Deploying Drupal via Packages in a Hosting Environment

Introduction

Drupal is an open source content management system (CMS) often used as a back-end system for Web sites. This document provides step-by-step instructions on how to easily create a package containing the Drupal application for one-click deployment in a hosting environment. For more information about the Drupal application, visit the link: [Windows Web App page for Acquia Drupal](#).

Requirements

Environment Requirements:

- Windows Server 2008 R2 or greater
- IIS 7.0 or greater
- Web Deploy 2.0

Application Requirements:

- PHP 5.2
- MySQL 5.1
- Windows Cache 1.1 for PHP 5.2

For information about how to install the necessary components, see [Step 1. Set Up Your Servers for Hosting](#).

Procedure

This procedure describes how to install Drupal using IIS 7 (or greater) with Web Deploy. IIS 7 (or greater) with Web Deploy makes installing sites simple and extremely flexible for Hoster Service Providers by allowing them to create deployment packages that include the site, its content, and its database. The package can then be deployed by importing it into IIS via the console or a script. For more information about Web Deploy, refer to the link: [Web Deploy 2.0: The Official Microsoft IIS Site](#).

 This procedure requires 3 basic steps:

Step 1. [Set Up Your Servers for Hosting](#)
Step 2. [Create an Application Package for Deployment](#)

Step 1. Set Up Your Servers for Hosting

Overview

This step describes the server components that are required to support deployment of install packages via Web Deploy.

Prerequisites

While it is possible to run a web server and database server on the same machine, it is not recommended due to performance and security reasons. We recommend that you use two separate machines running Windows Server 2008 R2: one acting as the Web server and one acting as the database server.

Note: Both machines must be connected to the Internet to download the required server components.

To simplify installation of the Web server components, installing Web Platform Installer (Web PI) on both machines is highly recommended. Web PI is a tool that automates the installation of a vast majority of server components and other products for Microsoft's Web Platform.

Install Web Server Components on Your Web Server and Database Server

To install the required components on your Web server and database server using Web PI 3.0, visit the link: Install Spotlight Components plus PHP/MySQL related components using Web PI and click **Install**, following the onscreen instructions.

**Note:** A complete list of components that are installed by default with Web PI 3.0 is included in the

Install MySQL on your Database Server

1. Download the latest stable version of MySQL from the official website and run the installer on a dedicated database server.
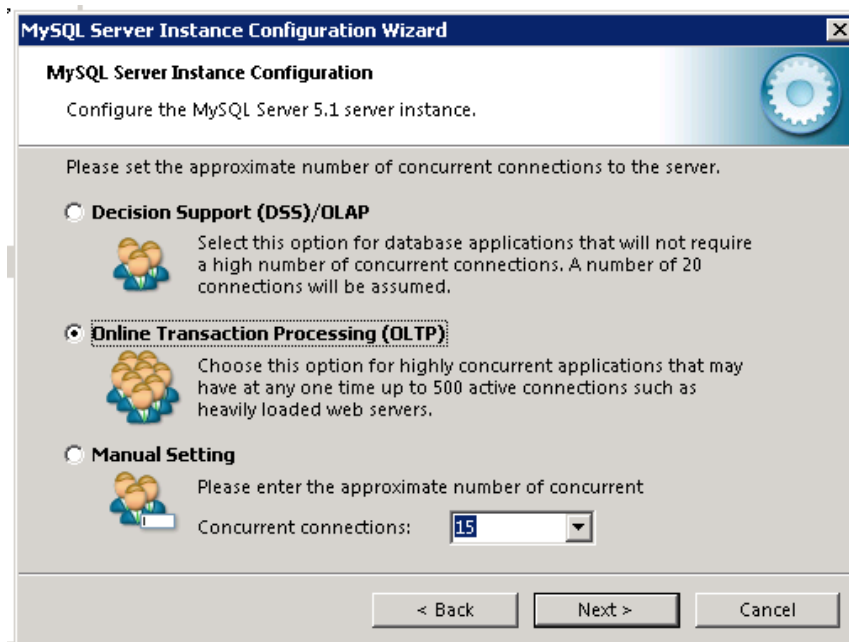
The MySQL Server Instance Configuration Wizard appears.
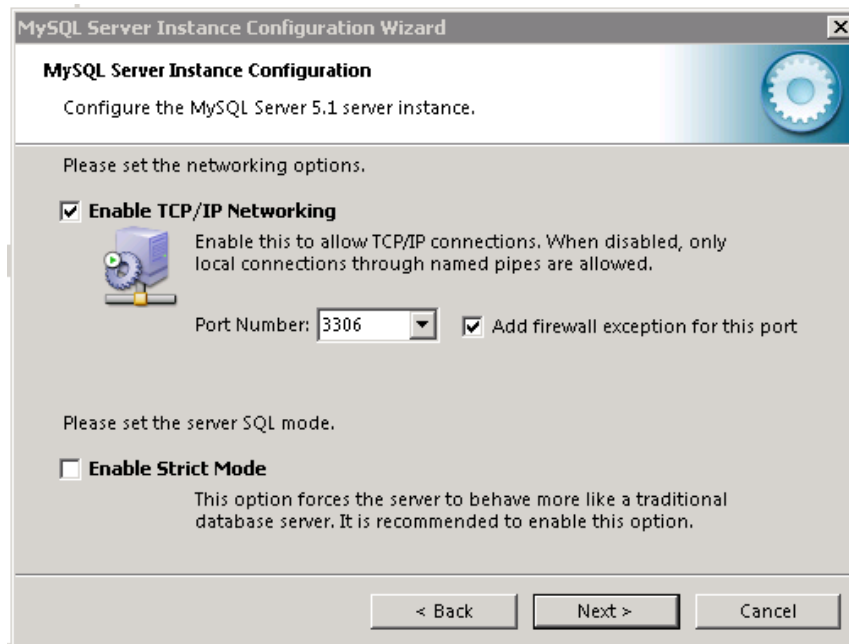

2. Click the **Dedicated MySQL Server Machine** option button, and click **Next**.
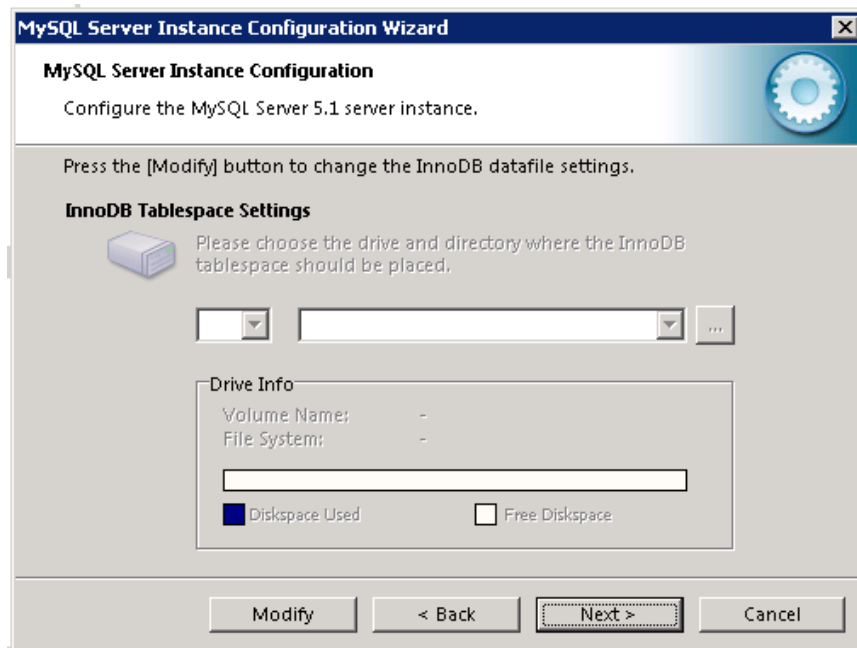
3. Click **Next**.

4. For concurrent connections, click the **Online Transaction Processing (OLTP)** option button (since this option reflects the workload of a typical shared hosting database server), and then click **Next**.


5. Click the **Enable TCP/IP Networking** check box, and then click the **Add firewall exception for this post** check box.

6. Clear the **Enable Strict Mode** check box if it is checked.
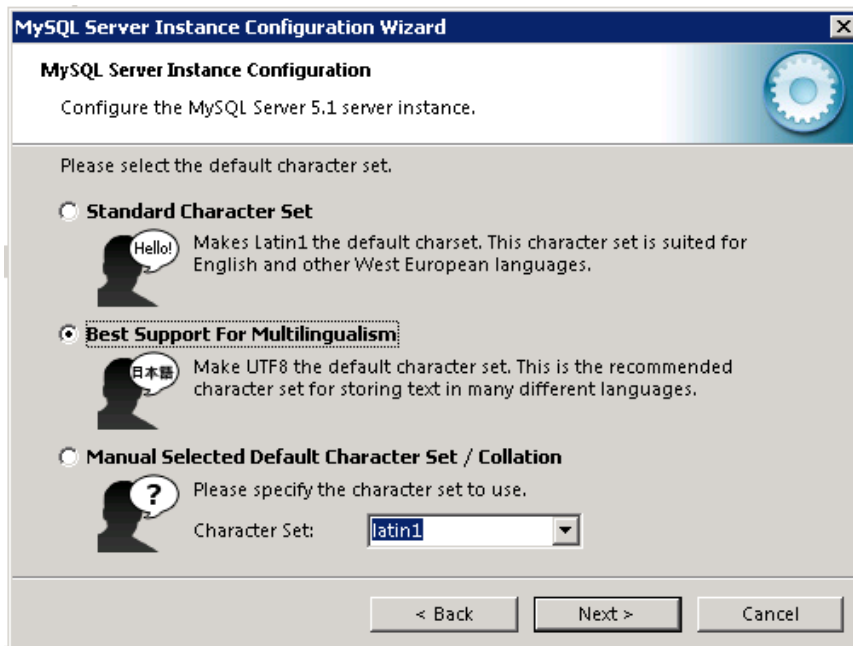
7. Click the Best Support For Multilingualism option button to enable support for the UTF-8 character set.

   IMPORTANT! You must choose the Best Support for Multilingualism option button since several applications in the Web Application Gallery require UTF-8 support in the database.

8. Click **Next**.
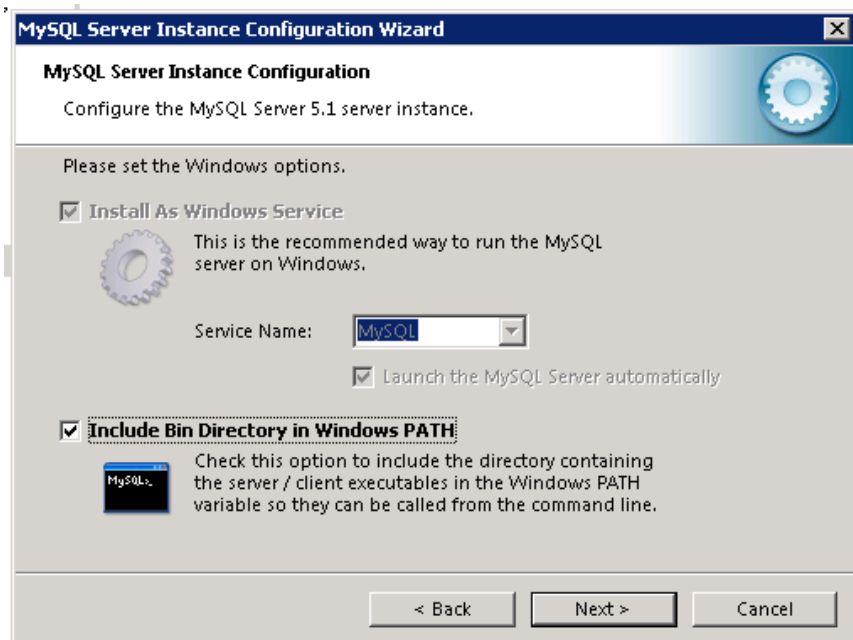
9. Click the **Include Bin Directory in Windows PATH** check box, and click **Next**.

10. Since MySQL is installed on the database server (which is different than the Web server that runs the Web Deploy web server component), you must complete these steps to ensure Web Deploy can access the database server:
    a. Copy mysqldump.exe (typically located in C:\Program Files\MySQL\MySQL Server 5.1\bin) to your Web server in C:\mysqldump\mysqldump.exe.
    b. On the Web server, set a registry key (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IIS Extensions\MSDeploy\1\mysqldumppath) to string value == "c:\mysqldump\mysqldump.exe"

Note: If you lose the credentials to your MySQL database, refer to the link: Resetting the Root Password: Windows Systems to reset your password.

Products Installed by Default using Web PI section of this document. You can also review the list of products before installation begins.

Install MySQL on your Database Server

To install MySQL, click here on this link and complete the steps on your database server:

Next Steps

- If the Web Deploy 2.0 component is newly installed, you must configure it to allow delegated deployments. Refer to the link: Configure the Web Deploy Component on the Web Server for more information.
- After you install and configure the server components, validate your server configuration by following the procedures described at the link: Validate Your Server Configuration.

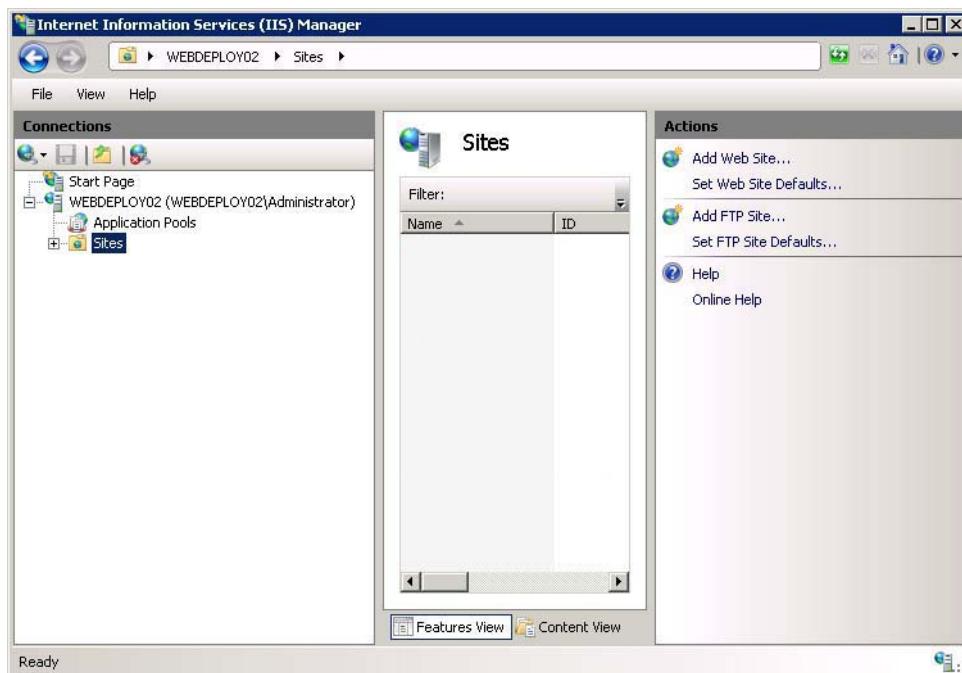# Step 2. Create an Application Package for Deployment

This step describes how to export an application package that can be used to quickly provision IIS 7 sites and includes these sub steps:

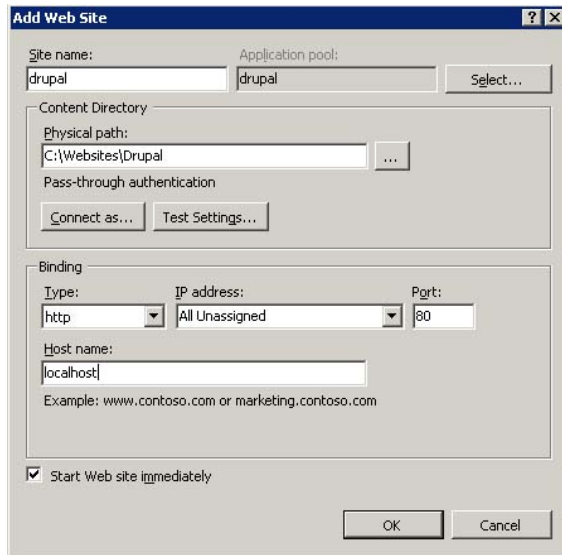## A. Create an IIS Web Site for this Application

1. Open Internet Information Services (IIS) Manager.

    The Internet Information Services (IIS) Manager application opens.

2. Expand your server connection, and click **Sites**.
3. In the Actions area, click the **Add Web Site** link.

    The Add Web Site dialog box appears.

4. Enter a site name for your Web site.
5. In the Physical path field, type the physical path to the folder in which the application is installed, or click the browse button (...) to navigate to the folder.
6. In the Binding area, select the protocol for your Web site from the Type drop-down list.
7. Type the IP address to your Web site in the IP address drop-down list. The default value is All Unassigned.
8. Type a port number in the Port field.
9. Type a host name for the Web site in the Host name field.
10. If you want the Web site to be immediately available and do not need to make further changes, select the **Start Web site immediately** check box.
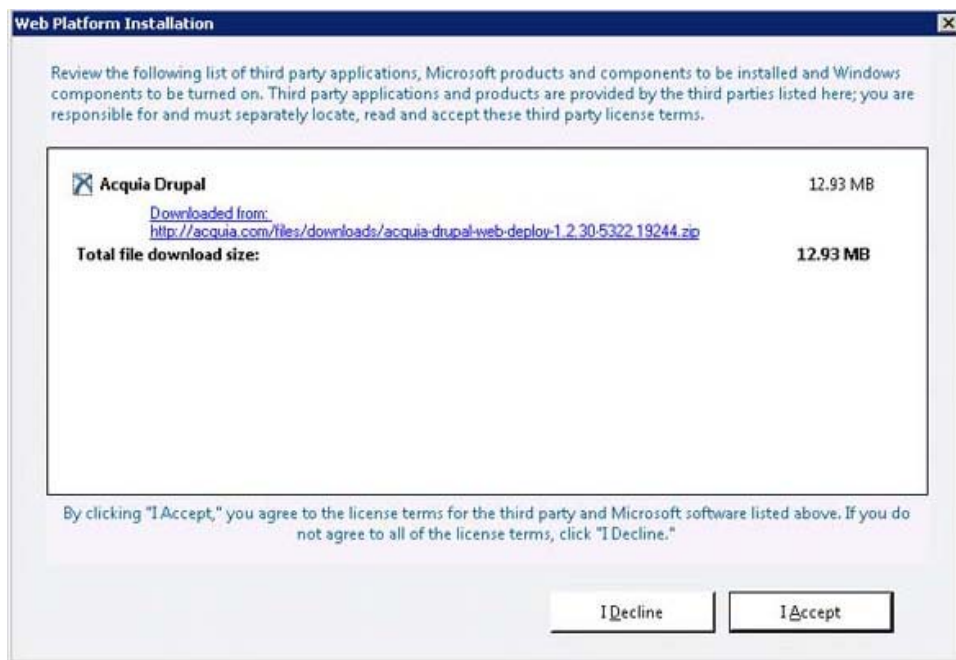11. Click **OK**.

B. Install the Application to the IIS Application Web Site

1. Open Web Platform Installer (Web PI) locally from the web server.
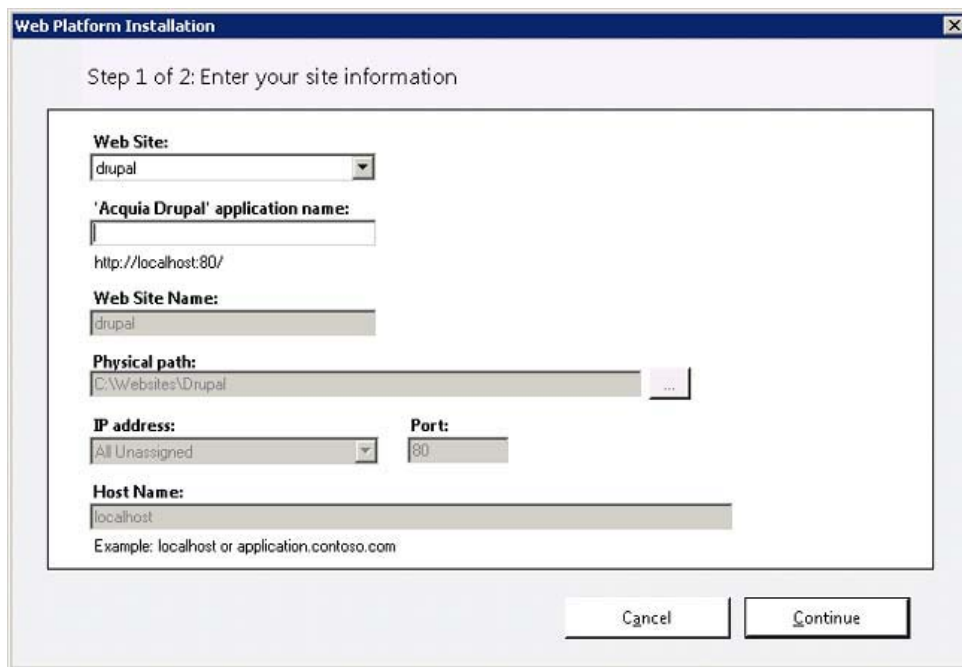   The Web Platform Installer application opens.
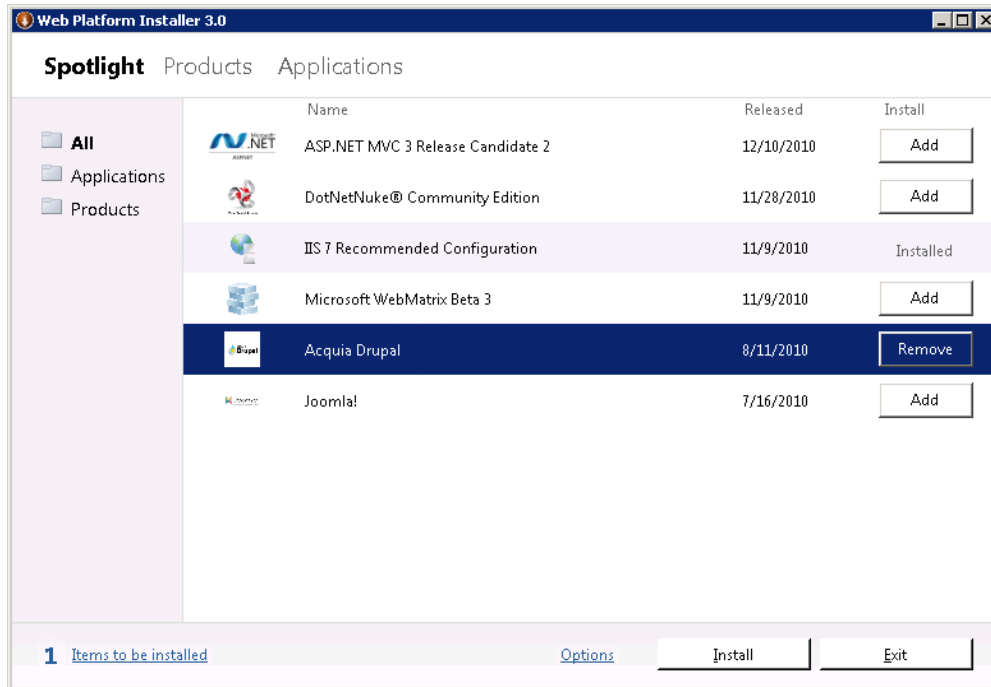
2. Select **Acquia Drupal**, and click **Install**.

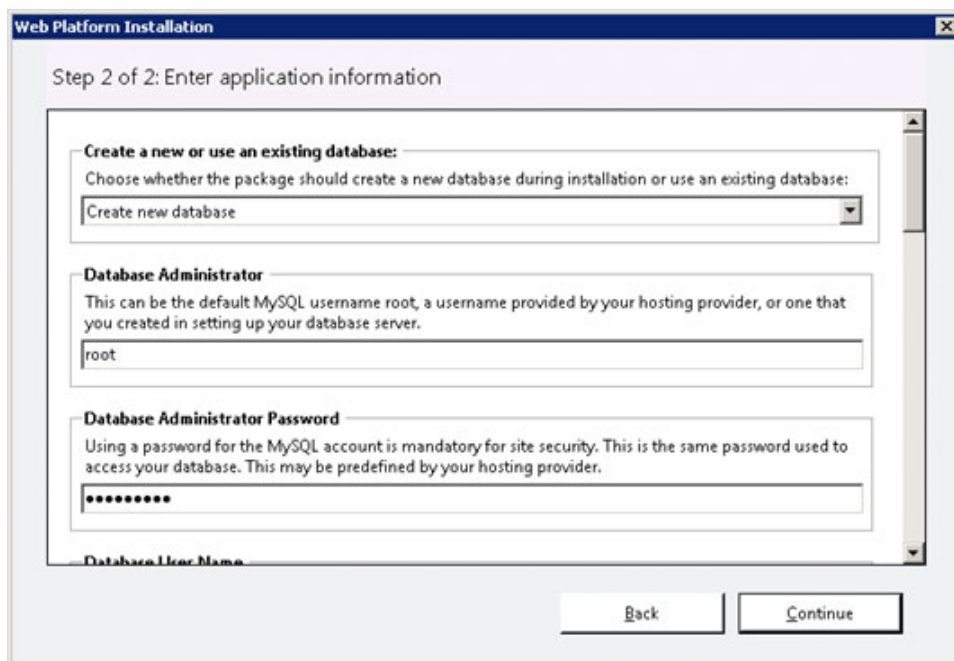   The Web Platform Installation window appears.

3.   Click **I Accept** to accept the end-user license agreement.

4.   From the Web Site drop-down list, select the Web site you created earlier in this procedure.

5.   In the application name field, enter an application name if it is different from the default value.

     The remaining fields are populated automatically based on your Web site information.
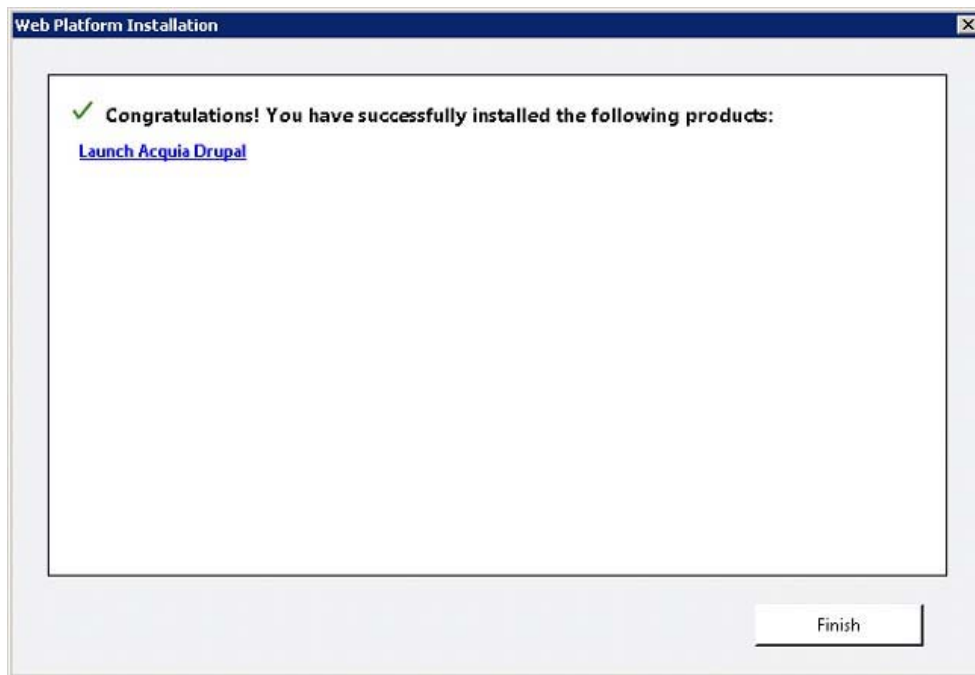
6.   Click **Continue**.

7.   From the Create a new or use an existing database drop-down list, select **Create new database**.

8. Enter the following information in the appropriate fields to create a new database:

- Database administrator user name and password—User name and password to allow administrative access to this database.

- Database user name and password—User name and password to allow general access to this database.

- Database server—Name of the server on which this database resides.

- Database name—Name of this database.

- Database prefix—Prefix added to database table names used to identify this database.

- Load sample data—Indicate whether you want to load sample data into this database.

- Web site name—Name of your Web site.

- Site administrator—User name for the Web site specified in the Website Name field.

- Site administrator password—Password for the Web site administrator.

- Site administrator email address—Email address for the Web site administrator.

Note: You must scroll down to view all required fields.

9. Click **Continue**.

10. Once the Web PI process is complete, you can do one of the following:

- Click **Finish** to complete the installation process.

If you click Finish to complete the installation process, proceed to step to continue creating an application package for deployment.

- Click **Launch Drupal** to launch your Drupal application.

If you click the Launch Drupal link, the Drupal application page appears:
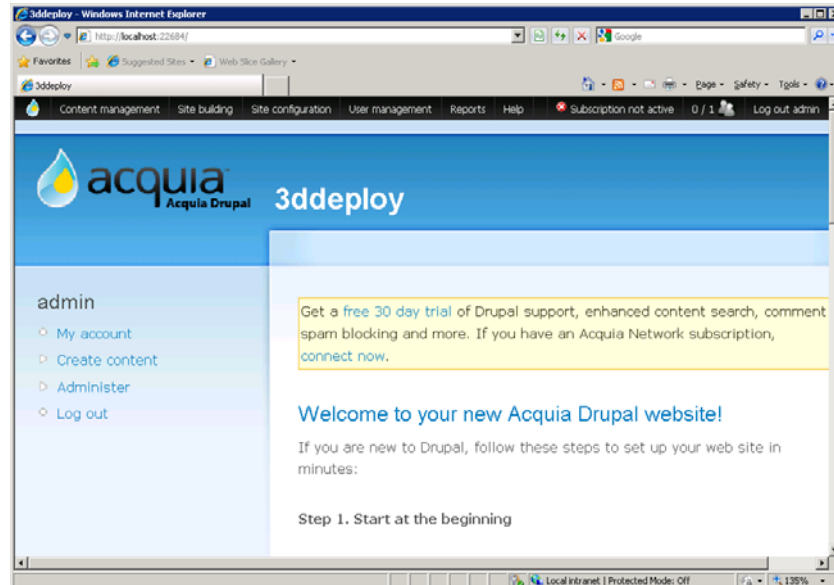
For information about setting up your Drupal site, refer to the link: Drupal documentation Web page.

C. Enable Permalinks (Optional)

By default, Drupal uses query string parameters for all the URLs it generates for your Web site. You can change this default behavior by enabling clean URLs, a feature which relies on the URL rewriting functionality. To enable permalinks (clean URLs) in Drupal, complete these steps:

1. Open the Web.config file located in the root of the Drupal install directory. The root directory path should be similar to the following: C:\inetpub\wwwroot\drupal
2. Copy and paste the following XML code into the Web.config file.

```
<?xml version="1.0" encoding="UTF-8"?>
  <configuration>
    <system.webServer>
      <rewrite>
        <rules>
          <rule name="Drupal Clean URLs" stopProcessing="true">
            <match url="^(.*)$" />
            <conditions>
              <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
```

```
            <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
          </conditions>
          <action type="Rewrite" url="index.php?q={R:1}" appendQueryString="true" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```

3. Save the web.config file.

4. Launch your Drupal application in a Web browser and log in as the administrator

5. Browse to **Administer > Site Configuration > Clean URLs**.

6. Under **Clean URLs**, click the **Enabled** option button, and then click **Save configuration**.

D. Export the Package from the IIS Application Web Site

1. Open IIS Manager.

    The Internet Information Services (IIS) Manager application opens.

2. In the Connections navigation pane, expand **Sites**, and select the Web site you just created.

3. Click the **Export Application** link located in the Deploy section of the Actions pane.

The Export Application Package (Select the Contents of the Package) dialog box appears.
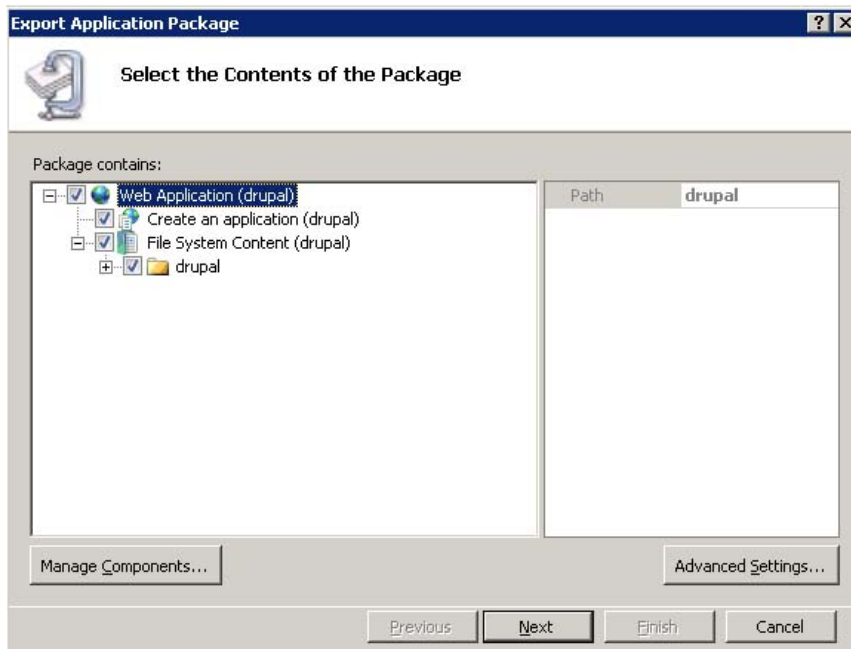
4. Click **Manage Components**.

The Manage Components dialog box appears.
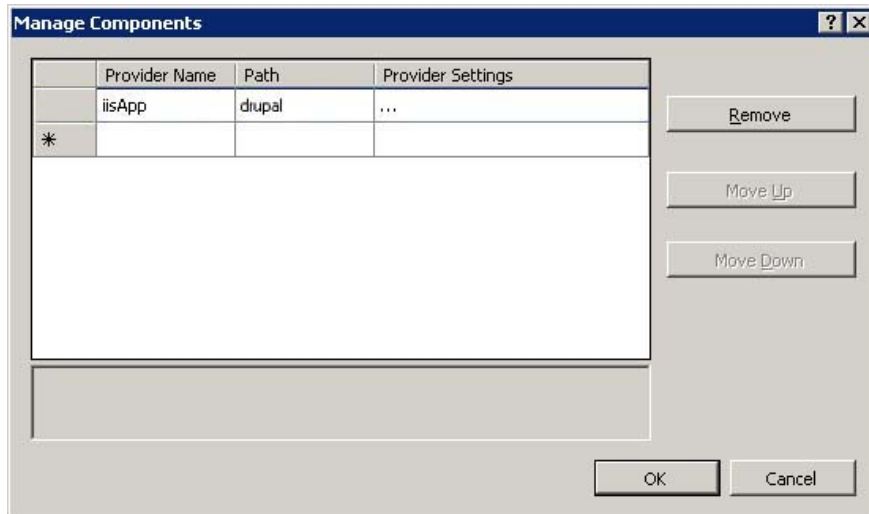
5. Double-click the second row of the Provider Name column.
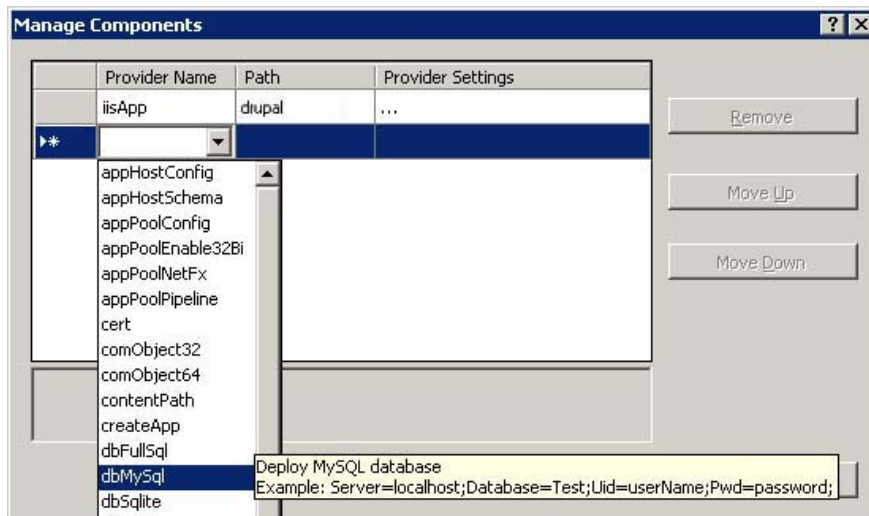
6. Select **dbMySql** from the drop-down list.
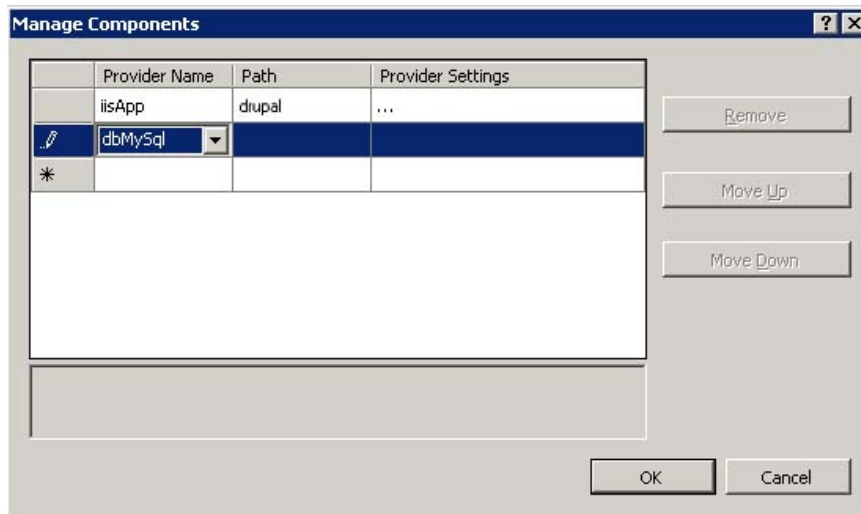
7. Double-click the second row of the Path column, and enter the connection string for your package export.
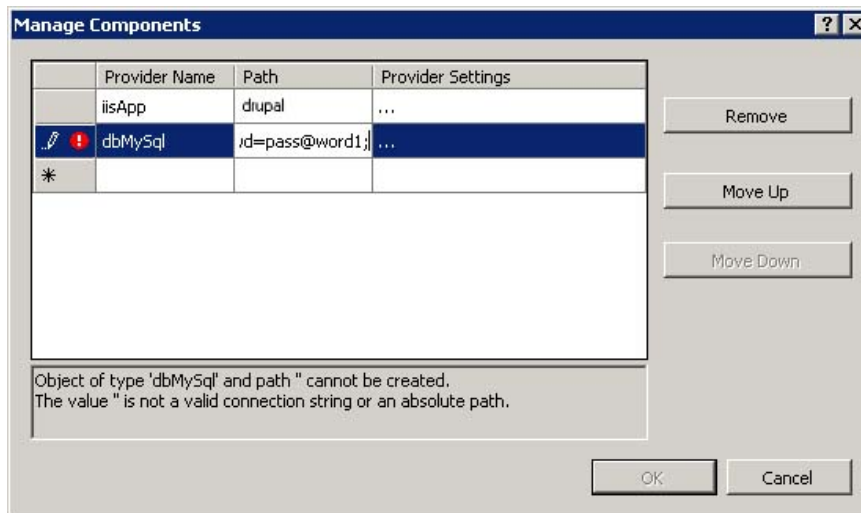
8. Click outside of the box in which you entered the connection string, and click **OK**.

The newly created parameter now appears in the Export Application Package (Select Parameters) dialog box.
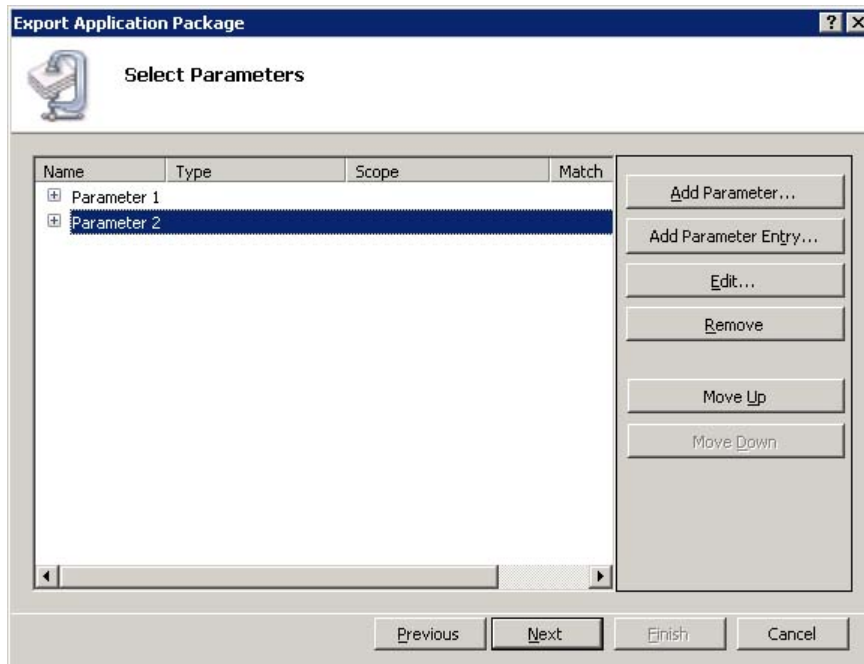
9. Rename the parameters:

a. Select **Parameter 1**, and click **Edit**.

   The Edit Parameter dialog box appears.

b. Enter **SiteName** in the Name field, and click **OK**.

c. Repeat these steps to rename Parameter 2 to "ConnectionString."

The parameters appear renamed in the Export Application Package (Select Parameters) dialog box.
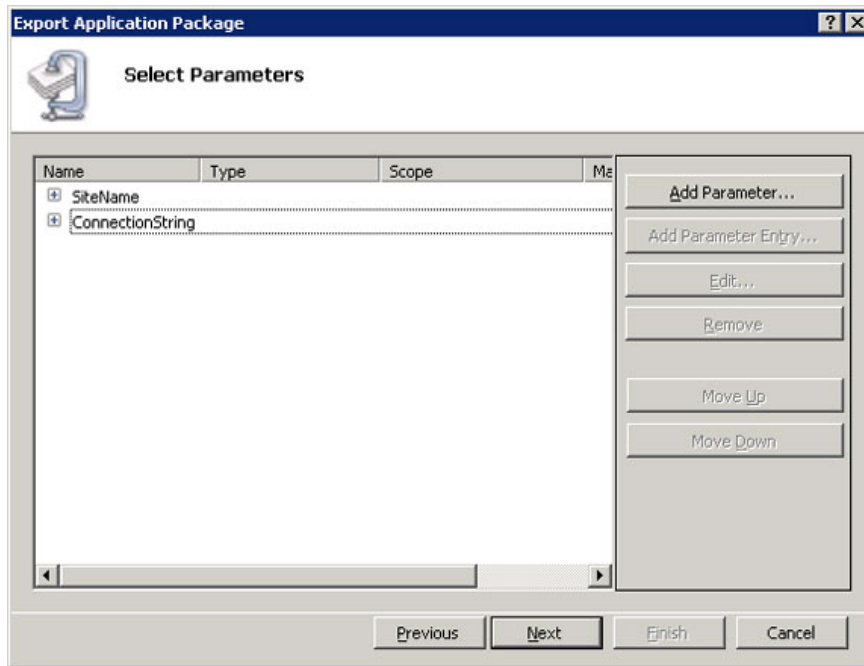
10. Click **Next**.

The Export Application Package (Save Package) dialog box appears.
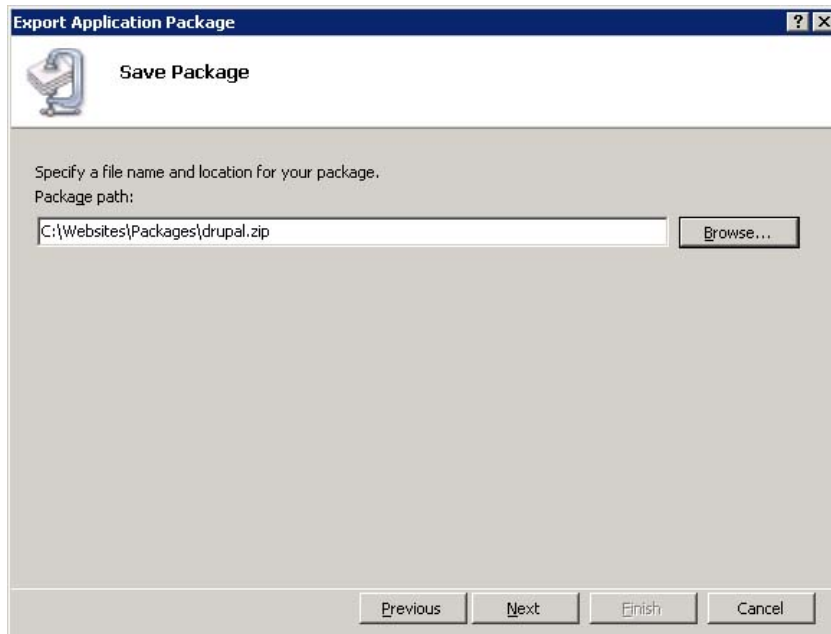
11. Enter the path to the application package, or click Browse to locate the application file.

12. Click **Next**.

The Export Application Package (Export Progress and Summary) dialog box displays the progress of the export process. Once the process is complete, a summary report appears.
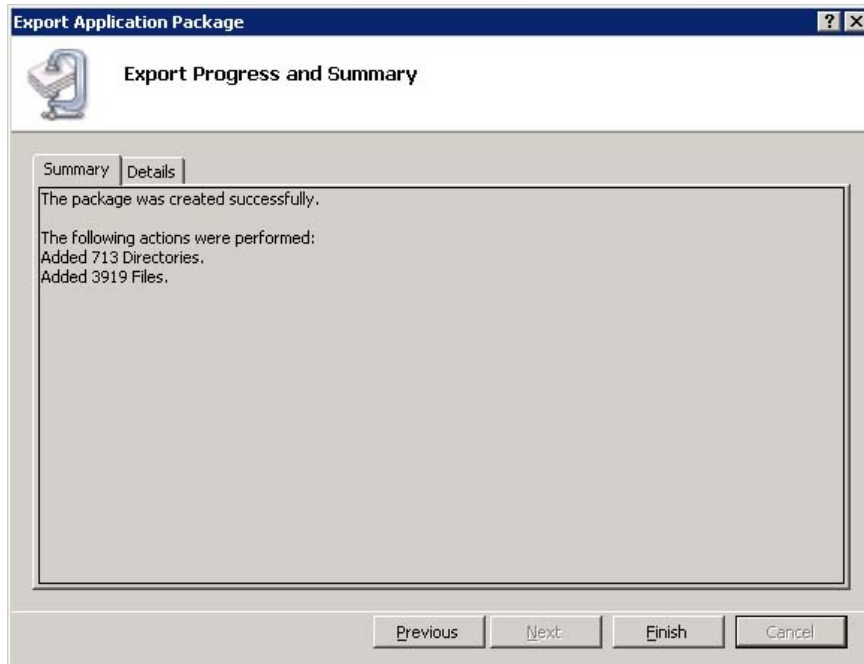
13. Click **Finish**.

Use this application package to deploy to customer Web sites as described in

Step 3. Deploy a Customer Site using IIS 7 with Web Deploy.

Step 3. Deploy a Customer Site using IIS 7 with Web Deploy

Overview

When deploying a customer site, most Hosting Service Providers use a control panel that allows them to capture the information required for deployment. This information can then be passed to scripts that create the user, configure the site for hosting, create a database, and import the application package. Here are the basic steps required to deploy a customer site:

A. Create a Customer Application Web Site
Tips:

When you create the Web site in IIS, give your application a user-friendly name by creating a fully qualified domain name (FQDN) for the host header and adding that FQDN to your Domain Name Service.

- You can automate this process using a script similar to the one described at the link: Automated Deployment Script.

B. Create a Customer Application Database
C. Import the Application Package using a CLI Command or PowerShell Script

A. Create a Customer Application Web Site

To create a customer application Web site, complete the steps described in

A. Create an IIS Web Site for this Application.

Tips:
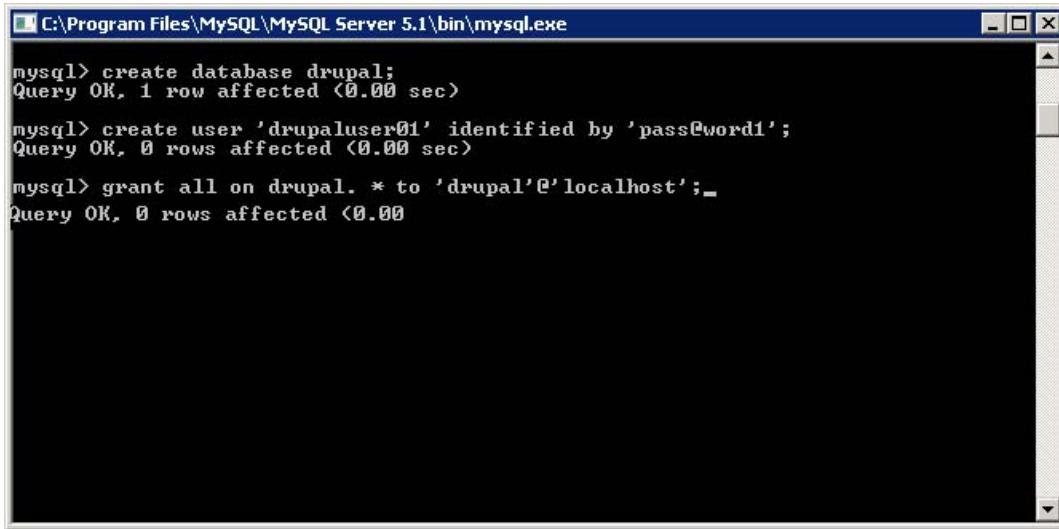
- When you create the Web site in IIS, give your application a user-friendly name by creating a fully qualified domain name (FQDN) for the host header and adding that FQDN to your Domain Name Service.
- You can automate this process using a script similar to the one described at the link: Automated Deployment Script.

B. Create a Customer Application Database

To create a customer application database, complete these steps:

1. Open a command-line prompt (or MySQL Manager) at the MySQL bin directory. If you used the default installation path, the bin directory path will be similar to the following: C:\Program Files\MySQL\MySQL Server 5.1\bin
2. Enter the administrator password, and press **Enter**.
3. Type the following command to create the database, and press **Enter**.
   ```
   mysql> create database drupal;
   ```
4. Type this command to create a user, and press **Enter**.
   ```
   mysql> create user 'drupaluser'@'localhost' identified by 'password';
   ```
5. Type this command to grant permissions, and press **Enter**.
   ```
   mysql> grant all on drupal. * to 'drupal'@'localhost';
   ```

Your command window should appear similar to the following:

Tip: You can automate this process using a script similar to the one described at the link: MySQL Database Creation Script.

C. Import the Application Package using a CLI Command or PowerShell Script

Once you export the application package (as described in Step 2. Create an Application Package for Deployment), you can easily import the package into the customer site using the CLI command or PowerShell script listed below. These methods allow you to add the parameters that are required to configure an application for use.

The examples in this topic show how to import Drupal into an IIS Web site and install the application database. This package is configured to allow two parameters: one for the IIS site name and one for the database connection string.

CLI Command

msdeploy.exe -verb:sync -source:package=C:\Websites\Packages\Temp\drupalMSDeployPackage.zip
   -dest:auto -setParam:name="SiteName",kind=ProviderPath,scope=iisApp,Value=drupal/;
   -setParam:name="ConnectionString",kind=ProviderPath,scope=dbMySql,Value=server=localhost;
   database=drupal;Uid=drupal;Pwd=password; > DWSpackage7.log

PowerShell Script

```
[Array]$arguments = "-verb:sync", "-source:contentPath=`"$web_staging_directory`"",
    "-dest:contentPath=`"\\$Server\$share\$appname\$web_project_name`"""
    $proc = Start-Process $msdeploy -ArgumentList $arguments -NoNewWindow -Wait -PassThru
if($proc.ExitCode -ne 0)  { throw "Failed to deploy" }
```

Sample Powershell Scripts

Automated Deployment Script

```
# Web Deploy: Powershell script to setup IIS sites for Hosting.
# Copyright (C) Microsoft Corp. 2010
#
# Requirements: IIS 7, Windows Server 2008 (or higher)
#
# You should use this script if you want to set up a Windows 2008 (or higher)
# server for delegated Web Deploy deployments.
#   1. Create an IIS Site
#   2. Creates an IIS Manager user and assigns it
#        permissions to the IIS Site Created
#
# ================= PARAMS =======================
#
# All of these parameters are required.
#
# $website              IIS Web site.
# $iisManagerUsername   User name of IIS Manager user
# $iisManagerPassword   Password of above user.
# $hostheaderName       Web site Host Header/FQDN
#
# All of these parameters are optional.
#
# $wmsvcUsername,
# $directory         App physical directory       c:\websites by default.
# $serverPort        Web site Port                80 by default.
#
# sample usage AddHostingSite.ps1
#       -website MyWebSite01
#       -iisManagerUsername TestUser01
#       -iisManagerPassword TestPass01
#       -hostheaderName www.mysite.com
#       -directory c:\Websites -serverPort 80

param($website, $iisManagerUsername, $iisManagerPassword, $hostheaderName, $wmsvcUsername,
$directory,  $serverPort)

clear-host

sv APPCMD -value $env:systemroot\system32\inetsrv\AppCmd.exe

# Check if params are missing
if($website -eq $null){
    write-host 'Please specify a website. Script aborting.'
    break
```

```
}
if($iisManagerUsername -eq $null){
     write-host 'Please supply an IIS Manager username to create account. Script aborting.'
     break
}
if($iisManagerPassword -eq $null){
     write-host 'Please supply an IIS Manager password to create account. Script aborting.'
     break
}
if($wmsvcUsername -eq $null){
     $wmsvcUsername = "LOCAL SERVICE"
}
if($directory -eq $null){
     $directory = "c:\Websites"
}
if($hostheaderName -eq $null){
     write-host 'Please specify a domain name for the websites host header. Script aborting.'
     break
}
if($serverPort -eq $null){
     $serverPort = '80'
}

# ================= GLOBAL VARIABLES ====================

# IIS management assemblies
$ENV_APPPOOL_NAME = $website + "_AP"

$global:mwaAssembly = $null
$global:mwmAssembly = $null

# the instance Microsoft.Web.Administration.ServerManager we use to interact
# with IIS's administration.config
$global:serverManager = $null

# collection of Web Deploy delegation rules
$global:delegationRulesCollection = $null

# =============== METHODS ======================

function LoadAssemblies{
    trap [Exception]{
        write-host 'Failed to load Microsoft.Web.*.dll. Are you sure IIS 7 is installed?'
        break
    }
    $global:mwaAssembly = [System.Reflection.Assembly]::LoadFrom(
[System.Environment]::ExpandEnvironmentVariables("%WINDIR%") +

"\system32\inetsrv\Microsoft.Web.Administration.dll" )
    $global:serverManager = (New-Object Microsoft.Web.Administration.ServerManager)
    $global:mwmAssembly = [System.Reflection.Assembly]::LoadFrom(
[System.Environment]::ExpandEnvironmentVariables("%WINDIR%") +

"\system32\inetsrv\Microsoft.Web.Management.dll" )
}

 function NotServerOS{
    $sku = $((gwmi win32_operatingsystem).OperatingSystemSKU)
    $server_skus = @(7,8,9,10,12,13,14,15,17,18,19,20,21,22,23,24,25)
```

```
    return ($server_skus -notcontains $sku)
}

 function CheckDelegationRulesExist{
    trap [Exception]{
        write-host 'Did not find delegation rules in administration.config'
        return $false
    }
    $global:delegationRulesCollection =
$serverManager.GetAdministrationConfiguration().GetSection("system.webServer/management/delegatio
n").GetCollection()

    if($global:delegationRulesCollection.Count -eq 0){
        return $false
    }
    else{
        return $true
    }
}

function CreateAndAuthorizeIISManagerUser {
    param ($username, $pwd)
    trap [Exception]{
        write-host "Could not create and / or authorize IIS Manager user on Default Web Site:
$username"
    }
    [Microsoft.Web.Management.Server.ManagementAuthentication]::CreateUser($username, $pwd)
    [Microsoft.Web.Management.Server.ManagementAuthorization]::Grant($username, $website, $FALSE)
    write-host "Created IIS Manager user: $username and granted it permissions on website:
$website"
}


 function GetPhysicalPathOfWebsite{
    $path = ""
    if($website -eq $null){
        $path = $serverManager.Sites[0].Applications[0].virtualDirectories[0].physicalPath
    }
    else{
        $i = 0
        $found = $false
        for ($i=0; $i -lt $serverManager.Sites.Count; $i++){
            if($serverManager.Sites[$i].Name -eq $website){
                $found = $true
                break;
            }
        }
      if($found){
            $path = $serverManager.Sites[$i].Applications[0].virtualDirectories[0].physicalPath
      }
    }
    # if website doesn't exist, create it
    if($path -eq ""){
            $fp = $directory+ "\" +$website
        $global:serverManager = (New-Object Microsoft.Web.Administration.ServerManager)
        $global:serverManager.Sites.Add($website, $fp, 8080)
        $global:serverManager.CommitChanges()
      new-item $fp -type directory
        $path = $fp
      # Create the AppPool for this site
```

```
        Invoke-Expression "$APPCMD add apppool /name:$ENV_APPPOOL_NAME"
        Invoke-Expression "$APPCMD set apppool /apppool.name:$ENV_APPPOOL_NAME
/processModel.identityType:NetworkService"
        Invoke-Expression "$APPCMD add app /site.name:$WebSite /path:/ /physicalPath:`"$path`"
/applicationPool:$ENV_APPPOOL_NAME"


        #appcmd add site /name: $website /physicalPath: $path   /bindings:http/*:$serverPort:
$hostheaderName


        echo $error


        write-host "Created new website name: $website, directory: $path , port:$serverport"
    }


    return $path
}


#================= Main Script =================

if(NotServerOS){
    write-host 'Please run this script on a server OS only. Script aborting.'
    break
}


LoadAssemblies

$physicalPathOfWebSite = GetPhysicalPathOfWebsite
$physicalPathOfApplicationHost = [System.Environment]::ExpandEnvironmentVariables("%WINDIR%") +
"\system32\inetsrv\config\applicationHost.config"


CreateAndAuthorizeIISManagerUser $iisManagerUsername $iisManagerPassword
```

MySQL Database Creation Script

```
# Powershell Args
$dbusername = $args[0]                                   # Administrative Username
$dbpassword = $args[1]                                   # Administrative Password
$dbname = $args[2]                                       # Database Name to Create
$appUsername = $args[3]                                  # Application Username
$appPassword = $args[4]                                  # Application Users Username

# Add MySQL Data Connector
[void][system.reflection.Assembly]::LoadWithPartialName("MySql.Data")

# Open Connection to SQL Server
$connStr = "server=127.0.0.1;port=3306;uid=root;pwd=SQLPassword"
$conn = New-Object MySql.Data.MySqlClient.MySqlConnection($connStr)
$conn.Open()

# Create MySQL Database
$createmysqldatabase = 'CREATE DATABASE `' + $dbname + '`'
$cmd = New-Object MySql.Data.MySqlClient.MySqlCommand($createmysqldatabase, $conn)
$da = New-Object MySql.Data.MySqlClient.MySqlDataAdapter($cmd)
$ds = New-Object System.Data.DataSet
$da.Fill($ds)

# Create MySQL User
$createmysqluser = 'CREATE USER `' + $appUsername  + '`@`localhost` ' + ' identified by `' +
$appPassword + '`'
```

```
$cmd = New-Object MySql.Data.MySqlClient.MySqlCommand($createmysqluser, $conn)
$da = New-Object MySql.Data.MySqlClient.MySqlDataAdapter($cmd)
$ds = New-Object System.Data.DataSet
$da.Fill($ds)

# Grant permissions to Database
$grantmysqldatabaseperms = 'Grant ALL on ' + $dbname + '.* to `' + $appUsername  + '`@`localhost`
'
$cmd = New-Object MySql.Data.MySqlClient.MySqlCommand($grantmysqldatabaseperms, $conn)
$da = New-Object MySql.Data.MySqlClient.MySqlDataAdapter($cmd)
$ds = New-Object System.Data.DataSet
$da.Fill($ds)
```
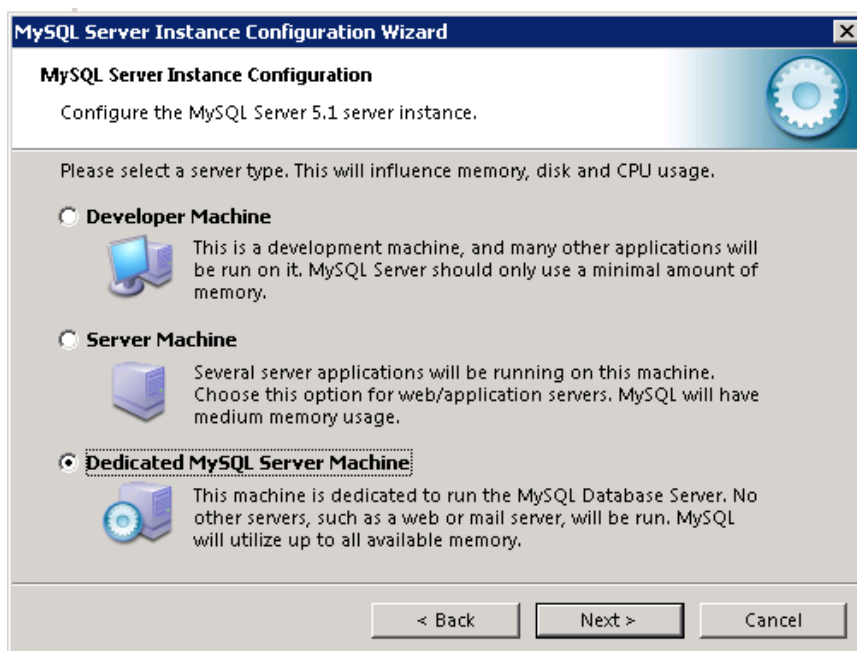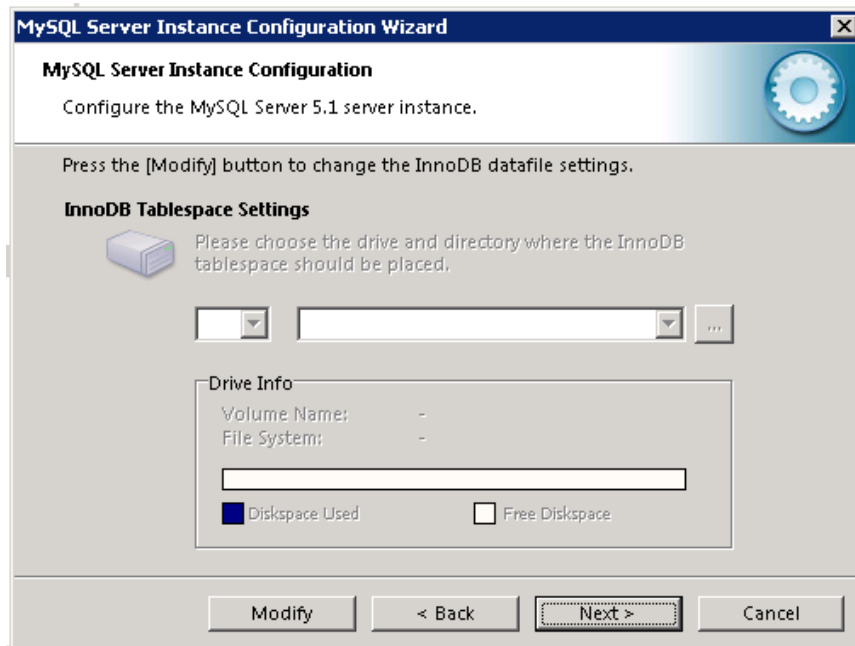
Install MySQL on your Database Server

11. Download the latest stable version of MySQL from the official website and run the installer on a dedicated database server.

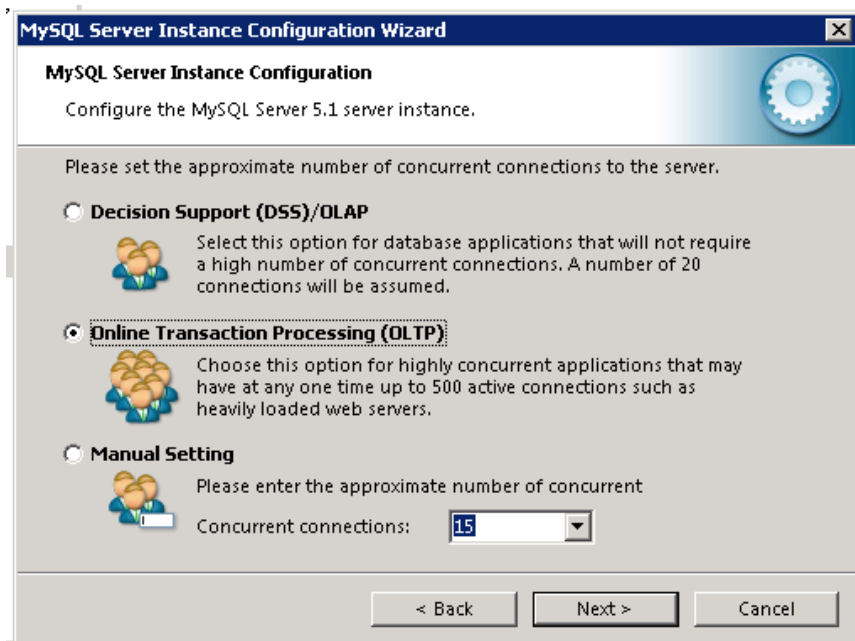   The MySQL Server Instance Configuration Wizard appears.

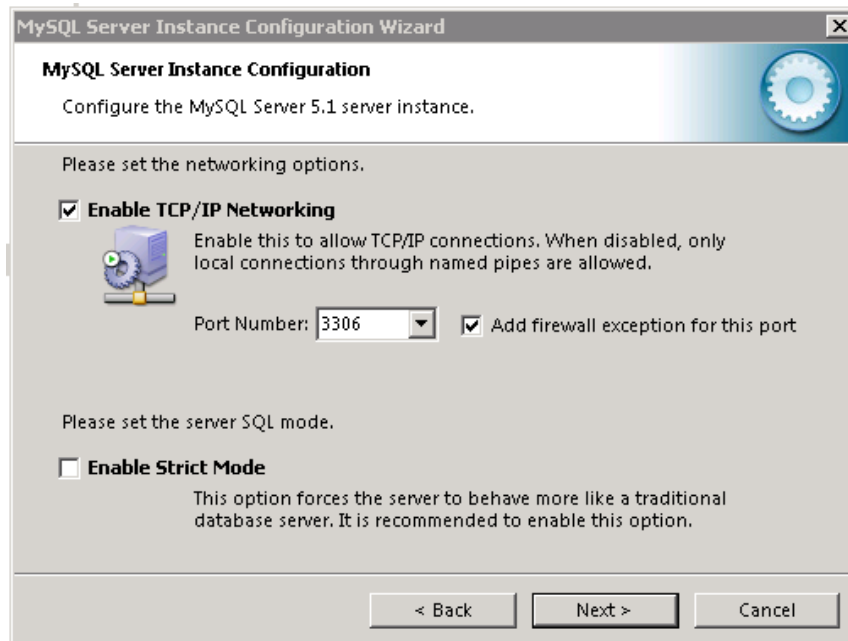12. Click the **Dedicated MySQL Server Machine** option button, and click **Next**.

13. Click **Next**.

14. For concurrent connections, click the **Online Transaction Processing (OLTP)** option button (since this option reflects the workload of a typical shared hosting database server), and then click **Next**.

15. Click the **Enable TCP/IP Networking** check box, and then click the **Add firewall exception for this post** check box.

16. Clear the **Enable Strict Mode** check box if it is checked.
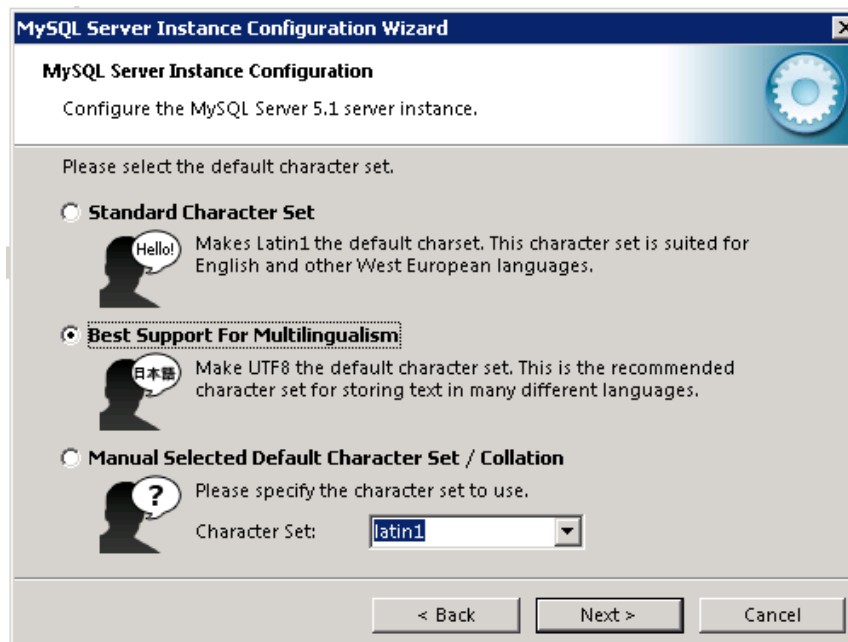
17. Click the Best Support For Multilingualism option button to enable support for the UTF-8 character set.
    IMPORTANT! You must choose the Best Support for Multilingualism option button since several applications in the Web Application Gallery require UTF-8 support in the database.

18. Click **Next**.

19. Click the **Include Bin Directory in Windows PATH** check box, and click **Next**.

20. Since MySQL is installed on the database server (which is different than the Web server that runs the Web Deploy web server component), you must complete these steps to ensure Web Deploy can access the database server:

    a. Copy mysqldump.exe (typically located in C:\Program Files\MySQL\MySQL Server 5.1\bin) to your Web server in C:\mysqldump\mysqldump.exe.

    b. On the Web server, set a registry key (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IIS Extensions\MSDeploy\1\mysqldumppath) to string value == "c:\mysqldump\mysqldump.exe"

Note: If you lose the credentials to your MySQL database, refer to the link: Resetting the Root Password: Windows Systems to reset your password.

Products Installed by Default using Web PI

This table provides additional information about the products that are installed by default using Web PI:

Product	Description
.NET Framework 3.5 SP1	.NET Framework 3.5 SP1 is a full cumulative update that contains many new features building incrementally upon .NET Framework 2.0, 3.0, 3.5, and includes cumulative servicing updates to the .NET Framework 2.0 and .NET Framework 3.0 subcomponents. For more information, refer to the link: Microsoft Download Center.
.NET Framework 4.0	The .NET Framework is Microsoft's comprehensive and consistent programming model for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes. For more information, refer to the link: .NET Framework Developer Center.
ASP.NET	ASP.NET is a free web framework that enables great Web applications. Used by millions of developers, it runs some of the biggest sites in the world. For more information, refer to the link: Microsoft ASP.net .
ASP.NET MVC	ASP.NET MVC is part of the ASP.NET Web application framework and is one of the two different programming models you can use to create ASP.NET Web applications. For more information, refer to the link: Microsoft ASP.net.
ASP.NET MVC3	ASP.NET MVC 3 builds on top of the features in ASP.NET MVC 1 and 2, adding on great features that both simplify your code and allow for deeper extensibility. For more information, refer to the link: Microsoft ASP.net.
ASP.NET Web Pages	The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages. For more information, refer to the Microsoft .net Framework SDK link: Introduction to ASP.NET Pages.
Common Gateway Interface (CGI)	CGI is a standard for interfacing external programs with information servers on the Internet. For more information, refer to the link: CGI: Common Gateway Interface.
FTP Publishing Service	The FTP Publishing Service for IIS 7.0 allows Web content creators to publish content more easily and securely to IIS 7.0 Web servers using modern Internet publishing standards. For more information, refer to the link: FTP Publishing Service.
IIS 7 Web Server	Internet Information Services (IIS) for Windows Server is a flexible, secure and easy-to-manage Web server for hosting anything on the Web. For more information, refer to the link: IIS.
IIS Remote Management Service	Internet Information Services (IIS) 7 Manager for Remote Administration provides end users and administrators with the ability to securely manage remote IIS 7 servers from Windows 7, Windows Vista, Windows XP, and

	Windows Server 2003. For more information, refer to the link: IIS Manager for Remote Administration.
Media Services 3.0	IIS Media Services, an integrated HTTP-based media delivery platform, delivers true HD (720p+) live and on-demand streaming, DVR functionality, and real-time analytics support to computers, TVs, and mobile devices. For more information, refer to the link: IIS Media Services.
Microsoft Driver for PHP for SQL Server 2.0 in IIS	The Microsoft Drivers for PHP for SQL Server provide connectivity to Microsoft SQL Server from PHP applications. For more information, refer to the link: Microsoft Drivers for PHP for SQL Server.
Microsoft SQL Server 2008	SQL Server delivers on Microsoft's Data Platform vision by helping your organization manage your data by enabling you to store data from structured, semi-structured, and unstructured documents within the database. For more information, refer to the link: Microsoft SQL Server 2008.
Microsoft Web Deploy 2.0	Web Deploy (Web Deployment Tool) simplifies the migration, management and deployment of IIS Web servers, Web applications and Web sites. For more information, refer to the link: Web Deploy.
MySQL Connector/Net 6.2.3	Connector/Net is a fully-managed ADO.NET driver for MySQL. For more information, refer to the link: MySQL Download Connector/Net.
PHP Driver for SQL Server	The SQL Server Driver for PHP v1.1 is designed to enable reliable, scalable integration with SQL Server for PHP applications deployed on the Windows platform. For more information, refer to the link: SQL Server Driver for PHP.
PHP	PHP is a general-purpose scripting language that can be embedded into HTML and is especially suited for Web development. For more information, refer to the link: PHP.
SQL Server 2008 Management Objects	The SQL Server Management Objects (SMO) is a .NET Framework object model that enables software developers to create client-side applications to manage and administer SQL Server objects and services. For more information, refer to the link: Microsoft TechNet SQL Server Management Objects (SMO).
URL Rewrite 2.0	IIS URL Rewrite 2.0 enables Web administrators to create powerful rules to implement URLs that are easier for users to remember and easier for search engines to find. For more information, refer to the link: URL Rewrite.
Windows Cache 1.1 for PHP	Windows Cache Extension for PHP is a PHP accelerator that is used to increase the speed of PHP applications running on Windows and Windows Server. For more information, refer to the link: Windows Cache Extension for PHP.

Related Information Links

- [Web Deployment Tool (MS Deploy) Forum](#)
- [PHP Manager for IIS 7 Community](#)
- [Configure and Optimize the Microsoft Web Platform for PHP Applications](#)