

Magic xpi 4.6 アドバンスド 開発ガイド

クラスタ環境

での実行



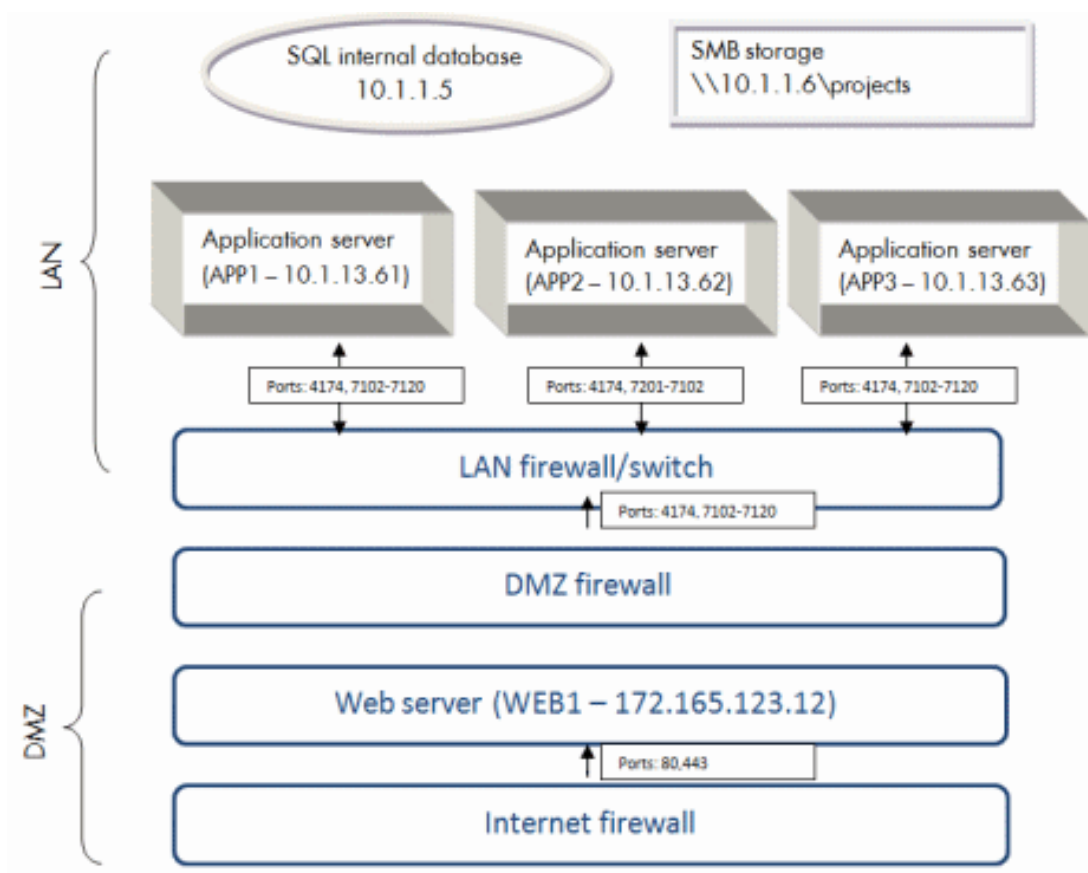
OUTPERFORM THE FUTURE™

はじめに	3
サーバのインストール	3
インストール概要	4
インストール前提条件	4
セットアップウィザード手順	5
セットアップウィザード終了後の設定手順	8
ファイアウォールの設定	10
プロジェクトの実装	11
プロジェクトフォルダの位置	11
プロジェクトの配置	11
プロジェクト自動起動の設定	11
ProjectsStartup.xml ファイルの例	12
Magic モニタ用に GigaSpaces を設定	13
コマンドラインからのプロジェクトの起動(オプション)	13
Web リクエストと SOAP サーバの実装	13
前提条件	14
インストール	14
Java Web リクエスト	14
スペースクラスタリング SLA	15
スペースの実装	17
グリッド コンポーネント-メモリの割当	20
GigaSpaces サービス設定	20
ライセンス	20
各プロジェクトのライセンススレッドの確保	21
ホストロックライセンス	21
Magic Software Enterprises について	22

はじめに

この文書では、高度なクラスター環境において、Magic xpi 4.6 のプロジェクトを実行するための推奨事項とガイドラインを提供します。また、高冗長性やセキュリティで保護された環境、スケーラブルな環境を実現するための必要なトピックをカバーします。

また、この文書では、小規模サイズのサーバクラスターとして、3つのアプリケーションサーバ、1つのデータベースサーバ、1つのフロントエンドのWebサーバ、及び共有ファイルシステム(SMB)を想定し、それらのインストールと構成の手順を説明します。以下にそのダイアグラムを示します。:



サーバのインストール

推奨される実行環境として、まず、クラスター内の各ノードにおけるアプリケーションサーバ(上図では APP1, APP2, APP3) に Magic xpi 4.6 をインストールします。プロジェクトは共有ネットワークフォルダに配置する必要があります。内部データベースは別のマシンに配置することもお勧めします。

アプリケーションサーバの中から、2台のコンピュータをルックアップサービス(LUS)サーバとして割り当てる必要があります。これら2台のマシンはクラスタのための名前解決サービスを提供します。

インストール概要

サーバインストールの手順を以下に示します。:

- 実行アーキテクチャを理解します。
- インストールの前提条件準備と確認をします。
- セットアップウィザードを使用して、各アプリケーションサーバに Magic xpi 4.6 サーバをインストールします。セットアップ時に、サーバの起動と初期構成が実行されます。
- セットアップウィザード後の構成手順を実行します。
- プロジェクトを共有フォルダに配置し、プロジェクトの起動を設定します。
- オプション: Web リクエストと SOAP サーバの配置を行います。

インストール前提条件

インストール前に、次の前提条件を確認する必要があります。:

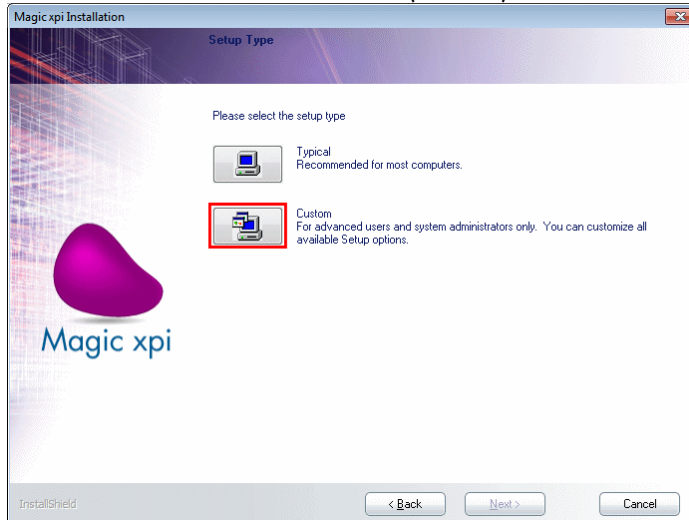
- 各アプリケーションサーバに適切な権限を持つローカルユーザまたはディレクトリユーザ(例: Magic xpi admin user)を定義する必要があります。Magic xpi admin user には、サービスとして実行する権限が必要です。
- データベースサーバに適切な権限を持つデータベース管理ユーザ(例: Magic xpi administrator DB user)を定義する必要があります。Magic xpi administrator DB user には、データベースとテーブルを作成する権限が必要です。
- データベースサーバに適切な権限を持つデータベースサーバユーザ(例: Magic xpi server DB user)を定義する必要があります。Magic xpi server DB user はデータベースとテーブルを作成する権限を必要とせず、データベースとテーブルにアクセスするためだけの権限が必要です。
- 全てのアプリケーションサーバはネットワークに接続しており、また Magic xpi admin user は共有フォルダにアクセスし、読み込み/書き込みを行う権限がなければなりません。
- 全てのアプリケーションサーバはデータベースにアクセスすることができ、Magic xpi DB user でデータベースに接続できる必要があります。
- 全てのアプリケーションサーバは、ディスカバリポート、LRMI ポート、Webster ポート(以下を参照)を介して相互に通信できる必要があります。.

注: クラスタの適切な操作のために、全てのホスト上の時計を同期させる必要があります。

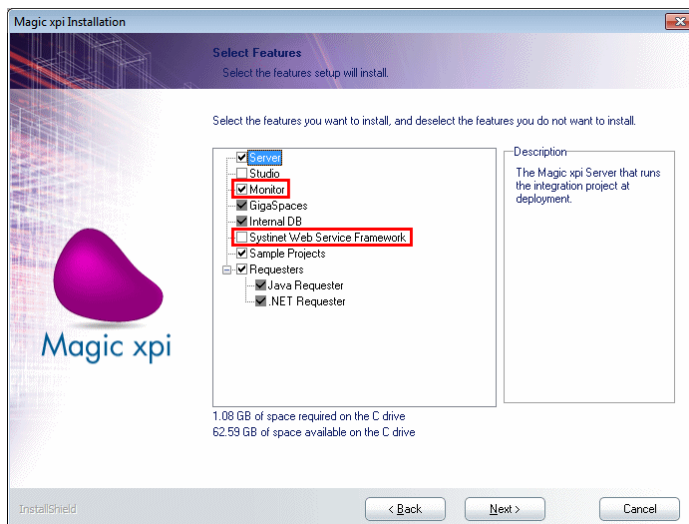
セットアップウィザード手順

各アプリケーションサーバで、インストールメディアからセットアッププログラム (setup.exe) を実行します。

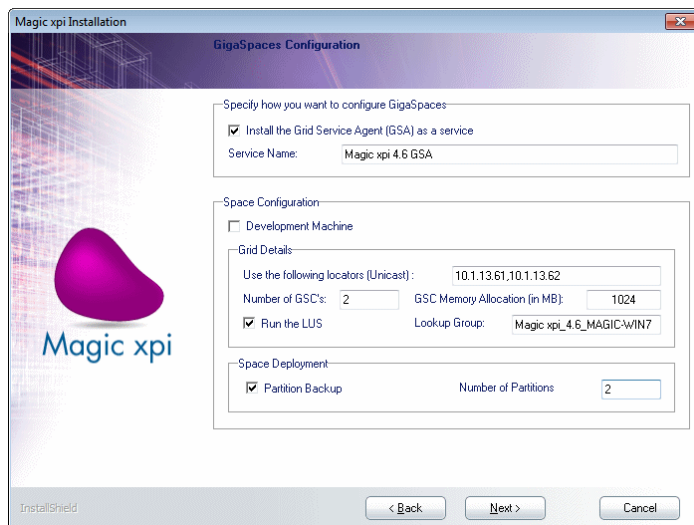
1. 設定ダイアログボックスに従い、インストールを行います。
2. セットアップタイプ画面で**カスタム(Custom)**を選択します。



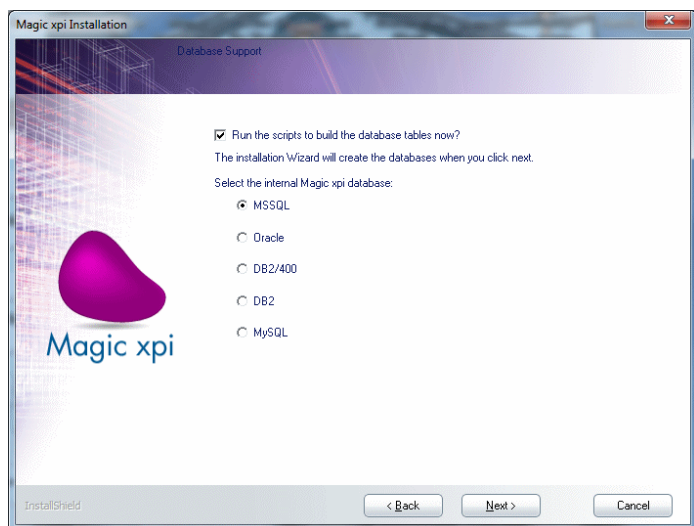
3. クラスタ内のいずれかのホスト上でのみ、**機能の選択画面**で **Monitor(モニタ)** チェックボックスを選択します。これにより、**Magic Monitor services** がホストマシンにインストールされます。全てのアプリケーションサーバのインストール設定において、同画面で **Systinet Web Service Framework** チェックボックスをオフにします。**Systinet Web Service Framework** はフロントエンド **Web** サーバに別途インストールします。



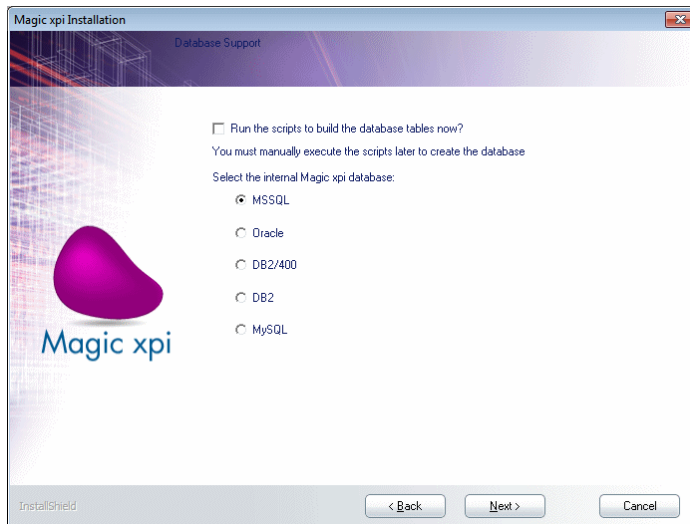
4. **GigaSpaces** の設定 画面で、以下のオプションを選択します。:
- グリッドサービスエージェント(**GSA**)をサービスとしてインストールするチェックボックスをオンにします。
 - 開発マシン チェックボックスはオフにします。
 - 次のロケーターを使用する(ユニキャスト)欄に、LUS サーバとして使用する 2 つのアプリケーションサーバの IP アドレス(カンマ区切り)を入力します。
 - GSC 数** 欄に **2** を入力し、**GSC メモリーアロケーション** 欄に **1024** を入力します。
 - 現在セットアップ中のアプリケーションサーバを LUS サーバとして構成する場合、**LUS 実行** チェックボックスをオンにします。
 - パーティション数 欄に **2** を入力し、パーティションバックアップ チェックボックスをオンにします。



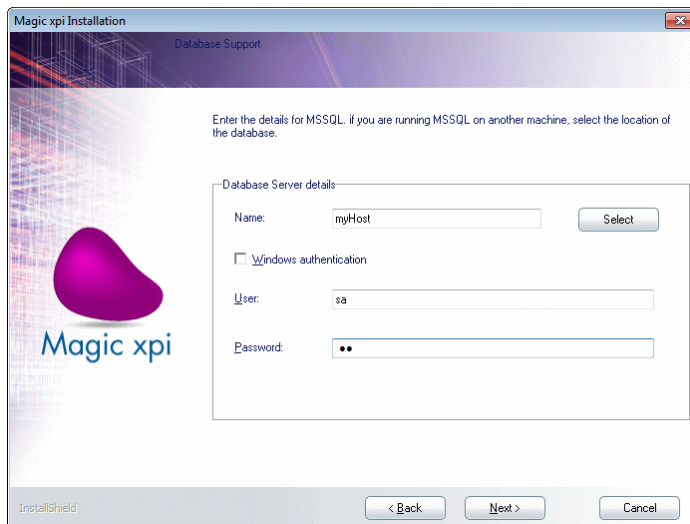
5. データベースのインストールを求める画面が表示されたら以下のオプションを選択します。:
- インストールする最初のアプリケーションサーバで**今すぐデータベーステーブルを作成するスクリプトを実行する**チェックボックスをオンにします。



- 他のアプリケーションサーバのセットアップでは**今すぐデータベーステーブルを作成するスクリプトを実行する**チェックボックスをオフにします。



- c. ○ 最初にインストールするアプリケーションサーバで、データベースサーバと Magic xpi administrator DB user(ユーザ、データベース、テーブルの作成権限を持つ)を入力します。



セットアップウィザード終了後の設定手順

Windows サービスユーザの設定

すべてのアプリケーションサーバーで、サービス コントロールパネルに移動して、サービスユーザを、プロジェクトで必要なすべてのリソースにアクセスするための十分な権限を持つユーザに変更します。たとえば、ユーザはプロジェクトを配置しているネットワークドライブに読み書きする権限を持っている必要があります。

内部データベースの設定

最初にインストールしたアプリケーションサーバ(セットアッププログラムが既に設定済み)を除き、全てのアプリケーションサーバの

<Magic xpi installation>\Runtime\config\datasource.xml ファイルを更新し、最初にインストールしたサーバのものと同一状態にします。

サーバライセンスのインストール

Magic xpi 4.6 を実行するにはライセンスが必要です。

1. Magic xpi ライセンス発行センターより送付されたライセンスファイルをコピーし、各アプリケーションサーバに配布するか、共有フォルダに配置します。
2. **Magic.ini** ファイルの[MAGIC_ENV]セクションにある **LicenseFile** エントリが共有フォルダ内のライセンスファイルの位置を指すように設定します。
例: **LicenseFile = \\10.1.1.6\licenses\License.dat**
3. 各プロジェクトフォルダ下の **ifs.ini** のライセンス名が本番サーバライセンスであることを確認します。
[MAGIC_ENV]LicenseName = IBPRSRVI

注: ライセンスファイルはクラスタ内のいずれかのホスト上で有効にする必要があります、このホストは Magic xpi エンジンの 1 つを起動する必要があります。このエンジンにより、ライセンスがスペースに一旦ロードされると、クラスタ内の他のエンジンからも使用できるようになります。

複数ネットワークカード構成の場合の設定(オプション)

アプリケーションサーバに複数のネットワークカードが存在する場合、特定のネットワークカードを使用するよう、以下のように Magic xpi 4.6 サーバを設定します。:

1. 該当ネットワークカードに割り当てられた IP、あるいはネットワークカード名を保持するために、以下のファイルを修正します。
ファイル名: <Magic xpi インストール先>\Runtime\Gigaspaces\bin\magicxpi-setenv.bat
修正エントリ: NIC_ADDR
修正例: NIC_ADDR=10.1.1.11 あるいは NIC_ADDR="#eth0:ip#" (eth0 はネットワークカード名)

ヒント: ネットワークカードの IP アドレスと名前は以下のスクリプトを実行することで取得できます。

<Magic xpi インストール先>\Runtime\Gigaspaces\bin\platform-info.bat -verbose
結果から **Network Interfaces Information** セクションを参照します。または以下に作成される GSA のログファイルからも取得することができます。
<Magic xpi インストール先>\Runtime\Gigaspaces\logs フォルダ

Network Interfaces Information セクションを参照します。

2. **Magic.ini** ファイルの **JVM_ARGS** セクションに **NIC_ADDR** を追加します。:
-Djava.rmi.server.hostname=<ネットワークカードの IP アドレス>

注: ホスト名や IP アドレスはシングルクォーテーションで囲む必要はありません。

ファイアウォールの設定

ここでは GigaSpaces インフラストラクチャを固定ポートで運用するために必要な設定について説明します。



このドキュメント文書内で取り扱っている 3 ノードのクラスタの場合、LRMI_PORT_RANGE を有効にしてポートの範囲を固定しを設定し、WEBSTER_PORT プロパティのポートを固定することをお勧めします。以下に説明するように、これらがファイアウォールで空けるポートとなります。

以下の 3 つの設定を有効にする必要があります。:

1. **DISCOVERY_PORT** – 検出リスニングポート
2. **LRMI_PORT_RANGE** – [LRMI](#) (グリッドコンポーネント間の内部通信用プロトコル)のポート範囲。
3. **WEBSTER_PORT** – リクエストとクラスタ間だけでなく、クラスタノード間でファイアウォールを使用する場合は、このポートを設定する必要があります。

これらの設定は、デフォルトでは `magicxpi_setenv.bat` ファイル(<Magic xpi インストール先>\Runtime\Gigaspaces\bin フォルダに存在)上で無効になっています。設定はそれぞれ有効にすることができます。例えば、**DISCOVERY_PORT** と **WEBSTER_PORT** はデフォルトのままとし、**LRMI_PORT_RANGE** 設定のみ有効にすることができます。

注: 検出ポートを GigaSpaces のデフォルトポート(XAP9.1 では 4174)以外のポートに設定した場合、定義したポートを使用するロケータの値も変更する必要があります。

最も一般的なシナリオは、ファイアウォールの背後に全ての GigaSpaces エンティティを配置し、DMZ 内からのアクセスは Web リクエストまたは Web サービスリクエストのみとすることです。ファイアウォールの設定は次のようにする必要があります。:

1. 全ての GigaSpaces ノードはユニキャストディスカバリモードで設定し、マルチキャストを無効にする必要があります。
2. **DISCOVERY_PORT**、**LRMI_PORT_RANGE**、**WEBSTER_PORT** は固定値で設定する必要があります。
3. 受信トラフィックのファイアウォール規則は、固定値で定義されたリスナー ポートごとに TCP ポート (受信と送信の両方) をオープンする必要があります。
4. ポート番号は 1024 – 65536 の範囲で設定する必要があります。未使用の空きポートのみ使用することができます。推奨されるポート番号は、IANA で未使用の **7100** 番以上のポートです (7102-7120, 7130-7160, 7167-7173, 7175-7199, 7228-7271, 7282-7299, 7366-7390..., 47558-47623, 47625-47805, 47809-47999, 48004-48127, 48620-49150)。

プロジェクトの実装

プロジェクトフォルダの位置

Magic xpi のプロジェクトは全てのアプリケーションサーバからアクセスできる共有フォルダに配置することをお勧めします。共有フォルダは潜在的にシングルポイント障害(SPOF)の可能性がありますが、今日のストレージシステムにおいてこの種のリソース場合、通常自らの高い冗長性を備えています。

プロジェクトの配置

プロジェクトを配置するには、開発/ステージング環境から共有フォルダの各場所に各プロジェクトのフォルダをコピーする必要があります。データベースやエンタープライズシステム等の外部リソースにアクセスするようなプロジェクトの場合は、**設定** ダイアログボックスのリソースやサービスを編集して、サーバの接続情報や資格情報を設定する必要があります。

プロジェクト自動起動の設定

`projectsStartup.xml` というファイルを作成することで、Magic xpi 4.6 GSA サービスが Magic Space を起動するときにプロジェクトと Magic xpi サーバを自動起動するように設定することができます。`projectsStartup.xml` ファイルの構造は、プロジェクト毎に作成される `start.xml` の構造と同じです。

1. `config` フォルダにある `start.xml` ファイルあるいは `projectsStartup.xml.example` ファイルをコピーします。
2. 必要な変更を行い、`projectsStartup.xml` として `config` フォルダに保存します。

グリッドの起動シーケンスは以下の通りです。:

1. グリッドサービスエージェント(GSA)は、各アプリケーションサーバのサービスとして起動され、このアプリケーションサーバはグリッドの一部となります。GSA は、コアグリッドインフラストラクチャを開始し、全てのアプリケーションサーバを同時に接続します。
2. 正常に起動した GSA の 1 つが、Magic Space を起動します。この Magic Space には、グリッド上で共有されるすべての Magic 関連オブジェクトが含まれています。
3. また GSA は `projectsStartup.xml` 情報を Magic Space 内に取り込み、各サーバ用の **ServerData** オブジェクトを作成します。これにより、プロジェクトを自動的に開始することができます。
4. GSA は Space から Magic xpi エンジンを実行するコマンドを受け取ります。これは、実行および自動起動プロセスの一部とし Space に作成された **ServerData** オブジェクトのリストに基づいています。

5. 配置されたプロジェクト毎に、最初に起動する Magic xpi サーバが Magic Space 内に取り込まれたプロジェクトのメタデータオブジェクトを初期化し、プロジェクトを開始できるようにします。

ProjectsStartup.xml ファイルの例

下記は、Demo1 というサンプルプロジェクトのスタートアップコンフィグレーションを定義した projectsStartup.xml ファイルの例です。

```
<Magicxpi_Startup>
  <Projects>
    <Project Name="Demo1" ProjectsDirPath="//10.1.1.6/projects">
      <Servers>
        <Server host="APP1" alternateHosts="APP2,APP3">
          <NumberOfWorkers>5</NumberOfWorkers>
          <NumberOfInstances>1</NumberOfInstances>
          <Triggers load="true"/>
          <Scheduler load="true"/>
          <AutoStart load="true"/>
        </Server>
        <Server host="APP2" alternateHosts="APP1,APP3">
          <NumberOfWorkers>3</NumberOfWorkers>
          <NumberOfInstances>1</NumberOfInstances>
          <Triggers load="false"/>
          <Scheduler load="false"/>
          <AutoStart load="false"/>
        </Server>
        <Server host="APP3" alternateHosts="APP1,APP2">
          <NumberOfWorkers>5</NumberOfWorkers>
          <NumberOfInstances>1</NumberOfInstances>
          <Triggers load="true"/>
          <Scheduler load="false"/>
          <AutoStart load="false"/>
        </Server>
      </Servers>
    </Project>
  </Projects>
</Magicxpi_Startup>
```

このサンプルファイルでは以下のような設定を行っています。:

- プロジェクト共有フォルダは \\10.1.1.6\projects 。
- プロジェクトフォルダは \\10.1.1.6\projects\Demo1 。
- 各アプリケーションサーバは 1 サーバプロセス/インスタンスで起動。
- APP1 は 5 ワーカーを起動し、全てのトリガ、自動起動、スケジューラを実行する。
- APP2 は 3 ワーカーを起動し、トリガ、自動起動、スケジューラを全く使用しない。
- APP3 は 5 ワーカーを起動し、トリガ起動は行おうが自動起動、スケジューラを使用しない。
- 各アプリケーションサーバは、他の 2 つのアプリケーションサーバを指すように **alternateHosts** プロパティを設定します。これはあるホストが使用不能になった場合、他のホストで代替処理を行うための設定です。

プロジェクト共有フォルダは、属性 **ProjectsDirPath** によって設定されます。すべてのアプリケーションサーバに同じフォルダを定義するか、11 ページで説明したように各サーバのローカルプロジェクトフォルダをアクセスすることもできます。各トリガが実行されるサーバを定義することも、エンジン間でトリガ分割することまで来ます。

Magic モニタ用に GigaSpaces を設定

Magic Monitor は、<Magic xpi インストール先>\Runtime\RTView\magicmonitor フォルダ内の **runwebmonitor.bat** ファイルで定義された設定を使用して Space を検索します。

コマンドラインからのプロジェクトの起動 (オプション)

プロジェクトはコマンドラインから開始することができます。コマンドラインからプロジェクトを開始するには:

1. プロジェクトフォルダの **Start** リンクに移動します。
2. 右クリックし、**プロパティ オプション** を選択します。
3. リンク先欄は以下のように設定されています。:
`C:\Magicxpi4.6\Runtime\MgxpCmdl.bat start-servers -startup-config-file "C:\Users\administrator\Documents\magic\projects\Project3\Project3\start.xml" -space-name "MAGIC_SPACE" -group "Magicxpi-4.6.0" -locators ""`
4. このパスは、プロジェクト配下の特定の **start.xml** ファイルを指します。複数のプロジェクトをロードできる設定を行った **start.xml** ファイルをロードするようにパスを変更します。
5. **OK** をクリックします。

Web リクエストと SOAP サーバの実装

[はじめにの章](#) にある図の構成にあるように Web サーバ及び SOAP サーバは、DMZ ファイアーウォール内のフロントエンドサーバに独立して配置されます。このサーバには通常の Magic xpi とは異なるセットアップが必要です。

前提条件

- Web リクエストを使用する場合、IIS7 が必要です。
- Java Web リクエストを使用する場合、Apache Tomcat 7 以降が必要です。

インストール

DMZにある Web サービスサーバ上で、Magic xpi を以下の設定でインストールする必要があります。:

1. データベース設定で、後でデータベースをインストールするよう選択します。
2. **Systinet Web Service Framework** チェックボックスをオンにします(プロジェクトで SOAP アクセスを行う場合)。
3. **グリッドサービスエージェント(GSA)をサービスとしてインストール** チェックボックスをオフにします。
4. フロントエンドのサーバが **Space** に接続できるようにするために、他のアプリケーションサーバのロケータと同じ値を設定します。

これらの設定により、Web サービスリクエストと Systinet SOAP サーバは Space と通信することができます。

Magic xpi 4.6 GSA サービスはこのマシンでは実行されず、外部リクエストのホスト役を果たすのみで、グリッドコンポーネントや Magic xpi エンジンを実行されません。

Java Web リクエスト

Java Web リクエストを使用するには以下の PDF ファイルを参照してください。:

ファイル名 : **Magic xpi - Java-Based Installation Instructions.pdf**

フォルダ : <Magic xpi インストール先>\Runtime\Support\JavaWebRequester\Tomcat

スペースクラスタリング SLA

スペースクラスタリング SLA はスペースパーティションの数、パーティションバックアップの数、それらが利用可能なグリッドコンテナ(GSCs)に展開する方法を定義します。

このドキュメントで説明されている 3 つのノードを持つクラスタでは、最低でも 2 つのパーティションと 1 つのバックアップを持つスペースを実装することをお勧めします。さらに、SLA はパーティションとそのバックアップが決して同じホストの下に存在しないように設定する必要があります。MAGIC_SPACE SLA は以下のようになります：

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:os-
sla="http://www.openspaces.org/schema/sla"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.openspaces.org/schema/sla
http://www.openspaces.org/schema/8.0/sla/openspaces-sla.xsd">
    <os-sla:sla cluster-schema="partitioned-sync2backup" number-of-
instances="2" number-of-backups="1" max-instances-per-machine="1">
        </os-sla:sla>
</beans>
```



このセクションでは、SLA に関連するさまざまな設定について詳しく説明します。：

スペースクラスタリングは、SLA 定義によって管理されます。つまりグリッドは Space が実装されるときに定義されたクラスタリングを常に維持しようとします。

クラスタリング SLA は config フォルダ内の 2 つの SLA ファイルで定義されます。：

- **MAGIC_INFO** スペースの場合、クラスタリングは **magicinfo_sla.xml** ファイルで定義されます。
- **MAGIC_SPACE** スペースの場合、クラスタリングは **magicxpi_sla.xml** ファイルで定義されます。

デフォルトでは SLA ファイルにはそれぞれ 1 つのバックアップを含む 2 つのパーティション(合計 4 つ)を定義します。また、プライマリパーティションとそのバックアップパーティションを同じプロセスで実行できないという制限があります。

グリッドが SLA 定義に準拠するためには、十分な GSC を定義する必要があります。

上記のデフォルト設定では、プライマリパーティションは同じプロセスでバックアップを実行できないため、正常な Space 実装のために少なくとも 2 つの GSC が稼働している必要があります。

最も一般的な SLA 設定は以下の通り:

1. **cluster-schema** – これは常に **partitioned-sync2backup** に設定する必要があります。これによりデータをパーティションに入れることができ、各パーティションに同期されたバックアップを持たせることができます。
2. **number-of-instances** – Magic processing unit のインスタンスを意味し、ロードされる必要なスペースパーティションの数。デフォルトは **1** です。メモリに大量のデータがある場合は、この数を増やす必要があります。
3. **number-of-backups** – 各プライマリパーティションのバックアップパーティションの数。開発時には、バックアップが不要であると判断し、この値を **0** に設定できます。 **number-of-instances="2"** かつ **number-of-backups="1"** と設定した時、Magic processing unit として 4 つの 4 つのインスタンスが存在します。
4. **max-instances-per-vm** – この値を **1** に設定すると、プライマリパーティションとそのバックアップを同じ JVM (GSC) にプロビジョニングできません。同じ JVM(GSC) に実装される同じパーティションのインスタンス数、つまり同じプロセスのインスタンス数。デフォルトの **max-instances-per-vm="1"** のままにした場合、同じパーティションのプライマリインスタンスとバックアップインスタンスは同じ GSC に実装されません。
5. **max-instances-per-machine** – この値を **1** に設定すると、プライマリパーティションとそのバックアップを同じマシンにプロビジョニングできません。この値を **1** に設定すると、最低 3 台のマシンを含むクラスタリングに制限されます。いずれかのマシンに障害が発生すると、失われたパーティションは 3 台目のマシンに移動します。また、2 台のマシンのクラスタリングで使用することもできますが、2 台目のマシンがバックアップされて実行されるまで、バックアップのないプライマリパーティションを持つリスクがあります。

SLA の設定例:

1. 1 パーティションに対して 2 つのバックアップパーティションを設定し、かつプライマリパーティションとバックアップパーティションを別々の GSC に設定する場合、**magicxpi_sla.xml** ファイルは次のように設定します。:

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

上記の例では、1 台のマシンに少なくとも 3 つのコンテナが必要です。各コンテナは 1 つのパーティションを保持します。

注: 2 つのバックアップを使用することはお勧めしません。ここでは必要な GSC 数の計算方法を示します。

2. 2 つのパーティションに対してそれぞれ 1 つのバックアップパーティションを設定し、プライマリとバックアップパーティションを別々の GSC に設定する場合、**magicxpi_sla.xml** ファイルは次のように設定します。:

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

上記の例では、1 台のマシンに少なくとも 2 つのコンテナが必要です。各コンテナは 2 つのパーティションを保持します。

3. 2 つのパーティションに対して一つのバックアップパーティションを設定し、プライマリとバックアップパーティションを別々のマシンに設定する場合、**magicxpi_sla.xml** ファイルは次のように設定します。:


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

上記の例は少なくともマシンが 2 台あり、各マシンに少なくともコンテナが 1 つ必要です。各マシンでは、コンテナはパーティションを 2 つ保持します。マシン 2 台構成のクラスタの場合、1 台で障害が発生すると、スペースの実装は不完全な状態となり、マシンが再起動するまではバックアップなしのパーティションはバックアップパーティションを失った状態になります。

*** `max-instances-per-machine = "1"` と設定する場合、クラスタは最少構成でも 3 台のマシンである必要があります。その場合、もしあるマシンに障害が発生すると、失われたパーティションは 3 台目のマシンに移行します。

*** GSC の数は、`Gigaspaces\bin` フォルダの下にある `magicxpi-gs-agent.bat` ファイルで定義されています。そのファイル中の `call gs-agent.bat` で始まるコマンドにパラメータ `gsa.gsc` がありますが、これで必要なパーティション当たりの GSC の数を設定します。

スペースの実装

Magic xpi には 2 つのスペース(Space)と 1 つの処理ユニット(Processing Unit)が存在します。:

- **MAGIC_SPACE** – このスペースはプロジェクトメタデータリカバリーとメッセージングの管理を担当します。
- **MAGIC_INFO** – このスペースには、アクティビティログと、モニタ用の統計および ODS データが格納されます。MAGIC_INFO スペースはスペースにない ODS レコードが要求された場合はデータベースから読み取ることができます。
- **MGMirror** – この処理ユニット(PU)は、アクティビティログおよび ODS データをデータベースに書き込む操作の管理を担当します。

Magic xpi OS サービスは、`<Magic xpi インストール先>\Runtime\Gigaspaces\config\gsa` フォルダから以下を実行し、グリッドサービスエージェント(GSA) を起動します:

- Magic Space の実装を担当する `mgdeploy` アプリケーション。
- MAGIC_INFO スペースの実装を担当する `mginfo` アプリケーション。
- MGMirror 処理ユニットの実装を担当する `mgmirror` アプリケーション。

実装のプロセスでは、`magicxpi_sla.xml` ファイルで定義されている SLA 設定が使用されます ([スペースクラスタリング SLA](#) の項を参照)。

実装のプロセスは 1 つのサーバの障害が Magic Space のオペレーションに影響を与えず、データの損失が引き起こされないように、パーティションを利用可能なコンテナ(GSC)に広げようとしています。

このプロビジョニングプロセスは自動的に行われますが、一旦完了すると、自動的に再構成は行われません。

Magic Space の実装プロセス中に 1 台のマシンしか稼動しておらず、1 台のマシン(max-instances-per-machine)に関する SLA 定義に制限がない場合、このマシンは全てのパーティションを保持します。実装が完了した後に他のマシンで起動するコンテナは Magic Space パーティションを保持しません。現在、Magic Space を実行している 1 台のマシンは、単一障害点(SPOF)とみなされます。

複数のマシンがグリッドの一部である場合、Magic Space がいつ実装されるかを制御したいと思うでしょう。グリッドサービスエージェント(GSA)がロードされ、マシンがグリッドの一部になると、グリッドに Magic Space が既に実装されている場合、そのマシンは Magic Space の一部をホストしません。

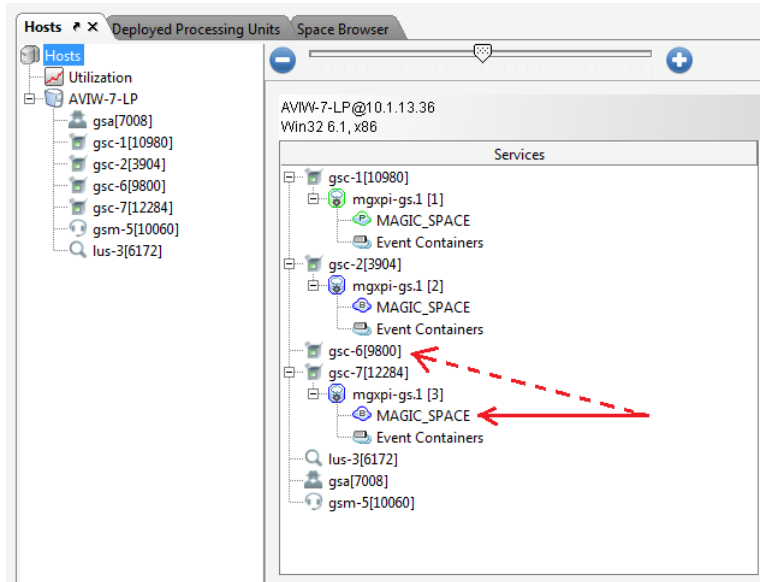
1 つのマシンがすべてのパーティションを保持しているときに複数のマシンにパーティションを分散するには、次のオプションがあります。:

1. SLA で **max-instances-per-machine** 制約を使用することができます。このメソッドは、少なくとも 3 台のマシンによるクラスター構成である必要があり、グリッド内の少なくとも 2 台のマシンで Space パーティションを確実に実行しなければなりません。
 - a. **magicxpi_sla.xml** ファイルで、[スペースクラスタリング SLA](#) セクションで説明されているように、**max-instances-per-machine = "1"** エントリを定義します。
 - b. 自動実装プロセスが開始されると、少なくとも 2 台のマシンが Magic Space パーティションをホストするまで、実装は完了しません。
2. Magic xpi は、同じホスト上で実行されているプライマリとバックアップの両方のパーティションの単一障害点を自動的に監視し、再調整することができます。パーティションのインスタンスの再バランスが必要かどうかを定期的に確認します。このメカニズムは、**mgdeploy.xml** ファイル(<Magic xpi インストール先>\Runtime\Gigaspaces\config\gsa フォルダ内に存在)で定義されている次の 2 つのプロパティにより制御されます:
 - **rebalance-partitions** : このプロパティが **true** (デフォルト)に設定されている場合、または存在しない場合は、再調整メカニズムがアクティブになります。
 - **rebalance-interval** : このプロパティは、再バランスチェックの間隔を定義します。プロパティが存在しない場合、デフォルトは **5** 分です。

これらのプロパティは **mgdeploy.xml** ファイルに次のように定義されています:

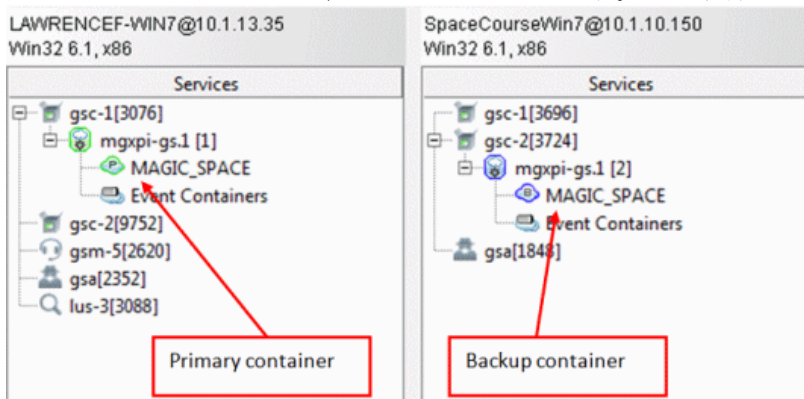
```
<argument>-rebalance-partitions</argument>
<argument>true</argument>
<argument>-rebalance-interval</argument>
<argument>5</argument>
```

- GigaSpaces UI からパーティションを手動で再配置できます。 これを行うには、Gigaspace UI Hosts タブを開き、左側の階層ツリーの最上部にある **Hosts** エントリを選択します。 Gigaspace UI 画面の右側の **Services** ペインに、コンテナとパーティションのツリーが表示されます。 次の図に示すように、パーティション (プライマリまたはバックアップ) を選択して別のコンテナにドラッグできるようになります。



- バックアップ GSC を再起動すると、GigaSpaces はグリッドをプロビジョニングします。 以下の手順で行います。 :
 - バックアップパーティションの GSC ノードにカーソルをパルクします。
 - コンテキストメニューから、**Restart**(再起動)を選択します。

GigaSpaces は、下図に示すように、バックアップコンテナを 2 番目のコンピュータに実装しようとします。 これにより、アプリケーションの冗長性が確保されます。 2 台目のマシンが利用できない場合、GigaSpaces は現在のマシン上にバックアップパーティションを作成します。 2 台目のマシンが再び利用可能になっても、GigaSpaces は 2 台目のコンピュータのバックアップを自動的に再実装しないことがあります。 この場合は手動で再実装を行います。



グリッドコンポーネント - メモリの割当

各種 GigaSpaces エンティティのメモリ割当は `magicxpi-gs-agent.bat` バッチファイルで定義します。このファイルは <Magic xpi インストール先>\Runtime\Gigaspaces\bin フォルダ配下に存在します。

このバッチファイル内には、`gigaSpaces Memory related settings` セクションがあります。

GSA、GSM、LUS エンティティは、メモリの占有スペースが非常に小さいため、これらの設定をそのまま使用できません。GSC は Space パーティションを実行し、プロジェクト内の全データを保持するコンテナです。GSC に関するメモリ関連の問題が発生した場合は、この値を 1024MB 以上に変更することを検討する必要があります。

GigaSpaces サービス設定

`magicxpi-gs-agent.bat` ファイルには、次の行が定義されています。:

```
call gs-agent.bat gsa.gsc 2 gsa.global.gsm 2 gsa.lus 1 gsa.global.lus 0 gsa.mgmirror 1  
gsa.mgdeploy 1 gsa.mginfo 1
```

このファイルには以下のエントリがあります。:

- **gsa.gsc:** GSC を実装する数。この数は必要なパーティション数と一致させる必要があります
- **gsa.global.gsm:** グリッド上でグローバルに実装/管理される GSM の数。
- **gsa.lus:** ローカルで開始する LUS の数。このエントリの値は、マシンが LUS を実行するかどうかによって異なります。
- **gsa.global.lus:** グリッド上でグローバルに展開/管理される LUS の数。
- **gsa.mgmirror:** 実装する mirror の数。
- **gsa.mgdeploy:** 実装する MAGIC_SPACE Space の数。
- **gsa.mginfo:** 実装する MAGIC_INFO Space の数。

GigaSpaces 設定に関する詳細は、以下をご覧ください。:

<http://docs.gigaspaces.com/xap110adm/moving-into-production-checklist.html>

ライセンス

Magic xpi 4.6 のライセンスメカニズムは、基本的にフローティングライセンスの形態をとっていますが、各プロジェクトに最少ライセンススレッド数を確保するオプションがあります。



各プロジェクトのライセンススレッドの確保

複数プロジェクトを実行する時、Web サービスや HTTP リクエストのように頻繁なアクセスを実行するプロジェクトでは、全てのスレッドを消費してしまう可能性があります。そのような場合、他の重要なプロジェクトを継続的に実行するためには、そのプロジェクトに対し、最少数のライセンススレッドを確保するよう考慮する必要があります。具体的な設定方法は、Magic xpi ヘルプの**ライセンススレッドの予約**を参照してください。

ホストロックライセンス

Magic Software Enterprises および Magic Software Japan によって提供される全てのライセンスは、ホスト毎に提供されます。一度アクティベートすると、アクティベートしたそのマシン上でそのライセンスを使用しなければなりません。Magic xpi 4.6 のクラスタ環境では、クラスタ全体に対して1つのホストロックライセンスが必要になります。すべての Magic xpi インストールは、同じライセンスファイルを指すように設定する必要があります(共有フォルダに配置することができます)。全ての Magic xpi サーバは、別のホストで動作していても正常に起動しますが、ロックされたホストのサーバが Magic xpi エンジンを実行するまでフローは実行されません。ホストロックサーバ上で実行される Magic xpi エンジンのみが、スペースに対してスレッドのカウンタをアップデートすることができます。

ホストロックサーバ上のマジックエンジンは、継続して実行する必要はありません。一度 Space に対してスレッド数をセットすれば、ライセンスを管理する必要はありません。従って、グリッド上の他の Magic xpi エンジンと同様、開始/停止を行うことができます。

Magic Software Enterprises について

Magic Software Enterprises (NASDAQ: MGIC) は世界中の顧客とパートナーが、様々な側面でエンタープライズロジック/データによるユーザ体験をできるよう、よりよい技術をご提供します。

30年に及ぶ経験とワールドワイドでの数百万のインストール実績、IBM, Microsoft, Oracle, Salesforce.com, SAP 等のグローバル IT リーダーとの連携によって、お客様が新しい技術をシームレスに導入でき、ビジネスの機会が最大限に生かせるよう努力します。

詳細については www.magicsoftware.com をご覧ください。



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, 2017

