

リッチクライアント
アプリケーションの実行
Magic xpa



OUTPERFORM THE FUTURE™

本マニュアルに記載の内容は、将来予告なしに変更することがあります。これらの情報について MSE (Magic Software Enterprises Ltd.) および MSJ (Magic Software Japan K.K.) は、いかなる責任も負いません。

本マニュアルの内容につきましては、万全を期して作成していますが、万一誤りや不正確な記述があったとしても、MSE および MSJ はいかなる責任、債務も負いません。

MSE および MSJ は、この製品の商業価値や特定の用途に対する適合性の保証を含め、この製品に関する明示的、あるいは黙示的な保証は一切していません。

本マニュアルに記載のソフトウェアは、製品の使用許諾契約書に記載の条件に同意をされたライセンス所有者に対してのみ供給されるものです。同ライセンスの許可する条件のもとでのみ、使用または複製することが許されます。

当該ライセンスが特に許可している場合を除いては、いかなる媒体へも複製することはできません。ライセンス所有者自身の個人使用目的で行う場合を除き、MSE または MSJ の書面による事前の許可なしでは、いかなる条件下でも、本マニュアルのいかなる部分も、電子的、機械的、撮影、録音、その他のいかなる手段によっても、コピー、検索システムへの記憶、電送を行うことはできません。

サードパーティ各社商標の引用は、MSE および MSJ の製品に対するコンパチビリティに関しての情報提供のみを目的としてなされるものです。

本マニュアルにおいて、説明のためにサンプルとして引用されている会社名、製品名、住所、人物は、特に断り書きのないかぎり、すべて架空のものであり、実在のものについて言及するものではありません。

Magic は Magic Software Japan K.K. の登録商標です。

Magic xpa は Magic Software Enterprises Ltd. のイスラエルその他の国での商標または登録商標です。

Magic xpa Enterprise Studio、Magic xpa Enterprise Client、Magic xpa Enterprise Server および Magic xpa RIA Server は Magic Software Japan K.K. の商標です。

一般に、会社名、製品名は各社の商標または登録商標です。

MSE および MSJ は、本製品の使用またはその使用によってもたらされる結果に関する保証や告知は一切していません。この製品のもたらす結果およびパフォーマンスに関する危険性は、すべてユーザが責任を負うものとします。

この製品を使用した結果、または使用不可能な結果生じた間接的、偶発的、副次的な損害（営利損失、業務中断、業務情報の損失などの損害も含む）に関し、事前に損害の可能性が勧告されていた場合であっても、MSE および MSJ、その管理者、役員、従業員、代理人は、いかなる場合にも一切責任を負いません。

Copyright 2015 Magic Software Enterprises Ltd. and Magic Software Japan K.K. All rights reserved.

2015年3月31日

概要	1
アプリケーション実行ファイルの作成	2
リッチクライアントインタフェースビルダ	2
製品を実行させる PC (サーバ) の準備	3
Magic エンジンとアプリケーション環境ファイルの設定	3
MRB の環境ファイルの設定	4
ClickOnce で必要な MIME タイプの定義	5
同じサーバで 2 つの RIA アプリケーションを実行する	6
ClickOnce を使用したリッチクライアントアプリケーションのインストールと実行	7
コマンドラインから ClickOnce を使用しているアプリケーションをインストールする	8
既知の問題	8
Web エイリアスの設定	9
独自のエイリアスの定義	10
Web リクエスタの設定	14
Magic の動作環境	14
アプリケーションの起動環境	14
公開用 HTML ファイル	15
RIA クライアントを (ClickOnce を使用せず) 手動でインストールする	16
Magic サーバのバージョンを更新する	17
付録	18
リッチクライアントモジュール	18
モジュールの分散	18

概要

リッチクライアントアプリケーション (RIA) は、インターネットまたはイントラネットを経由してアプリケーションサーバに接続するシン型の .NET クライアントから構成されています。実行環境でこれをサポートする設定が必要になります。

適切に動作させるためには、以下のような必要な設定がいくつかあります。

- アプリケーション実行ファイルの作成
- 製品を実行させる PC (サーバ) の準備
- Web エイリアスの設定 (オプション)
- Web リクエストの設定

RIA を実行するための最適な方法も知っておいてください。

リッチクライアントアーキテクチャについての一般的な説明は、「リッチクライアントモジュール」を参照してください。

アプリケーション実行ファイルの作成

RIA を実行する場合、各クライアント PC で Magic xpa クライアントのインストールを行うことなく Web 上のアプリケーションを実行したいと思うはずです。Windows OS の場合は、Microsoft の ClickOnce 技術を使用して実行されます。ClickOnce は、Microsoft によって開発された技術で、Web から直接 .NET アプリケーションを実行することができ、Web ブラウザーの制約に束縛されずに実行することができます。ClickOnce は、ユーザが Web ページ上でリンクをクリックするだけで .NET アプリケーションをインストールして、実行することができるようになります。この技術はクライアントアプリケーションがインターネットからだけでなく、デスクトップ環境から実行することも可能にし、すばらしい環境を提供します。

ClickOnce を使用する利点の 1 つは、実行権限 (例えば .NET フレームワークや MsHtmlSetup などの必要条件を除く) を必要とすることなくアプリケーションを配備して、アップグレードする方法を提供するということです。

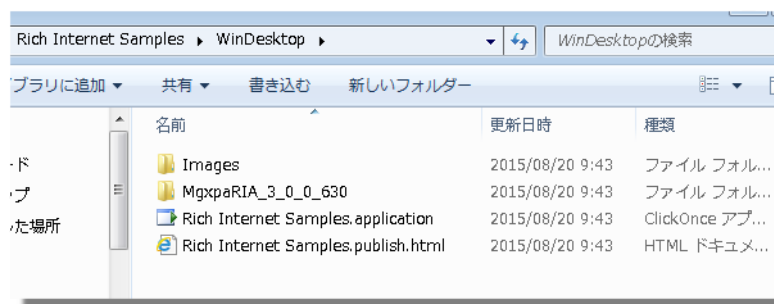
RIA を実行させる場合、外部に公開する必要のあるアプリケーションの要素があると同時に、公開したくない要素もあります。従って、Magic の RIA に対して安全な環境を構築する上で、アプリケーションの要素は以下の二つに分類することができます。

- 内部向けのリソースファイル …… これらは、ECF、基本色定義、フォント定義、ユーザ定義、などの Magic エンジンがアクセスするためだけに必要なローカルファイルです。内部ファイルは、Web サーバによってアクセス可能にしないで、プロジェクトフォルダ内のような ECF が置かれているフォルダ付近に配置してください。
- 外部向けのリソースファイル …… これらは、HTML やマニフェスト、PDF などのクライアント側でアクセスする必要があるファイルです。外部ファイルは、Web サーバを経由してアクセスすることが可能であり、Web サーバのエイリアスや、リッチクライアントインタフェースビルダによって作成されるアプリケーションの公開エイリアスを使用して公開されます。

リッチクライアントインタフェースビルダ

Magic xpa のリッチクライアントインタフェースビルダは、ClickOnce を使用する RIA を配備するために必要な以下のファイルを作成することができます。

- Images フォルダ …… このフォルダには、HTML ファイルで使用されるイメージが含まれています。
- MgxpaRIA_X_X_X_XXX フォルダ …… このフォルダには、RIA クライアントモジュールが含まれています。
- .NET 実行マニフェストファイル …… ClickOnce の実行は、このファイルによって呼び出されます。
- HTML ファイル …… HTML ページは、ユーザがハイパーリンクを通して RIA クライアントをインストールできるようにします。このファイルを編集して、独自の HTML ページとしてコードを使用することができます。この Web ページは、RIA クライアントをインストールするために、一度だけ必要となります。一旦 RIA クライアントがインストールされると、ユーザはデスクトップ上のアイコンを使用して起動することができます。



これらのファイルとフォルダは、PublishedApplications フォルダにプロジェクト名のサブフォルダとして各プロジェクト独自の環境ファイルが保存されます。

リッチクライアント実行ビルダを実行する場合は、必ず .NET マニフェストファイルの証明書とパスワードを定義するようにしてください。Magic xpa のインストールで提供されるテスト用の証明書は、発行元が確認できないため使用しないことを推奨します。通常は、発行元が確認できるような特定機関の証明書を使用してください。

ノート

Magic xpa エンジンをアップグレードした場合は、マニフェストファイルを常に再形成しなければなりません。

「RIA クライアント (ClickOnce を使用せず) を手動でインストールする」も参照してください。

製品を実行させる PC（サーバ）の準備

アプリケーションをサーバ上で実行させるために必要な手順は、以下の通りです。

1. IIS をサーバ PC 上にインストールされていることを確認してください。
2. Magic xpa を PC にインストールします。
3. 開発 PC からサーバ PC に PublishedApplications フォルダをコピーしてください。このフォルダには、アプリケーションをクライアント上にインストールするために必要なファイルが含まれています。
4. アプリケーションが使用する他のすべてのフォルダ（例えば、画像ファイルなど）もサーバ PC にコピーしてください。
5. Magic エンジンとアプリケーション環境ファイル（Magic.ini）を設定してください。
6. ミドルウェアの設定を行ってください。
 - MRB（Magic xpa Studio で利用する場合）
 - a. Microsoft® の .NET フレームワーク CLR(Common Language Runtime) の 4.0 以上がインストールされていることを確認してください。
 - b. MRB の環境ファイル（Mgrb.ini）を設定してください。
7. ClickOnce の実行に必要な MIME タイプを定義してください。
8. 実行環境が存在するディレクトリに読込権が設定されていることを確認してください。
9. 実行環境が UNC システムをサポートしていない場合、すべてのクライアント環境で実行時のディレクトリ構成が同じ設定になっていることを確認してください。
10. MgxpRuntime.exe ファイルを開始するためのショートカットが実行時に同じディレクトリを示していることを確認してください。

Magic エンジンとアプリケーション環境ファイルの設定

カスタマイズされた Magic.ini ファイルを実行環境にコピーした場合、プロジェクトの開発環境の設定内容を変更する必要があります。環境設定は、Magic xpa Studio で行うことができますが、Studio を持っていない場合もあります。手動で Magic.ini ファイルを修正することでこれらの環境設定を定義することもできます。

以下は、確認し変更する必要がある実行環境です。リッチクライアントインタフェースビルダを使用する場合、これらの設定のいくつかを指定できるようになっているため、すでに設定されているかもしれません。

1. エイリアスを更新してください。変更方法は、「独自のエイリアスの定義」に説明があります。
2. Magic エンジンの実行モードをバックグラウンド（Magic.ini ファイルの DeploymentMode=B）に設定してください。これによって、サーバが確実にバックグラウンドで動作します。アプリケーションがバックグラウンドで動作する場合、ユーザ操作するウィンドウが表示されない点に注意してください。
3. [アプリケーションサーバとして動作]（Magic.ini ファイルの ActivateRequestsServer）を「Yes」に設定します。この特性は、Magic エンジンがアプリケーションサーバとして実行するようになり、MRB を経由してリクエストを受け付けます。
4. [HTTP リクエスタ]（Magic.ini ファイルの InternetDispatcherPath）に MGrqispi.dll か MGrqcgi.exe のパスを指定します。この特性は HTTP リクエスタとなり、インストール処理中に設定されます。ここには、インターネットリクエスタの名前と関連する Web パスを定義します。

ノート

Microsoft の IIS Web サーバを使用する場合は、ISAPI リクエスタ、その他の Web サーバ（例えば Apache）を使用する場合は CGI リクエスタを使用することを推奨します。

5. [開始アプリケーション] (Magic.ini ファイルの StartApplication) にアプリケーションのキャビネットファイルを以下の画像のように定義して、実行するアプリケーションを定義します。

```

4 InputPassword = N
5 NumberOfRecentProjects = 4
6 DefaultProject =
7 StartApplication = D:\Program Files\Magic\xpstudio\Studio 3.0\SampleProjects\Rich Internet Samples\Rich Internet Samples.ecf
8 DeploymentMode = B
9 Terminal = 0
0 DefaultDatabase = Default Database
1 TempDatabase = Memory
    
```

6. プロジェクトの開発時に、論理名を使用している場合、実行環境の Magic.ini ファイルに [MAGIC_LOGICAL_NAMES] セクションをコピーして、パスが正しいことを確認してください。
7. このアプリケーションで使用しているデータベース環境をコピーして、パスが実行時のデータベースを示していることを確認してください。この設定は、Magic.ini file の [MAGIC_DATABASES] セクションにあります。
8. 総じて、すべてのパスが正しく設定されることを確認してください。

```

0
1 [MAGIC_DATABASES]
2 Default Database = DBMS,1, , , , , MagicRecordLock, ChangeFileInToolkit+
3 , CheckDefinition, CheckKey, FileLocks, , , NoCheckExist, , ,
4 Memory = DBMS,22, , , , , NoMagicRecordLock, ChangeFileInToolkit, +
5 NoCheckDefinition, NoCheckKey, NoFileLocks, , , NoCheckExist, , ,
6 Local = DBMS, 11, , local.sqlite, , , , NoMagicRecordLock, DontChangeFile+
7 eInToolkit, CheckDefinition, NoCheckKey, NoFileLocks, , , CheckExist, 0, +
8 , NoAS400SrvrSort,
9 Database = DBMS, 10, , sqlite, , , , NoMagicRecordLock, ChangeFileInTo+
0 kit, CheckDefinition, NoCheckKey, NoFileLocks, , , CheckExist, 0, , +
1 NoAS400SrvrSort,
2 Default XML Memory Database = DBMS,22, , , , , NoMagicRecordLock, +
3 ChangeFileInToolkit, NoCheckDefinition, NoCheckKey, NoFileLocks, FILE_CHUNK+
4 _SIZE=50, , NoCheckExist, , ,
5 Default XML Database = XML,0, , , , , NoMagicRecordLock, ChangeFileInTo+
6 olkit, CheckDefinition, CheckKey, FileLocks, , , NoCheckExist, 0, , +
7 NoAS400SrvrSort, , Default XML Memory Database
8 RIADEMO=DBMS, 10, , RIADemo.sqlite, , , , NoMagicRecordLock, ChangeFileInt+
9 Toolkit, CheckDefinition, NoCheckKey, NoFileLocks, , , CheckExist, 0, , +
0 NoAS400SrvrSort,
1 OnlineSamples=DBMS, 10, , OnlineSamples.sqlite, , , , NoMagicRecordLock, +
2 ChangeFileInToolkit, CheckDefinition, NoCheckKey, NoFileLocks, , , +
3 CheckExist, 0, , NoAS400SrvrSort,
4 RIASamples=DBMS, 10, , RIASamples.sqlite, , , , NoMagicRecordLock, +
5 ChangeFileInToolkit, CheckDefinition, NoCheckKey, NoFileLocks, , , +
6
    
```

9. ログングを行うとプログラムの実行パフォーマンスを低下させるため、ログングをオフにすることを推奨します。これを行うには、Magic.ini の [MAGIC_LOGGING] セクションを「N」に設定するか Magic xpa Studio で [設定/ログング] で変更してください。

MRB の環境ファイルの設定

自動的に 1 ～複数の Magic エンジン起動するようにするには、MRB の設定ファイル (Mgrb.ini) を設定しなければなりません。これを行うには、[APPLICATIONS_LIST] セクションに以下の構文で定義します。

APPNAME=<コマンド>,<作業フォルダ>,<OS のユーザ名>,<OS のパスワード>,<MRB の初期化時に実行される回数>,<起動されるエンジンの最大数>

<MRB の初期化時に実行される回数>に「1」が設定された場合、定義されたアプリケーションの 1 インスタンスが起動されます。

例 :Rich Internet Samples = MgxpRuntime.exe,,,1,0

Magic.ini ファイルでこれらの環境を定義する代わりに、コマンドラインパラメータ (例えば StartApplication) を指定することができます。

例 :

Rich Internet Samples = MgxpRuntime.exe /DeploymentMode=B /LicenseName=MGRIA /StartApplication = SampleProjects\Rich Internet Samples\Rich Internet Samples.ecf,,,1,0

ノート

このトピックの手順は推奨内容ですが、オプションです。この手順を省略して、手動で実行エンジンを起動することもできます。



ClickOnce で必要な MIME タイプの定義

アプリケーションを実行しているサーバに .NET のフレームワークがインストールされていない場合、サーバ上で以下の MIME タイプの拡張子を定義する必要があります。

ノート

このトピックで説明されている設定は、Magic xpa のインストール処理の中で自動的に行われますが、Windows のバージョンによっては設定されない場合があります。この場合は以下の手順で、手動で行ってください。

- .manifest application/x-ms-application
- .application application/x-ms-application



これを行うには、IIS マネージャーで以下のようにします。

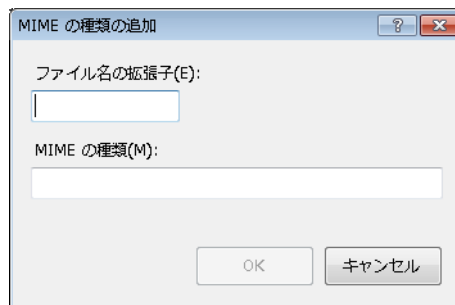
1. [MIMEの種類] のアイコンをクリックします。



- ここに表示されているリストに2つの MIME タイプが存在していない場合、画面上で右クリックし [追加] をクリック選択します。



- [MIME タイプの追加] ダイアログに情報を入力して、[OK] クリックします。



同じサーバで2つのRIAアプリケーションを実行する

複数の RIA アプリケーションを実行する場合、サーバ上で RIACache フォルダを分けることを推奨します。サーバ上でアプリケーション毎に異なる Magic.ini ファイルを読み込ませることで可能になります。アプリケーションの指定した設定に加え、各 .ini ファイルには異なる RIACache パスが定義されていなければなりません。これによって、キャッシュファイルが異なるフォルダに保存されることになります。

ClickOnce を使用したリッチクライアントアプリケーションのインストールと実行

サーバを設定しアプリケーションを実行すると、アプリケーションは HTML ファイルを参照する URL を指定することでアクセスすることができます。リッチクライアントインタフェースビルダの最後の画面で、URL を参照することができます。

デフォルトでは、URL は以下のようになります。

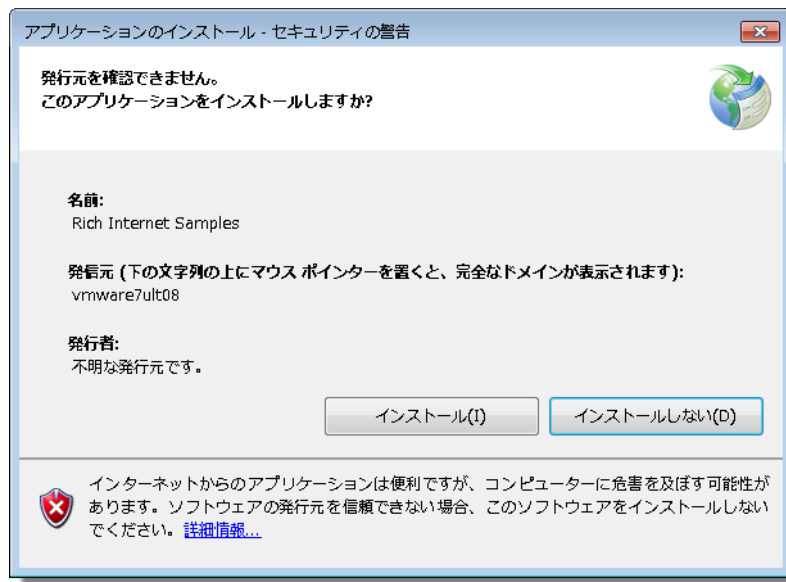
`http://appserver/Magic4xRIAApplications/appname/appname.publish.html`

URL によるアプリケーションの起動は、クライアント側で最初にアプリケーションを実行するときのみ必要です。最初の実行時によって、ClickOnce はスタートメニューにアプリケーションを実行するための新しいエントリを作成します。以降は、アプリケーションを起動するためにこのエントリを使用する必要があります。このため、ClickOnce を実行する場合の Web ブラウザへの依存は、最初だけということになります。

Web ブラウザを使用して初めてアプリケーションにアクセスする場合、以下のような画面が表示されます。



1. 以下の必要条件をクライアント上にインストールしなければなりません。
 - Microsoft® .NET Framework CLR(Common Language Runtime) 4.0 以上 …… .NET フレームワークは、Windows 8 以降では既にインストールされているので、手動でインストールする必要はありません。.NET がまだインストールされていない場合、この画面の .Net Framework ボタンをクリックして、Microsoft 社のダウンロードサイトからインストールパッケージをダウンロードしてインストールできます。
2. 次に、[スタート] ボタンをクリックしてください。以下の画面が表示されます。



3. アプリケーションをインストールするには、[インストール] ボタンをクリックします。

コマンドラインから ClickOnce を使用しているアプリケーションをインストールする

Web ブラウザを使用しないで RIA アプリケーションをインストールしたい場合は、コマンドラインを使用してアプリケーションマニフェストを実行することができます。

例：

`http://appserver/Magic4xRIASamples/appname/appname.manifest`

この方法を使用すると、ClickOnce はクライアントファイルを配備して、さらに自動的にアップグレードします。

既知の問題

- RIA は、Guest アカウントで ClickOnce を使用して配備することはできません。
- ClickOnce は、統合プロキシ認証のみをサポートします。
- ClickOnce アプリケーションは、Microsoft の Internet Explorer または Edge の Web ブラウザからインストールすることができます。
- 他の Web ブラウザ (Google Chrome または Firefox など) …… この Web ブラウザーはダウンロードフォルダ内に実行マニフェストファイルをダウンロードするだけです。セキュリティ制約のため、自動で実行させることはできません。しかし、手動で開始させることができます。
- アプリケーションがオフラインモードで動作するように設計されていて、ClickOnce を使用して配備している場合、発行された Web ページ (HTML ファイル) と実行マニフェスト (アプリケーションファイル) がクライアントで利用できるようにする必要があります。

Web エイリアスの設定

Magic xpa をインストールすると、デフォルトの Web エイリアスが作成されます。しかし場合によっては、独自のエイリアスを定義する方が良い場合もあります。

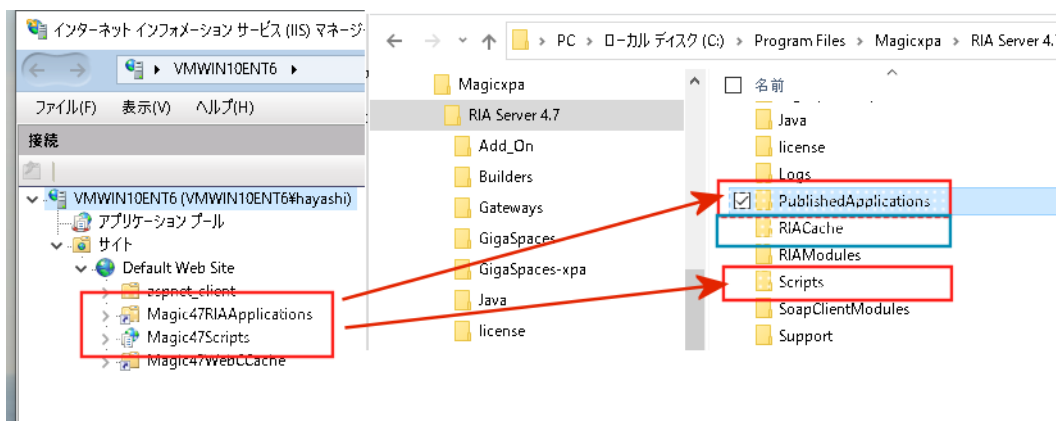
特定のリソース（例えば HTML や PDF ファイル）を公開したい場合は、特に必要です。デフォルトのエイリアスを使用することができますが、エンドユーザが参照するパスには、Magic xpa のエイリアス名が含まれます。例えば、は以下のようにすることでアクセスできます。

`http://myserver/Magic4xRIAApplications/customerreport.pdf`

最初に、Magic xpa によって定義されるデフォルトのエイリアスを見てみましょう。

ノート

Magic xpa 4.x では、インストール時に **MagicxpaAppPool4.x** という名前のアプリケーションプールを追加し、**Scripts** にマッピングされる仮想ディレクトリを作成します。



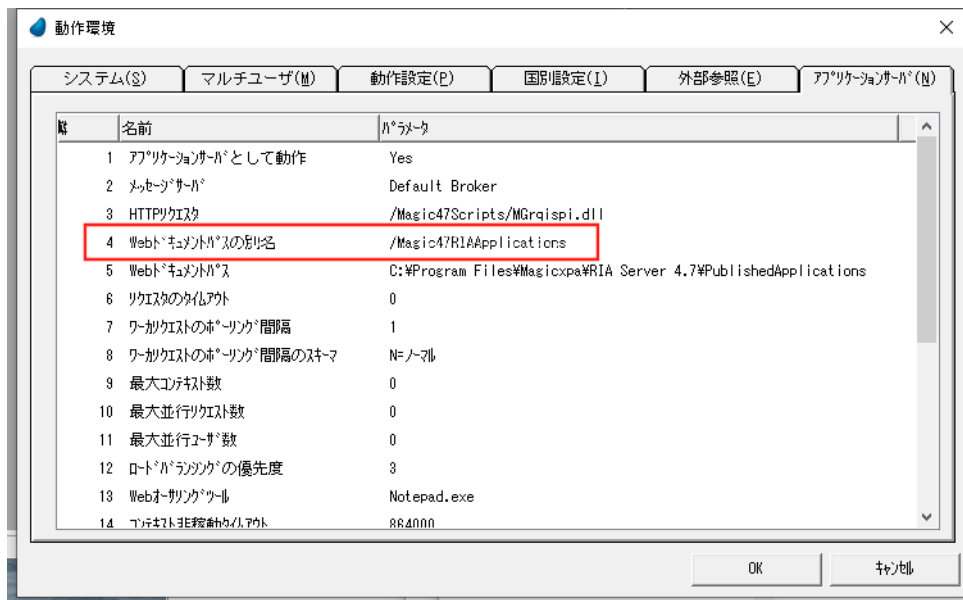
Magic xpa によって設定されるデフォルトのエイリアスは、以下の通りです。

- **Magic4xRIAApplications** …… このエイリアスは **PublishedApplications** ディレクトリを参照しています。これは、Magic xpa のインストールフォルダの一部で、RIA アプリケーションを配備するために必要なファイルも含まれています。そして、RIA モジュールと公開する必要のあるファイル（例えば HTML やマニフェストファイル）が含まれています。
- **Magic4xScripts** …… このエイリアスは **Scripts** ディレクトリを参照しています。これは、Magic xpa のインストールフォルダの一部で、Magic xpa のリクエストファイルや RIA に必要なファイルが含まれています。

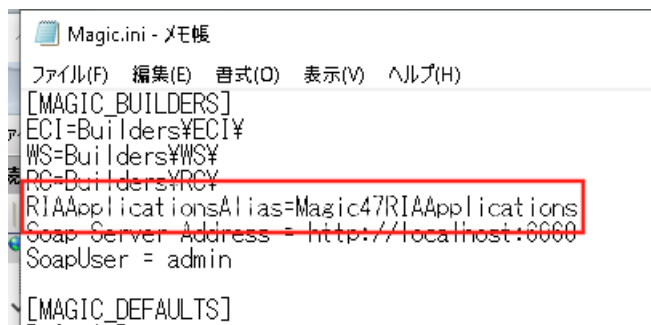
ノート

- **RIACache** と呼ばれるフォルダもあり、ここには、タスクやイメージ、メニューに対応したキャッシュデータが保存されます。このフォルダは、Magic エンジンが内部的に処理するためエイリアスは必要ありません。Magic xpa サーバが RIA のキャッシュファイルを保存するフォルダは、**Magic.ini** ファイルでの **RIACacheFilePath** または [リッチクライアントのキャッシュパス] で設定することができます。サーバ側では、これは **%TEMP%\RIACache** でフォルダに保存されます。
- 各エンジンが独自のキャッシュフォルダを使用するようにすることを推奨します。

使用されるエイリアスは、**Magic.ini** ファイルに設定します。IIS でこれらのエイリアスの名前を変更した場合、**Magic.ini** の設定も変更する必要があります。エイリアスのうちの 1 つは、[動作環境] ダイアログで変更することができます。



Magic.ini ファイルに直接に設定したい場合、WebDocumentAlias エントリを探してください。
アプリケーション用のエイリアスは、Magic.ini の RIAApplicationsAlias エントリにのみ保存されます。

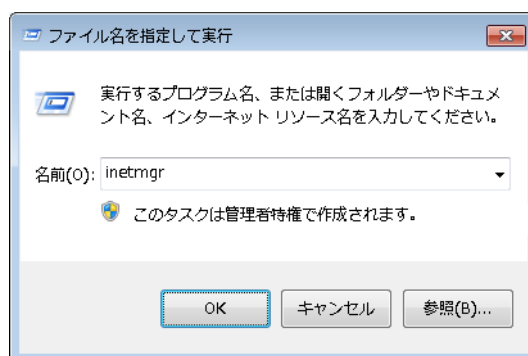


このエイリアスは、アプリケーション用の環境ファイルを作成するリッチクライアントインタフェースビルダによって使用されます。

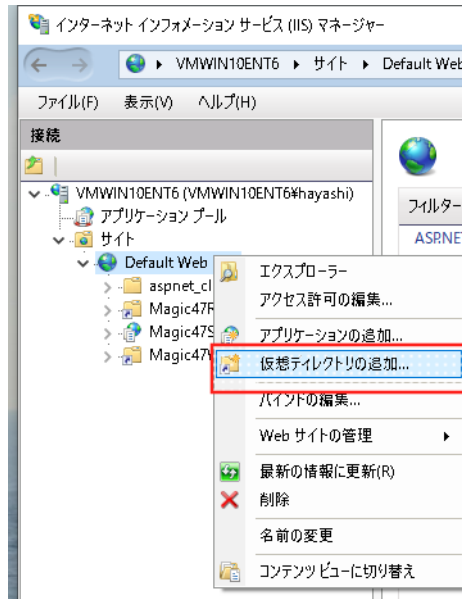
独自のエイリアスの定義

複数のファイルを公開する必要がある場合、新しいエイリアスを定義することを推奨します。この場合、IIS で以下のようにします。

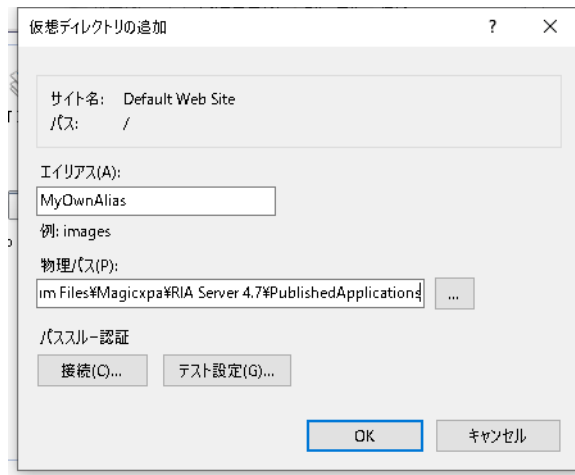
1. IIS マネージャを実行します ([ファイル名を指定して実行] から「inetmgr」を入力するか、[コンピュータ] メニューで右クリックして [管理] を選択し、[サービスとアプリケーション/インターネットインフォメーションサービス (IIS) マネージャ] を選択します)。



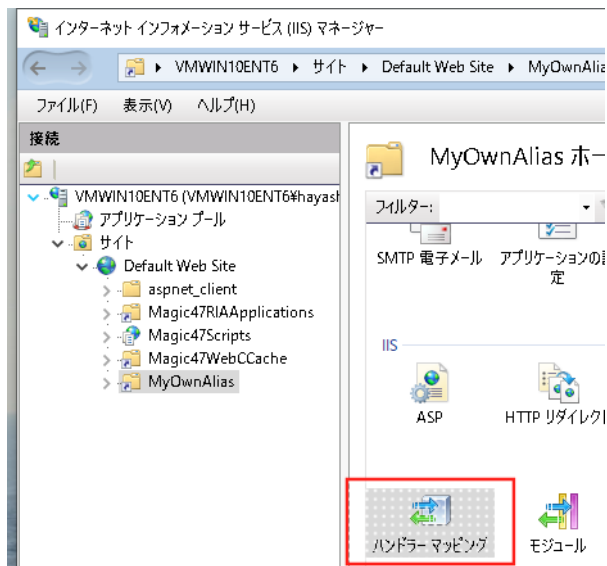
2. IIS マネージャーが開いたら、[Default Web Site] で右クリックして、[仮想ディレクトリの追加] オプションを選択します。



3. [仮想ディレクトリの追加] ダイアログで、エイリアス名と物理パスを入力し、[OK] をクリックします。



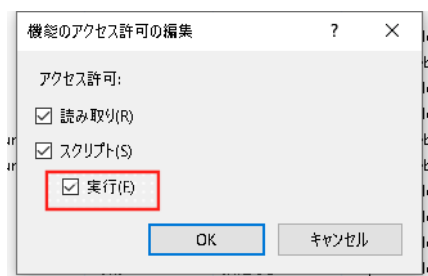
4. 追加したエイリアスに実行権を設定します。
 - a. エイリアス上にパークして、[ハンドラマッピング] アイコンをダブルクリックします。



- b. カーソルを関連するファイル (ISAPIdll か CGI-exe) に置いて、[操作] ペインの [機能のアクセス許可の編集] をクリックします。

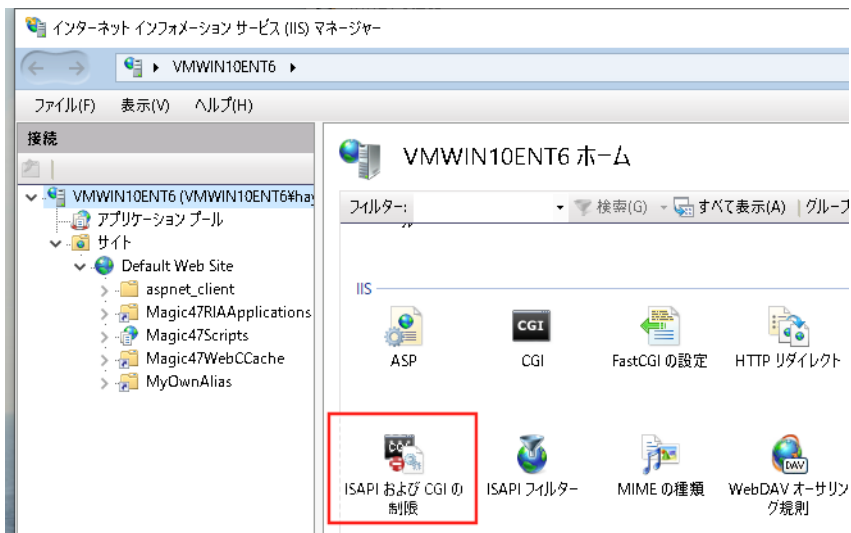


- c. [実行] のチェックボックスをチェックします。

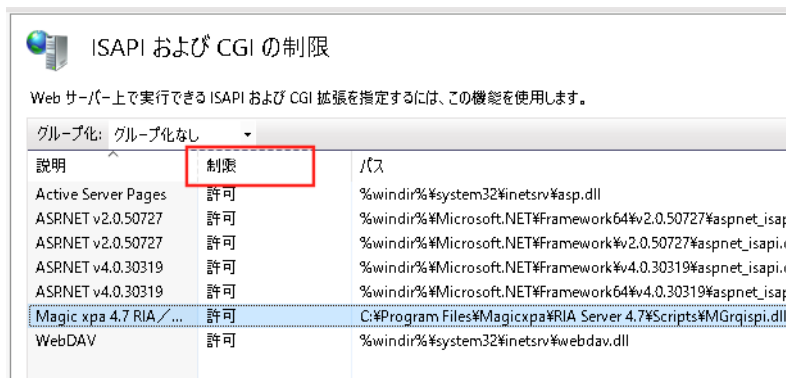


5. MGrqspi.dll と MGrqcgi.exe のファイルに実行権が設定されていることを確認してください。

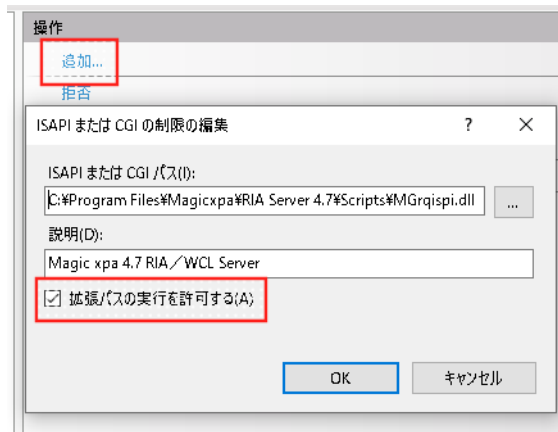
- a. [接続] ペインのルート (ホスト名が表示されています) に移動します。
- b. [ISAPI および CGI の制約] アイコンをダブルクリックしてください。



- c. 実行させたいスクリプトの「制限」カラムを確認してください。「許可」が表示されている場合は、実行権が既に設定されており、手順 10-11 は省略できます。



- d. 「許可」と表示されていない場合は、[操作] ペインの [追加] をクリックします。



- e. MGrqispi.dll/MGrqegi.exe の説明のパスを設定します。
- f. [拡張パスの実行を許可する] のボックスを選択して [OK] をクリックします。
6. これで、外部ファイルを参照するときは常にこのエイリアスを使用することができるようになります。実行環境でも同じ方法でこのエイリアスを定義する必要があります。

Web リクエストの設定

GigaSpaces 用の Web リクエスト (MgWebRequester.dll) は、GigaSpace ミドルウェア経由で Magic xpa サーバにアクセスするためのもので、公開用の HTML や実行特性ファイルに設定する必要があります。

MRB 用のインターネットリクエスト (Apache または CGI) と同じように、ISAPI インターネットリクエスト (MGRQISPI.dll) の代わりに設定することができます。

ノート

Magic xpa RIA Server では、デフォルトのミドルウェアとして **MRB** がインストールされるようになります。**GigaSpaces** を利用する場合は、**64bit** 版でカスタムインストールを選択してミドルウェアを変更してください。

GigaSpace ミドルウェアと Web リクエストを使用した場合に変更が必要な設定内容は以下のとおりです。

Magic の動作環境

- メッセージサーバ …… 「Default Soace」に変更します。
- HTTP リクエスト …… 「MgWebRequester.dll」に変更します。

アプリケーションの起動環境

GigaSpace が起動され Magic スペースが配備されたら Magic アプリケーションを起動するように設定する必要があります。

設定ファイルは、<Magic xpaインストールフォルダ>GigaSpaces-xpa¥config¥projectsStartup.xml で、以下のように設定します。

```
<Startup>
<Projects>
  <!-- one xpa server should check-out licenses into the space: -->
  <Project Name="<アプリケーション名>" StartApplication="<アプリケーションのキャビネットファイル名>" CMDLineArgs="/
MessagingServer=Default Space /InternetDispatcherPath=/Magic3xScripts/MGWebRequester.dll /LicenseName=MGRMC14 /
WorkersCountReservedForUsersBasedLicenses=1 /WorkersCountReservedForThreadsBasedLicenses=1">
    <Servers>
      <Server host="サーバ名">
        <StartApplication></StartApplication>
        <CMDLineArgs></CMDLineArgs>
        <NumberOfInstances>1</NumberOfInstances>
        <NumberOfWorkers>5</NumberOfWorkers>
      </Server>
    </Servers>
  </Project>
```

または、以下のようにも指定できます。

```
<Startup>
<Projects>
  <!-- one xpa server should check-out licenses into the space: -->
  <Project Name="<アプリケーション名>">
    <Servers>
      <Server host="サーバ名">
        <StartApplication><アプリケーションのキャビネットファイル名></StartApplication>
        <CMDLineArgs>/MessagingServer=Default Space /InternetDispatcherPath=/Magic4xScripts/MGWebRequester.dll /LicenseName=MGRMC14 /
WorkersCountReservedForUsersBasedLicenses=1 /WorkersCountReservedForThreadsBasedLicenses=1</CMDLineArgs>
        <NumberOfInstances>1</NumberOfInstances>
        <NumberOfWorkers>5</NumberOfWorkers>
      </Server>
    </Servers>
  </Project>
```

以下のパラメータは、使用するライセンス数に基づいて設定してください。

- WorkersCountReservedForUsersBasedLicenses …… 起動時に予約されるライセンス数
- WorkersCountReservedForThreadsBasedLicenses …… スレッド単位で予約されるライセンス数
- NumberOfInstances …… 起動するインスタンス数



- NumberOfWorkers …… 利用できるライセンス数

ノート

赤字のパラメータは必須です。それ以外は、MAGIC.INI で設定されていれば省略できます。

公開用 HTML ファイル

rcExecProps アイランドのリクエストの設定を変更します。

```
<property key="requester" val="/Magic4xScripts/MgWebRequester.dll"/>
```

RIA クライアントを（ClickOnce を使用せず）手動でインストールする

組織のセキュリティポリシーのために、PC に RIA アプリケーションをプリインストールしたいが ClickOnce を使用したくない場合があります。

ClickOnce を使用しないで RIA アプリケーションを配備して実行したい場合、コマンドラインを使用して行うことができます。

1. 最初に RIA クライアントファイルをクライアント PC にコピーする必要があります。クライアントファイルは、`¥RIAModules¥Desktop` ディレクトリにインストールされています。
2. 一旦、クライアント PC にこのフォルダがコピーすれば、そのフォルダから `Mgxpria.exe` ファイルを実行することによって RIA クライアントを実行することができます。
3. クライアントを実行する前に実行環境を設定する必要があります。これは、2 つの方法で行うことができます。
 - a. `execution.properties` を使用して `Mgxpria.exe` を実行します。
例：`Mgxpria.exe/ini=execution.properties`
`execution.properties` ファイルは、リッチクライアントインタフェースビルダによって作成される `<app_name>.publish.html` ファイルの `rcExecProps` セクション内と同じ情報（例えばサーバ名やアプリケーション名）が含まれていなければなりません。
`execution.properties` ファイルが `Mgxpria.exe` ファイルと同じディレクトリにある場合、単に `Mgxpria.exe` ファイルを実行するだけで可能になります。
 - b. コマンドラインからパラメータを付加して `Mgxpria.exe` を実行してください。コマンドラインの書式は、以下の通りです。

```
Mgxpria.exe protocol=xxx server=xxx requester=xxx appname=xxx prgname=xxx
```

例：

```
Mgxpria.exe protocol=http server=localhost requester=Magic4xScripts/MGrqispi.dll appname=RIATrans prgname=RC
```

ノート

コマンドラインを使用してクライアントでアプリケーションを実行する場合、クライアント側のバージョンがサーバ側と同じことを確認する必要があります。

Magic サーバのバージョンを更新する

RIA の実行環境では、サーバ側のバージョンと RIA クライアントのバージョンを合わせる必要があります。

例えば、ホットフィックスとして新しいクライアントを古いサーバ側と組み合わせて使用する場合、いくつかのシナリオがあります。この場合、動作はしますが、同じバージョンになるように、クライアントとサーバをアップデートするようなメッセージが表示されます。例えば、マイナー/メジャーなバージョンの変更があった場合、内部の通信プロトコルの変更のために、異なるクライアントとサーバのバージョンを使用することができなくなります。

実行時に ClickOnce を使用すると、Magic サーバのバージョンを更新した後に、(リッチクライアントインタフェースビルダを使用して作成された) 実行パッケージをアップデートしなければなりません。一旦アプリケーションを開始すると、クライアント側のバージョンを更新します。

ClickOnce を使用していない場合、(新旧のバージョンの) 2 つの Magic xpa サーバを維持することが推奨されます。各バージョンには、異なるリクエスト用エイリアスを含む独自のエン트리 URL があります。

この場合、以前のクライアントは古いサーバに接続し、新しいクライアントは新しいサーバに接続します。以前のバージョンの場合、ユーザに対して更新バージョンをインストールするように指示するメッセージを表示してアプリケーションをアップデートすることができます。これは、通常モバイルクライアントで必要となります。この場合、クライアントバージョンをコントロールすることはできません。

付録

リッチクライアントモジュール

リッチクライアントモジュールには以下のものがあります。

分散型アプリケーション・アーキテクチャ：Magic xpa によって提供されます。

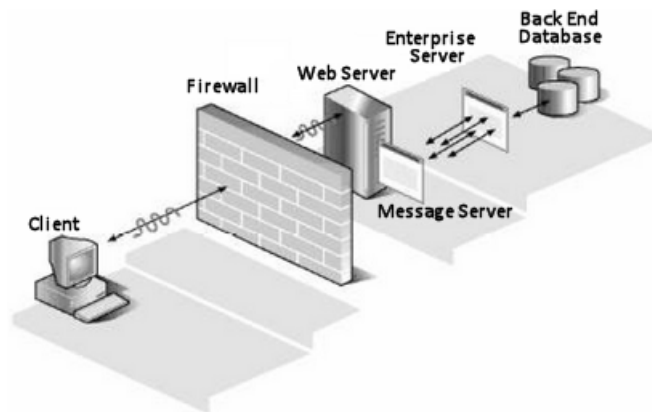
- Web サーバ …… リモートクライアントからリクエストを受け取るために、Web サーバが必要です。Web サーバは、Magic xpa サーバにリクエストを送ります。
- インターネットリクエスト …… クライアントと Magic xpa サーバの間を仲介するモジュール。このモジュールは、サーバのプールで1つのアイドル状態の Magic サーバにリクエストとデータを渡すことができます。これによって負荷を分散します。
- MRB (リクエストブローカ MRB) …… Magic xpa サーバのプールを維持します。Magic xpa は、リクエストブローカとして知られているミドルウェアエージェントを提供します。MRB は、利用可能なすべての Magic サーバエンジンを処理して、インターネットリクエストから利用可能なサーバエンジンへの各リクエストを指示します。MRB は、様々なフェールオーバーに対応するため、ロードバランシング機能やリカバリ機能を提供します。
- Magic サーバ …… Magic xpa サーバは、インタラクティブなリッチクライアント・アプリケーション環境の中心にあります。これは、実際の実行時単位で、各リクエストを処理して、受け取った各タイプのリクエストに対応した全てのアプリケーション・ロジックを実行します。Magic サーバは、MRB の位置を認識しており、それに接続し、それ自体でインターネットリクエストを利用できるようにする必要があります。

Magic サーバエンジンは、一回のエンジンプロセスを使用して複数のリクエストを処理するように設計されています。これは、Magic xpa サーバエンジンがマルチスレッド機能を使用して実現しています。

モジュールの分散

上記のモジュールは同じマシン上にインストールしたり、異なる OS を使用した異なるマシン上に分散させることもできます。

Magic xpa のインストール処理とアプリケーションサーバによって、必要なモジュールのインストールと実行のための環境設定は自動的に行われます。インターネットリクエストは、クライアント側から送られるリクエストを処理します。インターネットリクエストは、Magic エンジンと異なる PC 上に置くこともできます。



この図は、リッチクライアントがどのようにアプリケーションサーバと通信し、どのように返り、分散されている Magic xpa モジュール間でどのように相互作用しているかを表しています。