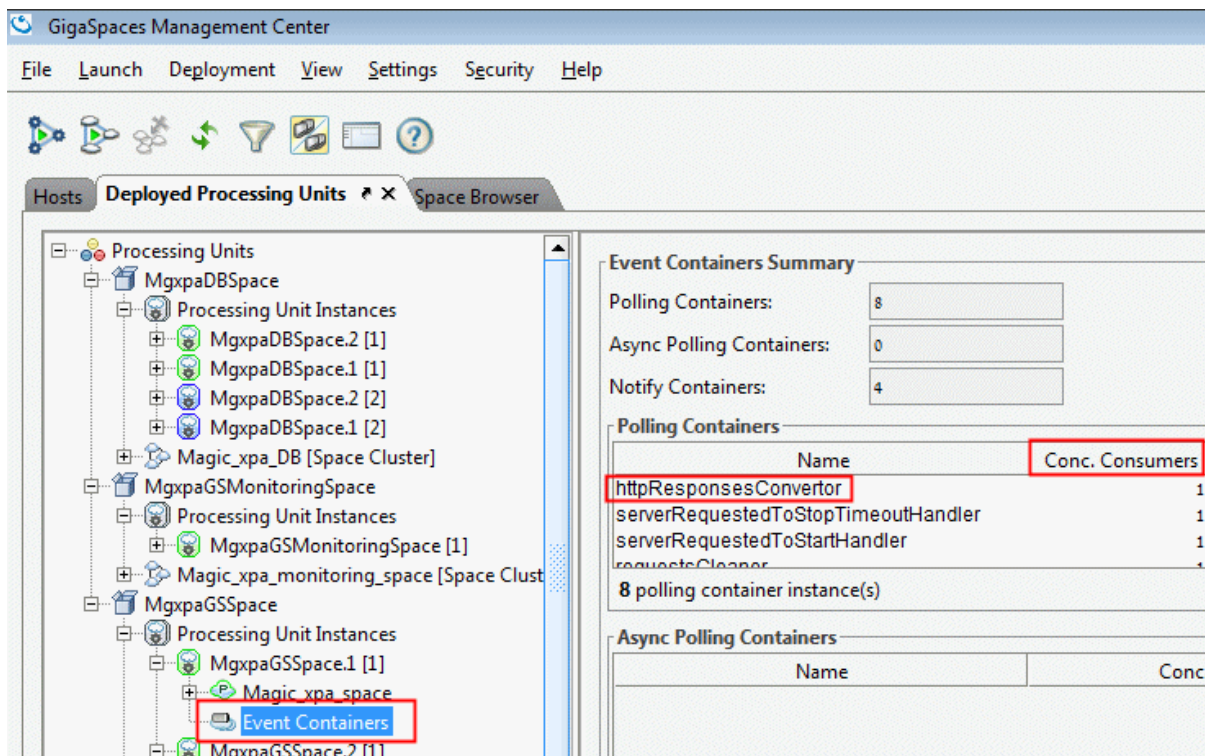


```

C:\Program Files (x86)\MSE\Magic xpa 3.2\GigaSpaces-xpa\config\MgxpGSSpace.properties ...
File Edit Search View Encoding Language Settings Macro Run Plugins Window ? X
MgxpGSSpace.properties
1 HttpRequestsConvertor.concurrentConsumers=1
2 HttpRequestsConvertor.maxConcurrentConsumers=1
3 HttpResponsesConvertor.concurrentConsumers=1
4 HttpResponsesConvertor.maxConcurrentConsumers=1
length:186 lin Ln:1 Col:1 Sel:0|0 Dos\Windows ANSI as UTF-8 INS
  
```


You can selectively increase the concurrency of the **HttpRequestsConvertor** and **HttpResponsesConvertor** event containers, which are potential bottlenecks in Web applications. For example, you can increase the concurrency in a clustered space, but not when the GSCs and the Magic xpa runtime processes are all on the same machine.

These event containers can be seen in the GigaSpaces Management Center's **Deployed Processing Units** tab as shown in the image below.



The screenshot shows the GigaSpaces Management Center interface. The left pane displays a tree view of Processing Units, with 'Event Containers' highlighted under 'MgxpGSSpace'. The right pane shows the 'Event Containers Summary' for the selected unit.

Event Containers Summary

Polling Containers:

Async Polling Containers:

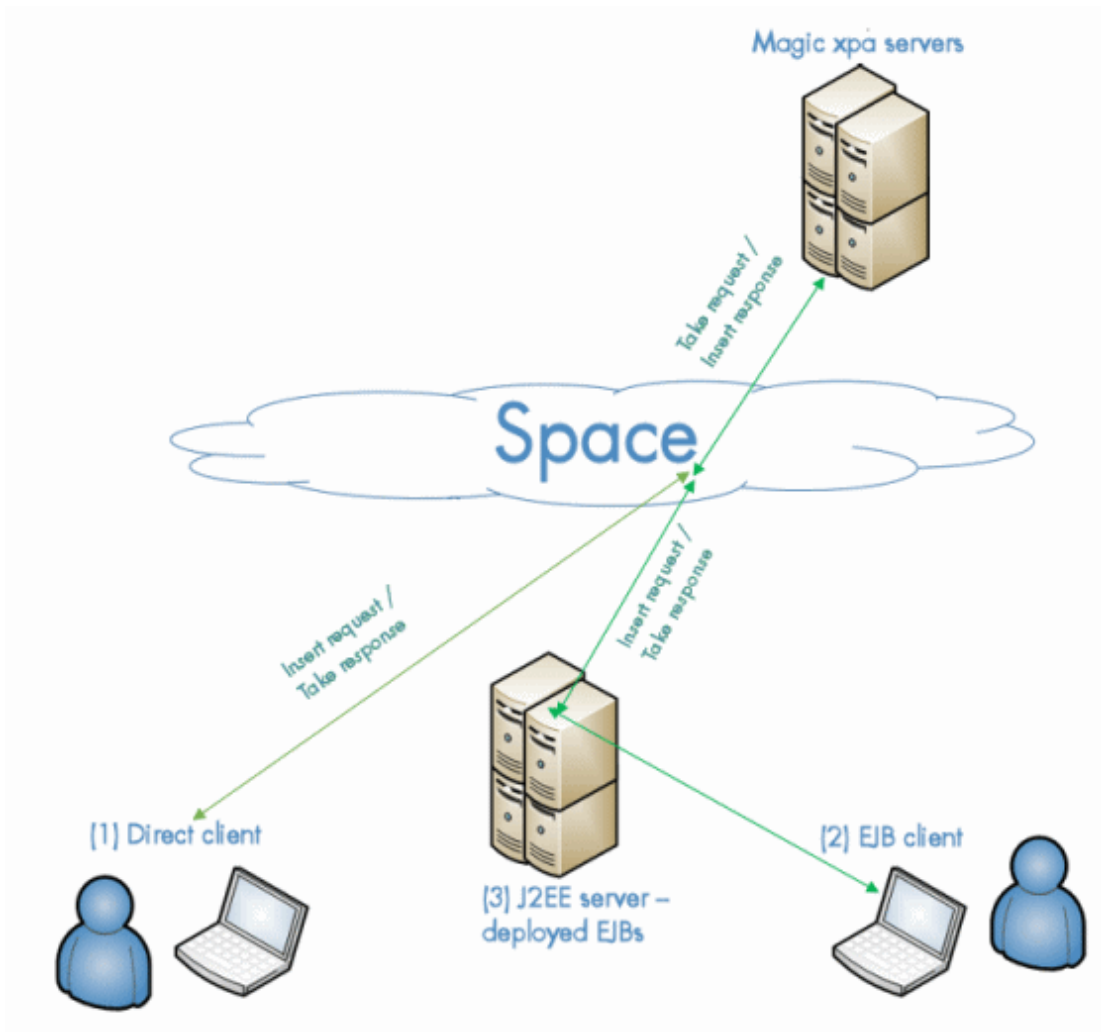
Notify Containers:

Polling Containers	
Name	Conc. Consumers
httpResponsesConvertor	1
serverRequestedToStopTimeoutHandler	1
serverRequestedToStartHandler	1
requestsCleaner	1
8 polling container instance(s)	

Async Polling Containers

Name	Conc.
------	-------

Accessing Magic xpa Servers over the Space from an EJB

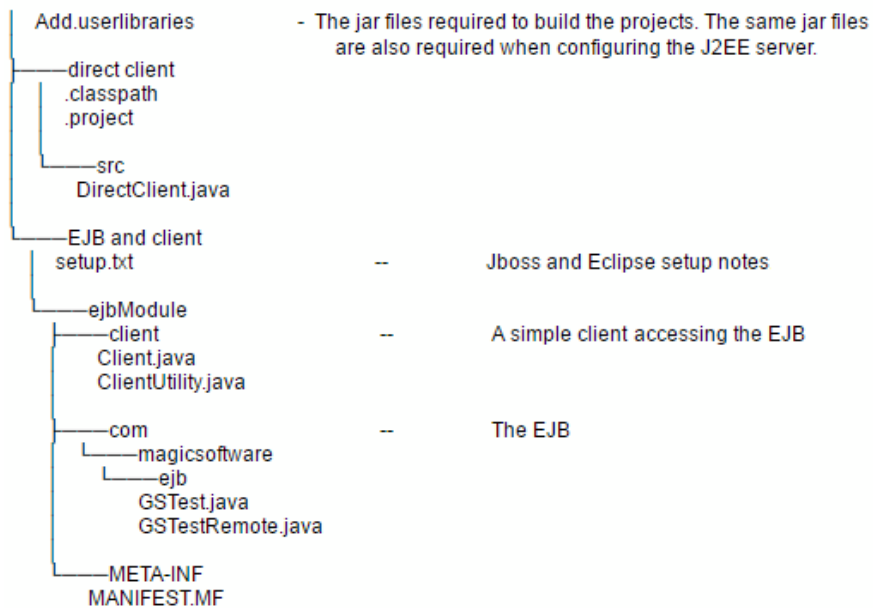


To access Magic xpa server(s) over the space from an EJB, the J2EE server (3) requires configuration similar to clients accessing the space (1):

- Lookup Locators, to connect to the space. For example:
Dcom.gs.jini_lus.locators=MyServer
- Magic xpa JAR files and GigaSpaces JAR files, listed in **GigaSpaces-xpa\samples\Add.userlibraries**.

The **GigaSpaces-xpa\samples** folder demonstrates a direct client (1), an EJB (3) and a client accessing the EJB (2).

The following folders are always installed in the `GigaSpaces-xpa\samples` folder:



Lesson 7

Space Clustering

In Space-based architecture, as you learned in the previous lesson, the Magic Space can reside on several computers that are viewed as a single logical unit. A cluster can be seen as a single "large" Space.

Space clustering defines the number of Space partitions, the number of partition backups, and the way they are spread on the available grid containers (GSCs). Space clustering is governed by the Service Level Agreement (SLA) definitions set in the **MgxpGSSpace_sla.xml** file. This means that the grid will always try to maintain the defined clustering when deploying the Space.

By default, this file defines two partitions with one backup each (four in total), and with a restriction that a primary partition and its backup partition cannot run under the same process.

For the grid to comply with the SLA definition, you need to ensure that you define enough GSCs. In the above default configurations, since a primary partition cannot run with its backup under the same process, you need at least two GSCs running for a successful Space deployment.

The most common SLA settings are:

1. **cluster-schema** – This should always be set to **partitioned-sync2backup**, which means that data can be in partitions and each partition can have a backup that is synchronized with it.
2. **number-of-instances** – The required number of Space partitions, meaning instances of the Magic processing unit, which will be loaded. The default is **2**. If you have a lot of data in memory, you may need to increase this number.
3. **number-of-backups** – The number of backup partitions for each primary partition. During development you can decide that you do not need a backup and you can set this value to **0**. If the **number-of-instances="2"** and the **number-of-backups="1"**, there will be four instances of the Magic processing unit.
4. **max-instances-per-vm** – When this is set to **1**, you ensure that a primary partition and its backup(s) cannot be provisioned to the same JVM (GSC). The number of instances of the same partition that will be deployed in the same JVM (GSC), that is, under the same process. If you left the default as is, **max-instances-per-vm="1"**, the primary and backup instances of the same partition will not be deployed on the same GSC.
5. **max-instances-per-machine** – When this is set to **1**, you ensure that a primary partition and its backup(s) cannot be provisioned to the same machine. Setting this to **1** should be restricted to a cluster containing a minimum of three machines. Then, if one of the machines fails, the lost partitions will move to the third machine. Or, it can also be used in a two machine cluster, but there is a risk having primary partitions with no backup until the second machine is back up and running.

Here are some SLA examples:

1. For single partitions with two backups, and primary and backup partitions on separate GSCs, set the following in the **MgxpGSSpace_sla.xml** file:

```
<os-sla:sla cluster-schema="partitioned-sync2backup" number-of-instances="1"
number-of-backups="2" max-instances-per-vm="1">
```

The above example requires at least **three** containers on a single machine. Each container will hold a single partition.

Note: Using two backups is not recommended. This example is brought here to show how the required number of GSCs is calculated.

2. For two partitions with one backup each, and primary and backup partitions on separate GSCs, set the following in the **MgxpGSSpace_sla.xml** file:

```
<os-sla:sla cluster-schema="partitioned-sync2backup" number-of-instances="2"
number-of-backups="1" max-instances-per-vm="1">
```

The above example requires at least **two** containers on a single machine. Each container will hold two partitions.

3. For two partitions with one backup, and primary and backup partitions on separate machines, set the following in the **MgxpGSSpace_sla.xml** file:

```
<os-sla:sla cluster-schema="partitioned-sync2backup" number-of-instances="2"
number-of-backups="1" max-instances-per-machine="1">
```

The above example requires at least **two machines** with at least **one** container on each machine. In each machine, the container will hold two partitions. If there is a cluster of two machines, and one of the machines fails, the Magic Space deployment will be incomplete (compromised) and no backup partition will replace the lost backup partitions until the failed machine starts up again.

*** The use of **max-instances-per-machine="1"** should be restricted to a cluster containing a minimum of three machines. Then, if one of the machines fails, the lost partitions will move to the third machine.

*** The number of GSCs is defined in the **gs-agent.bat** file, found under the **GigaSpaces-xpa\bin** folder. In the command starting with **call gs-agent.bat**, you should define the number of GSCs to match the number of required partitions by modifying the number next to the **gsa.gsc** parameter.

Note: If you are running your projects on a cluster, make sure that all of the machines' clocks are synchronized.

The recommended deployment method is to install Magic xpa on each node that participates in the cluster, and to deploy the project metadata to a shared network folder. It is also recommended to deploy the internal database on a separate machine.

Recommendations for Working with a Cluster

When working with a cluster, it is recommended to:

- Install Magic xpa on each node that participates in the cluster.
- Place the `projectsStartup.xml` (the project's metadata) in a shared network location and set `GigaSpaces-xpa\bin\setenv.bat` in all of the cluster's machines to access the same file. For example:
set `PROJECTS_STARTUP_XML=\\MyServer\config\projectsStartup.xml`.

Server Installation

When working in a cluster, two machines in the cluster need to be allocated to server as Lookup Service (LUS) servers.

Installation Prerequisites

Before installing, you need to verify the following prerequisites:

- A local or directory user (i.e. Magic xpa admin user) is defined with appropriate permissions on each of the application servers. The Magic xpa admin user should have permissions to run as a service.
- All application servers have network access and read/write permissions for the Magic xpa admin user to access the shared projects folder.
- All application servers can communicate with each other through the discovery port and LRMI ports as described in the [Firewall Settings](#) section above.

Installation Settings

1. On each application server, run the Magic xpa installation. Follow the installation dialog boxes.
2. Select the **Custom** option.
3. Notice by default that the both middlewares are selected, **Broker** and **GigaSpaces**. Leave this as-is.
4. Once you reach the **GigaSpaces Configuration** screen, select the **Install the Grid Service Agent (GSA)** as a service check box.

Post Installation Configuration Steps

Installing the Server License

Magic xpa requires a license to run.

1. Copy the license file that you received after purchase, and copy it to all of the servers, or to the shared location. It is good practice to put the license file on a path where all servers will have access to it so that each server does not need its own license file.
2. Modify the `magic.ini` file with the license file location and feature name by updating the `[MAGIC_ENV] LicenseFile` entry to point to the shared license file. For example: `LicenseFile = \\10.1.1.6\licenses\License.dat` and the `[MAGIC_ENV] LicenseName` to the production license name. For example `LicenseName = MGRIA`.

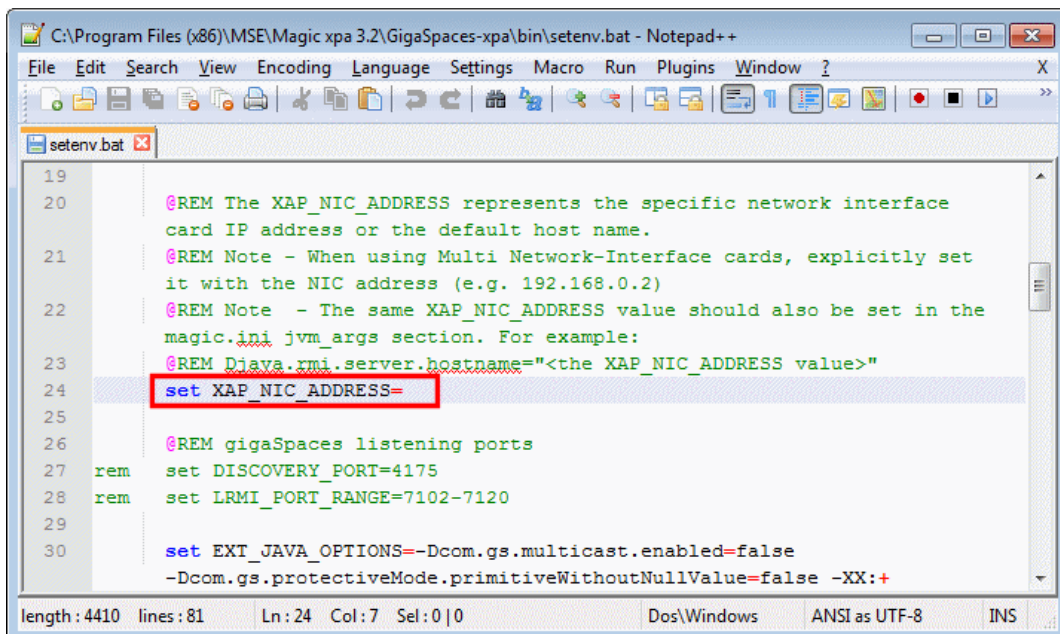
Note:

- Aside from the server hosting the license according to the license's `HOSTID` flag, any other server that loads will simply ignore this license.
- A special license is required for deployment on GigaSpaces.

Configuring Multiple Network Cards

If your application servers have multiple network cards, configure the use of a specific card for the Magic xpa server, as follows:

1. Modify the `NIC_ADDR` value found in the `<Magic xpa installation>\GigaSpaces-xpa\bin\setenv.bat` file to hold either the IP assigned to this network card or the name of the network card itself. For example: `NIC_ADDR=10.1.1.11` or `NIC_ADDR="#eth0:ip#"`, where `eth0` is the name of the network card.



```

C:\Program Files (x86)\MSE\Magic xpa 3.2\GigaSpaces-xpa\bin\setenv.bat - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
setenv.bat
19
20  @REM The XAP_NIC_ADDRESS represents the specific network interface
21  @REM card IP address or the default host name.
22  @REM Note - When using Multi Network-Interface cards, explicitly set
23  @REM it with the NIC address (e.g. 192.168.0.2)
24  @REM Note - The same XAP_NIC_ADDRESS value should also be set in the
25  @REM magic.ini jvm_args section. For example:
26  @REM Djava.rmi.server.hostname="<the XAP_NIC_ADDRESS value>"
27  set XAP_NIC_ADDRESS=
28
29  @REM gigaSpaces listening ports
30  rem set DISCOVERY_PORT=4175
31  rem set LRMI_PORT_RANGE=7102-7120
32
33  set EXT_JAVA_OPTIONS=-Dcom.gs.multicast.enabled=false
34  -Dcom.gs.protectiveMode.primitiveWithoutNullValue=false -XX:+
length: 4410 lines: 81 Ln: 24 Col: 7 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS
  
```

Tip: You can find the network cards' IPs and names by running the following script:

platform-info.bat -verbose

Then, look for the section called **Network Interfaces Information**.

The **platform-info.bat** files should be extracted from the **<Magic xpa installation folder>\GigaSpaces\bin\advanced_scripts.zip** file.

2. Add the **NIC_ADDR** value to the **magic.ini** file's **jvm_args** section:
-Djava.rmi.server.hostname=**<your network card IP address here>**

Note: The host name or IP address should not be surrounded by quotation marks.

Deploying on Multiple Servers

When the space is deployed on multiple servers, you should:

1. Configure the GSA in all of the servers to deploy the space and start the Magic xpa servers. This is done in the **gs-agent.bat** file, found in the **GigaSpaces-xpa\bin** folder as follows: **gsa.deployAndStartProjects 1**.
2. Start two local LUSs (LookUp Services) and two global GSMs (Grid Service Managers):
 - a. In two pre-selected machines, set **gsa.lus 1** in the **gs-agent.bat** file.
 - b. In each machine in which a GSA is started, set **gsa.global.gsm 2**. For an explanation about local and global setup refer to GigaSpaces's documentation by going to **Home > Administration > Runtime Configuration**. For example (GigaSpaces V10.1):
<http://docs.gigaspaces.com/xap101adm/runtime-configuration.html>.
3. In the **setenv.bat** file (**GigaSpaces-xpa\bin**), set:
XAP_LOOKUP_LOCATORS=host1,host2
(where **host 1** and **host 2** are the two machines where the LUSs were started). This setup should be applied in the **setenv.bat** file of each machine in which a GSA is started.
4. In each Web requester's **Mgreq.ini** file (found in **Scripts\config**), set **LookupLocators=host1,host2**.
5. In the GigaSpaces Monitor, set **host1,host2** in the **Connection details**.

Deploying Web Requesters on Different Servers

If you have a configuration where the Web servers are deployed separately, on a front-end server residing in the DMZ firewall, the server requires a different setup sequence.

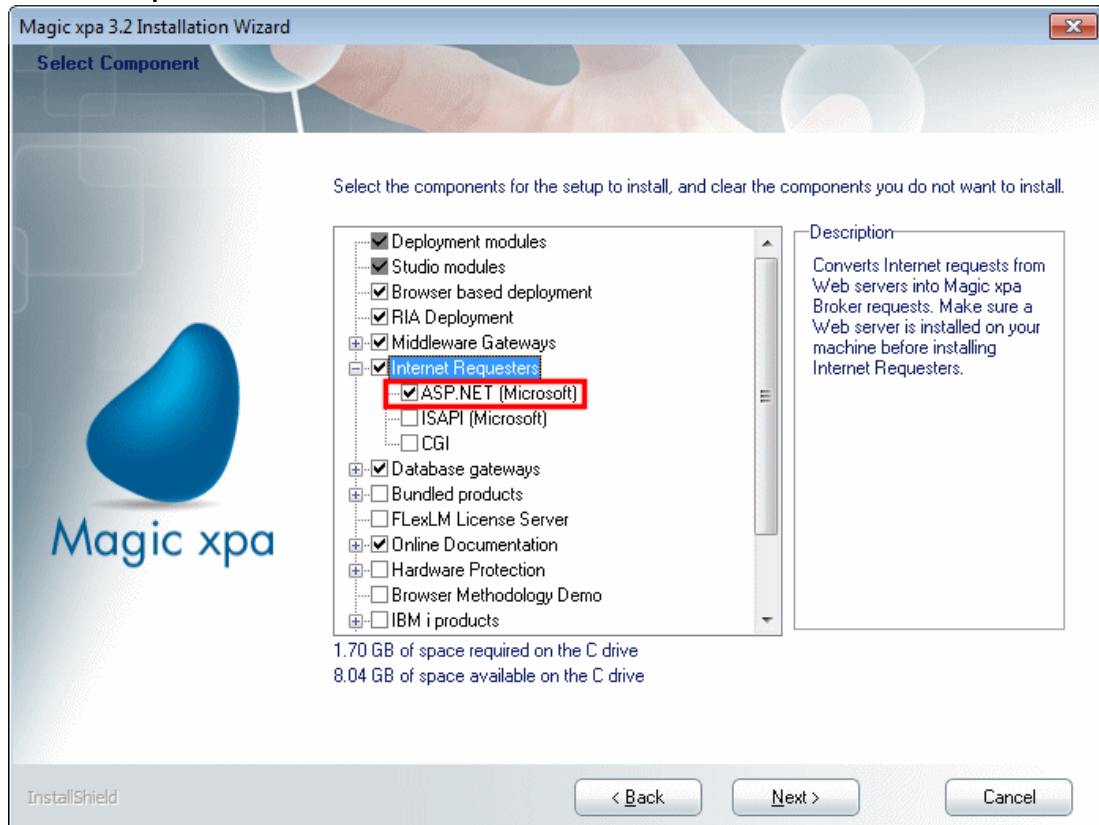
Prerequisites

For the Web requester, IIS7 is required.

Installation

On the Web service server located at the DMZ, you should install Magic xpa with the following configuration:

1. In the **Product Components** screen, select the **ASP .NET (Microsoft)** option in the **Internet Requesters** section.

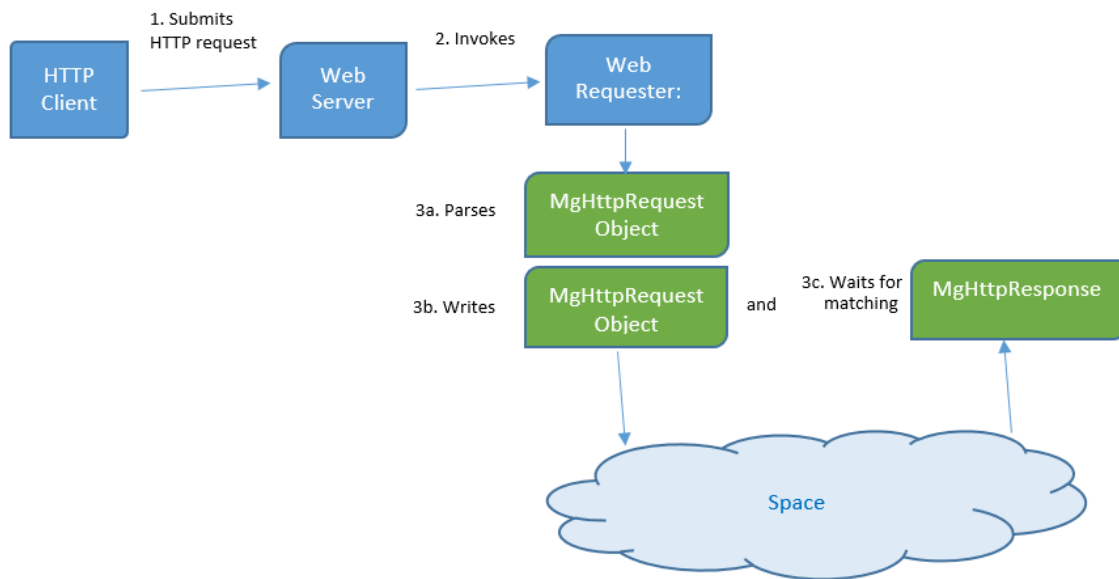


2. Uncheck the **Install the Grid Service Agent (GSA) as a service** check box.
3. To enable the front-end requester to connect to the Space, define the same value for the locators that you defined for the rest of the application servers.

These configurations will enable the Web requester to communicate with the space.

Since the Magic xpa Service will not be running on this machine, it will only be used to host the external requesters and will not run any of the grid components or the Magic xpa engines.

Flow of Web Requests



The flow of Web requests is as follows:

1. An HTTP client (be it a browser, a RIA client, or any software that makes HTTP calls) submits an HTTP request to the Web server.
2. The Web server invokes the Web requester (GigaSpace's Web requester is the **MgWebRequester.dll** file located at: **Scripts\bin**).
3. The Web requester:
 - a. Parses the HTTP request
 - b. Writes an **MgHttpRequest** object into the space
 - c. Waits for a matching **MgHttpResponse** object with the same **exchangeId** field.

Lesson 8

Troubleshooting

Troubleshooting is part of the life cycle of any project. It is important to have an understanding of the potential causes of problems and how to solve them.

This lesson covers various topics, including:

- Log files
- Space deployment issues
- Memory issues
- Various statuses

Logging

Magic xpa related logging uses the **log4j** infrastructure. The logging level is set by default to **info**, and is defined in the `<Magic xpa installation>\GigaSpaces-xpa\config\log4j2.xml` file.

The logs defined in the `log4j2.xml` file are written by default to the `<Magic xpa installation>\logs` folder. The logs' names are defined in the `log4j2.xml` file and include in their name the process ID of the process that wrote them.

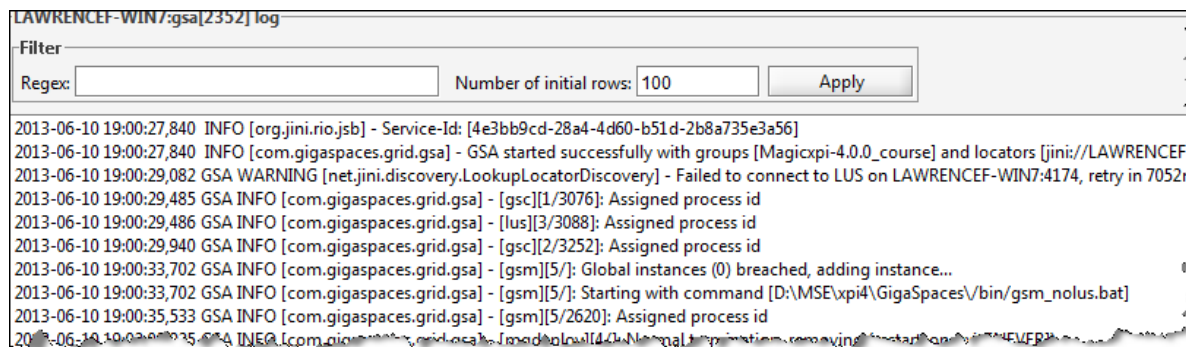
For additional information, see the **Log Files and Unknown Errors** topic in the *Magic xpa Help*.

Space Is Not Deploying

If the space does not deploy, you should:

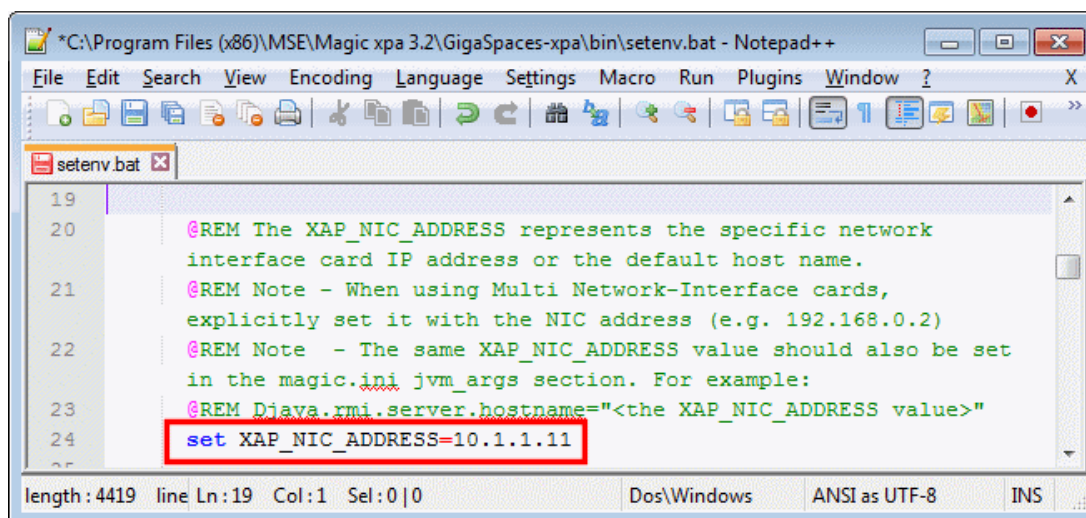
- Have a look at the log files. The log files will show whether the service was started, whether other components were started by the GS-agent, and any problem during startup.
- Make sure that the **Magic xpa GSA** service is started.
- If it is started, try stopping the **Magic xpa GSA** service, waiting for all grid processes (if they exist) to terminate, and restarting the **Magic xpa GSA** service.
- Check that there are no Java conflicts.
- Check the Java version. The Magic xpa engine can only be started with a 32-bit version.
- Check that the lookup locators server in the `setenv.bat` file is set correctly.
- If you did not select the **Install the Grid Service Agent (as a service)** check box during installation, the grid will not deploy. Run the `Install_GSA_service.bat` file from `GigaSpaces-xpa\OS_Service\Scripts` to install the GSA. With some operating systems, such as Windows 7 and above, you need to run this command using administrator credentials.

- Click on one of the running components and you will see the log for that component. Any errors will be seen in the log, as shown in the image below:



- For all of the application servers, if your machine is running multiple network interfaces, make sure that the **XAP_NIC_ADDRESS** value (in the **<Magic xpa installation>\GigaSpaces-xpa\bin\setenv.bat** file) is set to hold either the IP assigned to the network card or the name of the network card itself. Then, add the **XAP_NIC_ADDRESS** value to the magic.ini file's **jvm_args** section:
-Djava.rmi.server.hostname=<your network card IP address here>

Note: The host name or IP address should not be surrounded by quotation marks.



- When working in a clustered environment, if the grid entities are not available in the GigaSpaces UI, check that the firewall is not blocking the ports used by GigaSpaces. There are two settings that control the ports:
 - The Discovery port, which should be opened in the firewall.
 - The range of LRMI ports, which should be set to a fixed range and also opened in the firewall.

Memory-related Issues

If you encounter any memory-related issues with the GSC, try the following:

1. Go to the **setenv.bat** file, which is located at: **GigaSpaces-xpa\bin**.
1. Find the following line: **set XAP_GSC_OPTIONS=-Xms256m -Xmx512m**.
2. Increase the values.

Waiting for GS Agent Status

If you see the application in the Magic xpa GigaSpaces Monitor, but the application status is **Waiting for GS Agent**, this means that Magic xpa will keep retrying, every 60 seconds, to restart the Magic xpa server, until the GSA is started on the target machine. This status is used in the following scenario:

- Two or more machines are running on the grid.
- A Magic xpa server is started on one machine.
- The second machine is shut down for maintenance and restarted after a while (for example: two hours).

This status also appears when the host in the **projectsStartup.xml** was not defined properly.

In this case, the GS log (such as 'MgxpasGS_9744.log') located in the **Logs** folder under the Magic xpa installation will show the following error:

```
"ERROR: 2015-02-16 10:40:37,636 [Admin GS-startRequestedWatcher-1]: locateAgent - no GSA found.Looking for Host:[EYALR-8-bad]"
```

The log will also contain the following line before the error line:

```
"INFO: 2015-02-16 10:40:37,636 [Admin GS-startRequestedWatcher-1]: locateAgent - located Host:[EYALR-8-LP],IP:[10.111.2.21].Looking for Host:[EYALR-8-bad]"
```

You can see from this line that the **host** property in the **projectsStartup.xml** was defined as **EYALR-8-bad**, but the machine host name is **EYALR-8-LP** and the IP is **10.111.2.21**.

Solution: Change the **host** property in the **projectsStartup.xml** to contain a valid host name or IP address.

Failed to Start Status

If you see the application in the Magic xpa GigaSpaces Monitor, but the application status is **Failed to start**, this means that the engine was not started properly for one of the following reasons.

1. Lack of permission to start the Magic xpa Runtime engine.

In this case, the GSA log (such as '2015-02-16~09.58-gigaspace-gsa-eyalr-8-lp-10152.log') located in the **Logs** folder under the Magic xpa installation will show the following error:

```
"2015-02-16 09:58:57,507 GSA WARNING [com.gigaspace.grid.gsa] - [mgxpa][6/]:
Failed to start process. Caused By Cannot run program
"C:\Magicxpa30\GigaSpaces\..\MgxpaRuntime.exe" (in directory
"C:\Magicxpa30\GigaSpaces\.."): CreateProcess error=740, The requested operation
requires elevation; Caused by: java.io.IOException: Cannot run program
"C:\Magicxpa30\GigaSpaces\..\MgxpaRuntime.exe" (in directory
"C:\Magicxpa30\GigaSpaces\.."): CreateProcess error=740, The requested operation
requires elevation"
```

Solutions:

Make sure that the user that starts the GSA has administrator rights.

If you started the grid manually (via the **gs-agent.bat** file, make sure that you start the **Command Prompt** app with the **Run as Administrator** option set, or start using the **gs-agent-elevated.bat** file from the **GigaSpaces-xpa** folder.

2. The lookup locator's server address was not defined properly.

In this case, the log files located in the **Logs\GS** folder (the default **Test** project folder is **GigaSpaces-xpa\Test**) under the Magic xpa installation will show the following error:

Exception occurred during unicast discovery

Solution: Make sure that the address of the lookup locators server is properly defined in the **GigaSpaces-xpa\bin\setenv.bat** file.

ASP .NET Web Requester Installation

There are several known issues related to the ASP .NET Web requester installation.

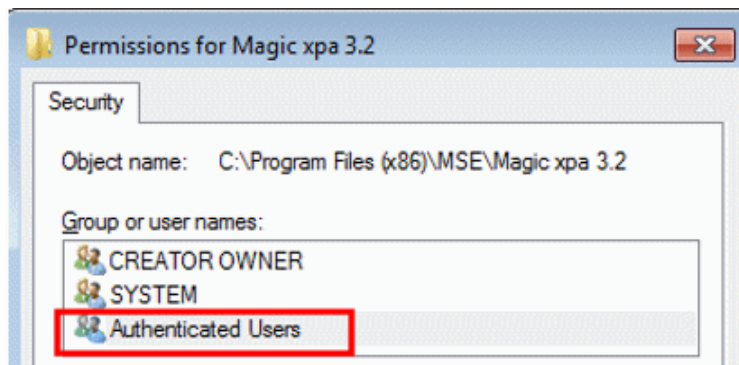
When calling Magic xpa from a browser or from a Rich Client program, you may get various errors:

1. An HTTP error

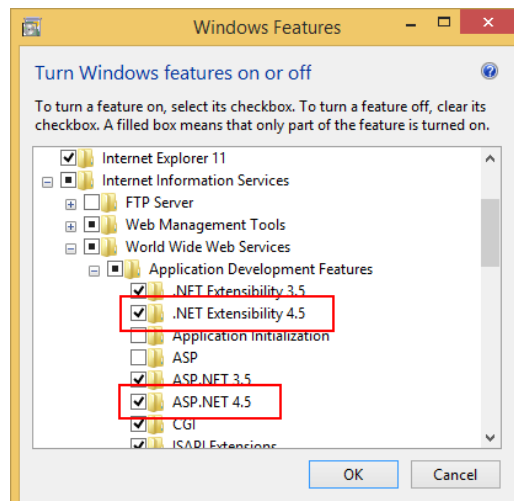
There are scenarios where you might receive the following error: "HTTP Error 500.19 - Internal Server Error. The requested page cannot be accessed because the related configuration data for the page is invalid."

Solution:

- a. Add the **Authenticated Users** to the Magic xpa installation folder's permissions, as the Web requester needs to access the installation's **JRE** and **GigaSpaces** folders.



- b. Install the **ASP.NET** and **.NET Extensibility** components (V4.0 or above) on your Web server.



2. Failure to load assembly

There are scenarios where you might receive the following error: "Configuration Error. **Description:** An error occurred during the processing of a configuration file required to service this request. Please review the specific error details below and modify your configuration file appropriately. **Parser Error Message:** Could not load file or assembly 'GigaSpaces.JavaToNet.DLL' or one of its dependencies. The specified module could not be found."

Solution: In this case, the MS VC++ 2010 Redistributable package is probably missing. You should install this package to fix this issue.

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 www.magicsoftware.com.



Magic is a 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. 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, 2016