

Migrating to the Bing Search API

September 2012

# Introduction

With the Bing Search API, you can use data that is collected by the Bing search engine in your desktop application, web application, or other software component. You can add search functionality to a website, create unique consumer or enterprise applications, or develop new mashups. Using the Bing Search API, you can access web, image, news, and video results, as well as related search and spelling suggestions.

The Bing Search API is now offered on a subscription basis through the [Windows Azure Marketplace](https://datamarket.azure.com/), a cloud-based data service. With the Windows Azure Marketplace, you can subscribe to various data sets and web services. From there, you can integrate the information you need into your applications via standardized data services. You can also analyze your data online by using the Service Explorer tool available in the Marketplace or by using tools like Microsoft SQL Server PowerPivot for Microsoft Excel.

The new version of the Bing Search API includes:

* Metered subscription of query limits.
* HTTPS service URIs that provide results in either XML or JSON formats.
* Open Data Protocol (OData) support for easy consumption across multiple development systems.
* Improved support for data types.
* Ability to monetize applications in the Windows Azure Marketplace.
* Access to fresher results and improved relevance.

To make the move to Windows Azure Marketplace, you’ll need to follow the migration process described in this guide.

# Migrating to the Bing Search API in Windows Azure Marketplace

To migrate your existing application from Bing API 2.0 to the Bing Search API in Windows Azure Marketplace, complete the steps in this section.

You can expect migrating to the new Bing Search API to involve the following changes in your existing application:

* Targeting of a new service URI
* Moderate changes to the request and response schemas
* Security requirement to authenticate access to your data subscription

## 1. Subscribe to a Bing Search API offering

With the Bing Search API, you can subscribe to one of the following subscription types.

|  |  |
| --- | --- |
| **Subscription type** | **Capabilities** |
| [Bing Search API](https://datamarket.azure.com/dataset/5BA839F1-12CE-4CCE-BF57-A49D98D29A44) | Gives you access to Bing web, image, video, news, related search results, and spelling suggestions. |
| [Bing Search API - Web Results Only](https://datamarket.azure.com/dataset/8818F55E-2FE5-4CE3-A617-0B8BA8419F65) | Gives you access only to Bing web results. |

With both subscription types, you can select a service level to meet the needs of your organization. Service levels are based on a monthly maximum of transactions. The lowest level (5000 transactions per month) is free.

## 2. Change your service root URI

Use the service root URI in the following table that corresponds to the subscription type you’re using.

|  |  |
| --- | --- |
| **Subscription type** | **Service root URI** |
| Bing Search API | <https://api.datamarket.azure.com/Bing/Search/> |
| Bing Search API - Web Results Only | <https://api.datamarket.azure.com/Bing/SearchWeb/> |

You can experiment making search requests with the Bing Search API by using the Service Explorer tool in Windows Azure Marketplace. You can access Service Explorer by using one of the following links, depending on which subscription type you’re using.

|  |  |
| --- | --- |
| **Subscription type** | **Service Explorer link** |
| Bing Search API | <https://datamarket.azure.com/dataset/explore/5BA839F1-12CE-4CCE-BF57-A49D98D29A44> |
| Bing Search API - Web Results Only | <https://datamarket.azure.com/dataset/explore/8818F55E-2FE5-4CE3-A617-0B8BA8419F65> |

The following illustration shows an example of developing and running a query in Service Explorer.



The full service URI is expressed using elements of the OData specification.

Here is what’s different in the Bing Search API from Bing Search API 2.0:

* You must now place your query text (and other string parameters) inside single quotation marks. Because single quotation marks need to be URL-encoded before being used in a web request, you’ll want to use the encoded equivalent (%27) instead. For instance:

https://api.datamarket.azure.com/Bing/SearchWeb/Web?Query=%27Xbox%27

It is best to URL-encode the entire query text, including the apostrophes that surround it. For example:

https://api.datamarket.azure.com/Bing/Search/Web?Query=%27New+Xbox+Games%27

This way, if you’ve got any characters in your query that require URL encoding (such as spaces), they’ll be properly handled as well.

* The names of the count and offset parameters have been changed to comply with the OData standard. Also, the way you specify which response format (JSON or XML) that you want has changed. The following table shows the changes.

| **Bing Search API Parameter** | **Description** | **Bing Search API 2.0 equivalent** |
| --- | --- | --- |
| $top | Specifies the number of results to return. The default is 50 for Web, Image, and Video searches, 15 for News. | &count= |
| Example:  https://api.datamarket.azure.com/Bing/Search/Web?Query=%27Xbox%27&$top=10 | | |
| $skip | Specifies the offset requested for the starting point of results . The default is zero. | &offset= |
| Example:  https://api.datamarket.azure.com/Bing/Search/Web?Query=%27Xbox%27&skip=20 | | |
| $format | Specifies the format of the OData response. Current options are Atom (for XML) or JSON. Default: Atom. | You receive different formats by targeting different ASPX files, that is, xml.aspx or json.aspx. |
| Example:  https://api.datamarket.azure.com/Bing/Search/Web?Query='Xbox'&$format=json | | |

For more information about OData, visit the [OData website](http://www.odata.org).

**Note:**

* + You don’t need to enclose the values of the OData parameters in single quotation marks ('), nor URL-encode them. However, you do need to do this for all other string parameter values such as *Market*, *Sources*, *ImageFilters*, and *Options*.
  + The Bing Search API supports HTTP compression to reduce transfer size. To enable HTTP compression, add header “*Accept-Encoding: gzip*” to your HTTP request.

## 3. Append the service operation to the end of the service root URI

A service operation is the data source that is targeted for the Bing Search API results. Bing Search API 2.0 references this as the *&Source=* parameter.

To target a particular service operation, append it to your service root URI. For instance, if you want image results, append the service operation *Image* to your service URI as follows:

https://api.datamarket.azure.com/Bing/Search/Image

The service operations available in the Bing Search API Marketplace are listed in the following table. Service operations are case-sensitive.

| **Service operation** | **Bing Search API 2.0 equivalent** |
| --- | --- |
| Web | &Sources=Web |
| Example:  https://api.datamarket.azure.com/Bing/Search/Web?Query=%27Xbox%27 | |
| Image | &Sources=Image |
| Example:  https://api.datamarket.azure.com/Bing/Search/Image?Query=%27Xbox%27 | |
| Video | &Sources=Video |
| Example:  https://api.datamarket.azure.com/Bing/Search/Video?Query=%27Xbox%27 | |
| News | &Sources=News |
| Example:  https://api.datamarket.azure.com/Bing/Search/News?Query=%27Xbox%27 | |
| SpellingSuggestion | &Sources=Spell |
| Example:  https://api.datamarket.azure.com/Bing/Search/SpellingSuggestion?Query=%27Xbox%27 | |
| RelatedSearch | &Sources=RelatedSearch |
| Example:  https://api.datamarket.azure.com/Bing/Search/RelatedSearch?Query=%27Xbox%27 | |
| Composite | Concatenation of values for the Sources parameter. Example: &Sources=Web+Image |
| Example:  https://api.datamarket.azure.com/Bing/Search/Composite?Sources=%27Web%2BNews%27&Query=%27Xbox%27 | |

**Multiple data sources:** To target multiple data sources in a single query, use the *Composite* service operation followed by its required parameter *Sources*. Make the value of *Sources* the names of the service operations concatenated with a plus (+) character. For example, to get results about Xbox from both web and news sources simultaneously:

https://api.datamarket.azure.com/Bing/Search/Composite?Sources=%27Web%2BNews%27&Query=%27Xbox%27

**Exception:** When you request a spelling suggestion as part of a composite query, use the service operation *Spell* instead of *SpellingSuggestion*. Example:

https://api.datamarket.azure.com/Bing/Search/Composite?Sources=%27Web%2BSpell%27 &Query=%27San+Fernado%27

Note that the number of results returned with the OData reserved parameter $top refers to all data sources specified in the query except News. News will always return a fixed number of 15 news results when you use it in a composite query.

## 4. Specify required and optional parameters

You can append parameterized name/value pairs to the service URI, as in common REST-like interfaces.

For instance, if you're building a query with the required parameter *Query* of value 'Xbox' and the optional parameter *Market* of value 'en-GB', append the parameterized name/value pairs to your service URI as follows:

https://api.datamarket.azure.com/Bing/SearchWeb/Web?Query=%27Xbox%27&Market=%27en-GB%27

You can view a table of parameters under the **Details** tab of the subscription type you're using.

* [Schema for Bing Search API](https://datamarket.azure.com/dataset/5BA839F1-12CE-4CCE-BF57-A49D98D29A44#schema)
* [Schema for Bing Search API – Web Results Only](https://datamarket.azure.com/dataset/8818F55E-2FE5-4CE3-A617-0B8BA8419F65#schema)

## 5. Obtain your account key

To authenticate a Bing Search API request with Windows Azure Marketplace, you’ll need to get an account key. The account key replaces the application ID as the means of authentication.

When you create an account in the Windows Azure Marketplace, you get a default account key, which you can’t change or remove. However, you can create other account keys.

It’s best to use an account key that you have created instead of the default one. That way, if your key is compromised, you can remove it and create another one in its place. You can manage your account keys on the [Azure Account Keys](https://datamarket.azure.com/account/keys) page.

## 6. Test the service URI in a browser

You can try a service URI by simply pasting it into the address bar of your browser. The Bing Search API supports Basic Authentication so you'll be prompted for a user name and password. Leave the user-name field empty and copy your account key into the password field. Your browser will display the results from the Bing Search API, or prompt you to save them depending on your browser and how it’s configured.

## 7. Adjust your response parsing logic

Adjust your response parsing logic to match the updated response format.

The following sections compare query responses between Bing Search API 2.0 and the Bing Search API that's available through Windows Azure Marketplace.

### Results container

Some changes were made to the elements that contain the results array. This includes the **SearchResponse** element, the **SearchResponse/Query** element, and the immediate children of the source-type-specific containers like **web:Web** and **mms:Image**.

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=web&count=2* | *https://api.datamarket.azure.com/Bing/Search/****Web****?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - Query |  | | - ~~SearchTerms~~ |  | |  | - link | | - web:Web | - entry | | - ~~web:Total~~ |  | | - ~~web:Offset~~ |  | | - web:Results | - content |   Note: Other properties are omitted for clarity. | |
| Removed data:   * SearchTerms. Because this is sent across the wire in the request, it is not necessary in the response. If some asynchronous programming pattern requires that the response be verified, the id response property can be used to obtain the full request URL. * Total. If there are no more pages of results, the /feed/link[@rel=’next’] property will no longer appear, and the Total value is no longer needed for enabling paging. * Offset. This value is used to help you build the request URL for the next page of results. In the OData protocol, the next page of results is requested via the href property of /feed/link[@rel=’next’]. The next page of results can be obtained simply by requesting this URL. | |

### Web results

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=web&count=2* | *https://api.datamarket.azure.com/Bing/Search/****Web****?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - web:Web |  | | - web:Results |  | | - web:WebResult | - entry | |  | - content | |  | - m:properties | |  | - d:ID | | - web:Title | - d:Title | | - web:Description | - d:Description | | - web:DisplayURL | - d:DisplayUrl | | - web:URL | - d:Url | | - ~~web:CacheUrl~~ |  | | - ~~web:DateTime~~ |  | | - ~~web:DeepLinks~~ |  | | - ~~web:SearchTags~~ |  |   Note: Other properties are omitted for clarity. | |
| Removed data:   * CacheUrl. Deprecated. * DateTime. Deprecated. * DeepLinks. Deprecated. * SearchTags. Deprecated.   Added data:   * ID. A result set/unique ID for this row.   Renamed data:   * URL. This property was renamed to Url. * DisplayURL. This property was renamed to DisplayUrl. | |

### Image results

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=image&count=2* | *https://api.datamarket.azure.com/Bing/Search/****Image****?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - mms:Image |  | | - mms:Results |  | | - mms:ImageResult | - entry | |  | - content | |  | - m:properties | |  | - d:ID | | - mms:Title | - d:Title | | - mms:MediaUrl | - d:MediaUrl | | - mms:Url | - d:SourceUrl | | - mms:DisplayUrl | - d:DisplayUrl | | - mms:Width | - d:Width | | - mms:Height | - d:Height | | - mms:FileSize | - d:FileSize | | - mms:ContentType | - d:ContentType | | - mms:Thumbnail | - d:Thumbnail | | - mms:Url | - d:MediaUrl | | - mms:ContentType | - d:ContentType | | - mms:Width | - d:Width | | - mms:Height | - d:Height | | - mms:FileSize | - d:FileSize |   Note: Other properties are omitted for clarity. | |
| Added data:   * ID. A result set/unique ID for this row.   Renamed data:   * URL. This property was renamed to SourceUrl. | |

### Video results

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=image&count=2* | *https://api.datamarket.azure.com/Bing/Search/****Image****?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - mms:Image |  | | - mms:Results |  | | - mms:VideoResult | - entry | |  | - content | |  | - m:properties | |  | - d:ID | | - mms:Title | - d:Title | | - mms:PlayUrl | - d:MediaUrl | | - ~~mms:SourceTitle~~ |  | | - mms:RunTime | - d:RunTime | | - mms:ClickThroughPageUrl | - d:DisplayUrl | | - mms:StaticThumbnail | - d:Thumbnail | | - mms:Url | - d:MediaUrl | | - mms:ContentType | - d:ContentType | | - mms:Width | - d:Width | | - mms:Height | - d:Height | | - mms:FileSize | - d:FileSize |   Note: Other properties are omitted for clarity. | |
| Removed data:   * SourceTitle. Deprecated.   Added data:   * ID. A result set/unique ID for this row.   Renamed data:   * PlayUrl. This property was renamed to MediaUrl. * ClickThroughPageUrl. This property was renamed to DisplayUrl. * StaticThumbnail. This property was renamed to Thumbnail. * StaticThumbnail.Url. This property was renamed to MediaUrl. | |

### News results

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=news&count=2* | *https://api.datamarket.azure.com/Bing/Search/****News****?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - news:News |  | | - news:Results |  | | - news:NewsResult | - entry | |  | - content | |  | - m:properties | |  | - d:ID | | - news:Title | - d:Title | | - news:Url | - d:Url | | - news:Source | - d:Source | | - news:Snippet | - d:Description | | - news:Date | - d:Date | | - ~~news:BreakingNews~~ |  |   Note: Other properties are omitted for clarity. | |
| Removed data:   * BreakingNews. Deprecated.   Added data:   * ID. A result set/unique ID for this row.   Renamed data:   * Snippet. This property was renamed to Description. | |

### Related search results

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=RelatedSearch&count=2* | *https://api.datamarket.azure.com/Bing/Search/ RelatedSearch?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - rs:RelatedSearch |  | | - rs:Results |  | | - rs:RelatedSearchResult | - entry | |  | - content | |  | - m:properties | |  | - d:ID | | - rs:Title | - d:Title | | - rs:Url | - d:BingUrl |   Note: Other properties are omitted for clarity. | |
| Added data:   * ID. A result set/unique ID for this row. | |

### Spelling suggestion results

|  |  |
| --- | --- |
| Bing Search API 2.0 | Bing Search API Marketplace |
| Requests  *http://api.search.live.net/xml.aspx?Appid=App&query=odata&sources=spell&count=2* | *https://api.datamarket.azure.com/Bing/Search/****SpellingSuggestions****?Query=%27****odata****%27&$top=****2*** |
| Response Trees   |  |  | | --- | --- | | SearchResponse | Feed | | - spl:Spell |  | | - spl:Results |  | | - spl:SpellResult | - entry | |  | - content | |  | - m:properties | |  | - d:ID | | - spl:Value | - d:Value |   Note: Other properties are omitted for clarity. | |
| Added data:   * ID. A result set/unique ID for this row. | |

# Usage samples

This section shows usage samples in a variety of languages to help you start using the new Bing Search API.

## Using the Bing Search API with the Microsoft .NET Framework in C#

The most straightforward approach to using the Bing Search API with C# is to download a service proxy from the Bing API landing page in Windows Azure Marketplace.

To prepare to use the Bing Search API with C#

1. Subscribe to one of the Bing Search API Marketplace offers.
2. Click **.NET C# Class Library** directly below the offer variants list to download the proxy to your computer.
3. Move the library into your project directory and then add it to your project.
4. Review the first two sections of [Walkthrough: Microsoft Translator in a C# Console Application](http://code.msdn.microsoft.com/Walkthrough-Translator-in-7e0be0f7) to get started with proxy.

After the proxy has been added to your project, you can start coding against it immediately. The following sample console application synchronously performs an image search and downloads the search results.

**Important:** Be sure to add a reference to **System.Data.Services.Client** before you build your project.

|  |
| --- |
| using System;  using System.Net;    namespace ImageSearchConsole  {      class Program      {          static void Main(string[] args)          {              var bingContainer = new Bing.BingSearchContainer(  new Uri("https://api.datamarket.azure.com/Bing/Search/"));                // Replace this value with your account key.              var accountKey = "YourAccountKey";                // Configures bingContainer to use your credentials.              bingContainer.Credentials = new NetworkCredential(accountKey, accountKey);                // Build the query.  var imageQuery =  bingContainer.Image("xbox", null, market, null, null, null, null);              var imageResults = imageQuery.Execute();                foreach (var result in imageResults)              {                  Console.WriteLine(result.Title);              }                Console.ReadLine();          }      }  } |

## Using the API with the Microsoft .NET Framework in Visual Basic

Because of the popularity of working with XML content in Visual Basic, the following sample has been tailored to consume the raw Atom response from the Bing Search API directly in Visual Basic .NET. The sample runs a simple image query against the Bing Search API and prints some attribute of each image response to the console. To use this sample, create a new Visual Basic console application and paste the sample code into the main .vb file.

|  |
| --- |
| Imports System.Xml  Imports System.Net  Module Module1      Sub Main()          ' Replace this value with your account key          Dim accountKey As String = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="          ' The Windows Azure Marketplace Service Root Url for the Bing Search offer          Dim serviceRoot As String = "[https://api.datamarket.azure.com/Bing/Search/](https://dallastestapi.cloudapp.net/Data.ashx/Bing/Search/)"          ' The Query type we're performing          Dim imageQueryRoot As String = serviceRoot + "Image?"          ' The Query string, surrounded by URL-encoded quotation marks          Dim imageQuery As String = imageQueryRoot + "Query=%27xbox%27"          ' An XML document we will use to read the results          Dim document As XmlDocument = New XmlDocument()          ' The next four lines configure XmlDocument to use your credentials          ' when querying the Image Search API          Dim accountCredential As New NetworkCredential(accountKey, accountKey)          Dim resolver As New XmlUrlResolver()          resolver.Credentials = accountCredential          document.XmlResolver = resolver          ' With credentials configured we download the query results          document.Load(imageQuery)          ' This namespace manager will assist with parsing the results          Dim namespaceManager As XmlNamespaceManager =  New XmlNamespaceManager(document.NameTable)          namespaceManager.AddNamespace("atom", "<http://www.w3.org/2005/Atom>")          namespaceManager.AddNamespace("m",  "<http://schemas.microsoft.com/ado/2007/08/dataservices/metadata>")          namespaceManager.AddNamespace("d",  "<http://schemas.microsoft.com/ado/2007/08/dataservices>")          ' This query gets the "Next" link for the result.  ' Set document. Load to this value to get the next          ' set of results. If the value is Nothing then there are no more results.          Dim nextResultSet As String =  document.SelectSingleNode(  "/atom:feed/atom:link[@rel='next']/@href",  namespaceManager).Value          ' This gets the properties node for each Image search result          Dim imageResults As XmlNodeList =  document.SelectNodes(  "/atom:feed/atom:entry/atom:content/m:properties",  namespaceManager)          For Each imageResult As XmlNode In imageResults            ' Select the Title element from this imageResult             Dim title As String =  imageResult.SelectSingleNode(".//d:Title", namespaceManager).InnerText             Console.WriteLine(title)          Next      End Sub  End Module |

## Using the Bing Search API with PHP

PHP has a powerful JSON parsing mechanism. Because PHP is a dynamic language, the parsing mechanism enables PHP developers to program against a JSON object graph in a straightforward way. The following sample shows a PHP page that sends a request to the JSON interface by using the **file\_get\_contents** function to call the JSON query URL, and the **json\_decode** function to turn the results into an object graph that can be walked and turned into HTML.

|  |
| --- |
| <html>  <head>  <title>PHP Bing</title>  </head>  <body>  <form method="post" action="<?php echo $\_SERVER['PHP\_SELF'];?>">  Type in a search:    <input type="text" id="searchText" name="searchText"  value="<?php  if (isset($\_POST['searchText']))  {  echo($\_POST['searchText']);  }  else  {  echo('sushi');  }  ?>"  />  <input type="submit" value="Search!" name="submit" id="searchButton" />  <?php  if (isset($\_POST['submit']))  {  // Replace this value with your account key  $accountKey = 'YourAccountKey';  $ServiceRootURL = ‘https://api.datamarket.azure.com/Bing/Search/';  $WebSearchURL = $ServiceRootURL . 'Image?$format=json&Query=';  $context = stream\_context\_create(array(  'http' => array(  'request\_fulluri' => true,  'header' => "Authorization: Basic " .  base64\_encode($accountKey . ":" . $accountKey)  )  ));  // Encode the query and the single quotation marks that must surround it.  $request = $WebSearchURL . urlencode("'{$\_POST['searchText']}'");  echo($request);  $response = file\_get\_contents($request, 0, $context);  $jsonobj = json\_decode($response);  echo('<ul ID="resultList">');  foreach($jsonobj->d->results as $value)  {  echo('<li class="resultlistitem"><a href="' . $value->MediaURL . '">');  echo('<img src="' . $value->Thumbnail->MediaUrl. '"></li>');  }  echo("</ul>");  }  ?>  </form>  </body>  </html> |

## Using the Bing Search API with other development environments

Windows Azure Marketplace exposes the Bing Search API by using the well-known and broadly supported OData protocol. In addition to providing a common language with which to expose data, OData provides Bing Search API customers with a broad set of tools and libraries that they can use to consume OData services like the Bing Search API in a variety of programming languages. To learn more about the library support available for OData, see the [OData SDK](http://www.odata.org/developers/odata-sdk).

To browse the Bing Search API OData services, you can query the OData metadata, which returns a complete description of how the service can be called, what parameters it expects, and what data structures it returns. It thus serves a roughly similar purpose as a method signature in a programming language. To access the OData metadata, append the OData keyword **$metadata** to the end of the query URL of the Bing Search API data subscription that you selected, as shown in the following table.

| **Product offering** | **Query URL** |
| --- | --- |
| Bing Search API | <https://api.datamarket.azure.com/Bing/Search/$metadata> |
| Bing Search API - Web Results Only | <https://api.datamarket.azure.com/Bing/SearchWeb/$metadata> |

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2012 Microsoft. All rights reserved.