Efficient Voice User Interface System Using VoiceXML and ASP.NET 2.0

Byung-Seok Kang[1] and Gi-Jong Yoo[2]

[1] Department of Electronics Engineering, Korea University
1, 5-ga, Anam-dong, Sungbuk-gu, 136-701, Seoul, Korea
kbsgasu@korea.ac.kr
[2] Graduate School of Education, Ajou University
San 5, Wonchon-dong, Yeongtong-gu, 443-749, Suwon, Kyonggi-do, Korea
mathink@naver.com

Abstract. The web-based application by VoiceXML service on the Internet is gradually being accepted for the human-machine interaction because it provides the speech-enabled function and makes telephone access a reality. Many companies are interested in building the effective dynamic Voice User Interface (VUI) system into the architecture of the already existing web application. The previous papers [1, 2] suggest that they demonstrate how to design and implement using VoiceXML and Active Server Pages. However, they have used only one server script language, so it is not efficient. For that reason, we have built another design that is more efficient for VoiceXML. Experimental results demonstrate that ASP.NET 2.0 shows the highest communication success rate and the lowest response time for web surfing.

1 Introduction

Today the computer distinguishes itself as a key player in the everyday human activity, be it business, research, engineering, or entertainment. The invention of the World Wide Web gave the computer even greater importance. By bringing the entire globe under its orb, the World Wide Web opened before us a new world characterized by the use of computer applications in every field. Voice and web technologies assumed a definite shape in the last decade of the 20th century and soon sparked a series of unprecedented changes in the way people interact long distance. There are a host of competent technologies, such as VoiceXML 2.0, that facilitate fast and accurate transfer of information all over the world. Furthermore, the Microsoft new computer language Active Server Page.NET 2.0 is powerful for designing the VUI system.

Many companies and personal users are using VXML for their customer web service. D. Mecanovic proposes the Voice User Interface Design for a telephone application [1]. They proved that the long and descriptive prompts make navigation difficult and female text-to-speech (TTS) voice is preferred in dynamic VUI. R. Vankayala and H. Shi [2], in their paper "Dynamic Voice Use Interface using VXML and Active Server Pages" implemented and demonstrated an existing e-Commerce

M. Xu et al. (Eds.): APPT 2007, LNCS 4847, pp. 608 – 616, 2007.

web application by using the BeVocal.com server and Microsoft IIS Web server. Furthermore M. Tsai [5] deploys a web-based Mandarin dialogue system, in which a user can use either a telephone channel or VoIP by personal computer to access the voice server. But no one suggested which programming language is the most efficient when it is used with VXML for existing web application services. In this paper, we perform comprehensive experiments to find out the best implementation tool to build an efficient VUI system. Specifically, we consider four tools: ASP, ASP.NET 2.0, JSP, and PHP. Experimental results demonstrate that ASP.NET 2.0 has the highest communication success rate and the lowest response time. This paper is organized as follows. Section 2 discusses some background information. Next, the design utilizing VoiceXML with some famous web programming languages is presented in section 3. Simulation results are presented in Section 4 to show that our proposed design can provide a good performance in the communication success rate and reduce the response time. Finally, Section 5 concludes the paper.

2 Background

2.1 VoiceXML System

While HTML assumes a graphical web browser with display, keyboard, and mouse, VoiceXML assumes a voice browser with audio output, audio input, and keypad input. Audio input is handled by the voice browser's speech recognizer. Audio output consists both of recordings and speech synthesized by the voice browser's text-to-speech system.

A voice browser typically runs on a specialized voice gateway node that is connected both to the Internet and to the public switched telephone network (see Figure 1). The voice gateway can support hundreds or thousands of simultaneous callers, and can be accessed by any one of the world's estimated 1,500,000,000 phones, ranging from antique black candlestick phones up to the very latest mobiles.

The user interacts with a Web site over the phone using a VoiceXML Browser, which is hosted on a Gateway. Instead of rendering and interpreting HTML, the VoiceXML Browser renders and interprets VoiceXML. The Gateway is the key

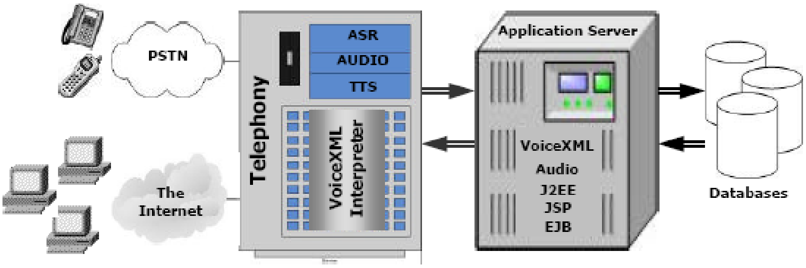


Fig. 1. VoiceXML serving architecture

bridge technology, responsible for VoiceXML Browser, ASR Resource, TTS Resource, Telephony Resource, Audio Resource and TCP/IP Resource.

2.2 ASP.NET 2.0

Active Server Pages.NET (ASP.NET) is a web development technology from Microsoft. Part of the .NET Framework, ASP.NET allows developers to build dynamic web applications and web services using compiled languages like VB.NET and C#. Using Visual Studio, the development tool from Microsoft, web developers can develop very compelling applications using ASP.NET, with the ease of drag-and-drop server controls. Currently in its next major release, ASP.NET 2.0 is slated to be released in November 2005.

ASP.NET 2.0 is a compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET 2.0 can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.

Two aspects of ASP.Net 2.0 makes it fast (compiled code and caching). In the past, the code was interpreted into "machine language" when website visitor viewed web page. Now, with ASP.Net 2.0 the code is compiled into "machine language" before visitor ever comes to web site. Caching is the storage of information that will be reused in a memory location for faster access in the future. ASP.Net 2.0 allows programmers to set up pages or areas of pages that are commonly reused to be cached for a set period of time to improve the performance of web applications. In addition, ASP.Net 2.0 allows the caching of data from a database so the website isn't slowed down by frequent visits to a database when the data doesn't change very often.

3 Design and Implementation Using VoiceXML and ASP.NET 2.0

This section gives the main implementation details of our site. We describe our algorithm and four designed systems.

3.1 System Architecture

To build VUI systems, we use VoiceXML café, “BeVocal.com” [6] for a VoiceXML server. The BeVocal Café is a world-class, web-based development environment that provides all the tools and resources developers need to create their own innovative speech applications.

In our VUI systems, two web servers (Microsoft IIS and Apache web server), two operating systems (Windows 2003 server and Linux server), four web programming languages (ASP, ASP.NET 2.0, JSP, and PHP), and two database systems (MS-SQL 2005 and MySQL 5.0) are used. Figure 2 shows the VUI system architecture.

3.2 Algorithm of the Web Surfing System

To demonstrate the behavior of VUI systems, the following application scenario is adopted. Once a user calls a VUI homepage, the VUI system says “Welcome to my homepage. Which do you want? Notice, Free board, Public data or On-line poll?” If

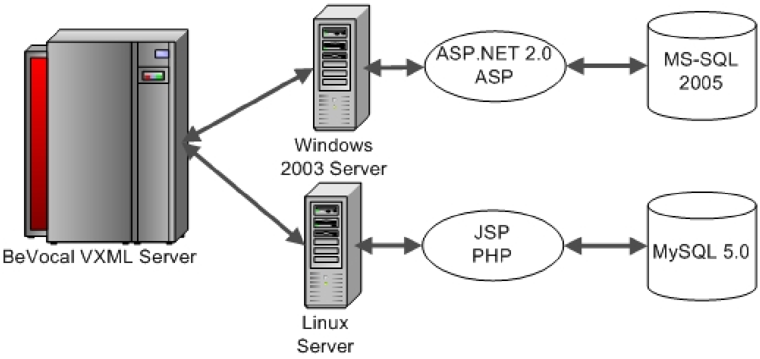


Fig. 2. VUI architecture

the user selects a free board (b), the system says “Do you want to write or not?”. If the user chooses “yes”, the system shows a “write article” page; otherwise, the system asks you a next question. Detailed application scenario is illustrated in Figure 3.

The source code which is below is the detail for main page in Figure 3’s application scenario [7]. During some system design we describe a part of source code under ASP.NET 2.0.

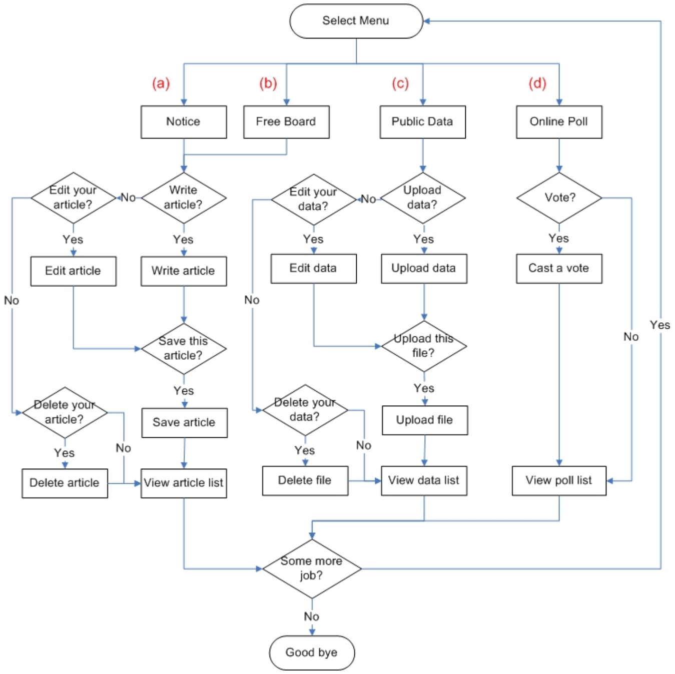


Fig. 3. Application scenario in VUI systems

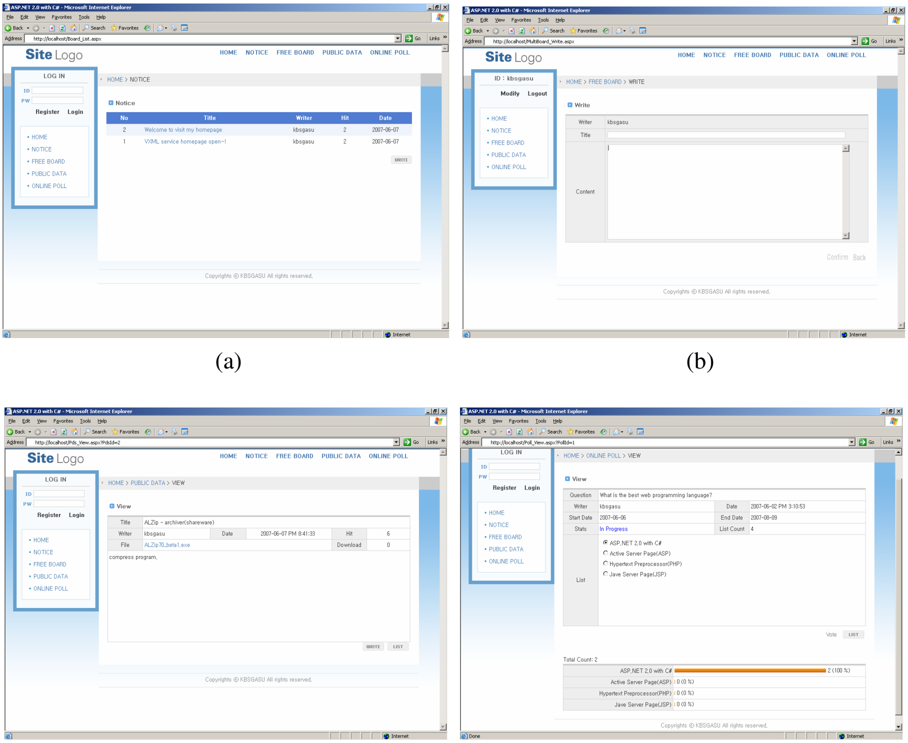
Sample Source Code of Main Page

```
<%@ Page Language="C#" MasterPageFile="~/Default.master"
Title="Simulation no.4 - ASP.NET 2.0 with C#" %>

<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
<form id="main" scope="dialog">
  <prompt bargein="true">
    Welcome to visit my homepage. Which do you want?
  </prompt>
  <grammar> Notice | Free Board | Public Data | Survey </grammar>
  <noinput>
    No response, say one more please.
    <reprompt/>
  </noinput>
  <noinput count="4">
    Disconnect the phone.
  <disconnect/>
  </noinput>
  <nomatch>
    Say one more please.
  <reprompt/>
  </nomatch>
  <filled namelist="user_input" mode="all">
  <if cond="user_input == 'Notice'">
    <goto next="/List.aspx?TblName=Notice" fetchhint="safe"/>
  <elseif cond="user_input == 'Free Board'"/>
    <goto next="/List.aspx?TblName=Notice" fetchhint="safe"/>
  <elseif cond="user_input == 'Public Data'"/>
    <goto next="/List.aspx?TblName=Notice" fetchhint="safe"/>
  <elseif cond="user_input == 'Survey'"/>
    <goto next="/Poll_List.aspx" fetchhint="safe"/>
  <else/>
    <disconnect/>
  </if>
  </filled>
</form>
</VXML>
```

If users request the web service through voice interface, the BeVocal server should interpret the voice and connect with this site for response. Then the web server shows the correct web page and waits for the next request. Figure 4 shows some sample web pages.

First, in figure 4(*a*), visitors view a notice board. Second, Figure 4(*b*) shows a guest write the free board. Third, figure 4(*c*), users connect to the public data for download some useful data. Finally in figure 4(*d*), users view the result of online poll page.



ASP.NET 2.0 with C# - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Board_List.aspx
Go Links
Site Logo
HOME NOTICE FREE BOARD PUBLIC DATA ONLINE POLL
LOG IN
ID
PW
Register Login
• HOME
• NOTICE
• FREE BOARD
• PUBLIC DATA
• ONLINE POLL
HOME > NOTICE
WRITE
Copyrights © KBSGASU All rights reserved.
Internet

Notice

No	Title	Writer	Hit	Date
2	Welcome to visit my homepage	kbsgasu	2	2007-06-07
1	VXML service homepage open~!	kbsgasu	2	2007-06-07

(a)

ASP.NET 2.0 with C# - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/MultiBoard_Write.aspx
Go Links
Site Logo
HOME NOTICE FREE BOARD PUBLIC DATA ONLINE POLL
ID : kbsgasu
Modify Logout
• HOME
• NOTICE
• FREE BOARD
• PUBLIC DATA
• ONLINE POLL
HOME > FREE BOARD > WRITE
Confirm Back
Copyrights © KBSGASU All rights reserved.
Internet

Write

Writer	kbsgasu
Title	
Content	

(b)

ASP.NET 2.0 with C# - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Pds_View.aspx?PdsId=2
Go Links
Site Logo
HOME NOTICE FREE BOARD PUBLIC DATA ONLINE POLL
LOG IN
ID
PW
Register Login
• HOME
• NOTICE
• FREE BOARD
• PUBLIC DATA
• ONLINE POLL
HOME > PUBLIC DATA > VIEW
WRITE LIST
Copyrights © KBSGASU All rights reserved.
Internet

View

Title	ALZip - archiver(shareware)				
Writer	kbsgasu	Date	2007-06-07 PM 8:41:33	Hit	6
File	ALZip70_beta1.exe			Download	0

compress program.

(c)

ASP.NET 2.0 with C# - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Poll_View.aspx?PollId=1
Go Links
LOG IN
ID
PW
Register Login
• HOME
• NOTICE
• FREE BOARD
• PUBLIC DATA
• ONLINE POLL
HOME > ONLINE POLL > VIEW
Vote LIST
Copyrights © KBSGASU All rights reserved.
Done
Internet

View

Question	What is the best web programming language?		
Writer	kbsgasu	Date	2007-06-02 PM 3:10:53
Start Date	2007-06-06	End Date	2007-08-09
Stats	In Progress	List Count	4
List	● ASP.NET 2.0 with C# ○ Active Server Page(ASP) ○ Hypertext Preprocessor(PHP) ○ Jave Server Page(JSP)		

Total Count: 2

ASP.NET 2.0 with C#	2 (100 %)
Active Server Page(ASP)	0 (0 %)
Hypertext Preprocessor(PHP)	0 (0 %)
Jave Server Page(JSP)	0 (0 %)

(d)

Fig. 4. (*a*) View notice list. (*b*) Write the free board. (*c*) Download some data. (*d*) The result of online poll.

4 Simulations

In the implemented VUI systems, we measure the average response time and the success rate. The average response time is the elapsed time to obtain the result page when a user requests a page, and the success rate is the probability that a correct page is obtained. Figure 5, 6 shows the average response time as the number of web pages surfed by users using VUI system or not. As you can assume, VUI is much more efficient than usual web application system in using circumstances (figure6). It is shown that ASP.NET 2.0 has the shortest response time and JSP and ASP exhibit longer response time than PHP and ASP.NET 2.0. This can be explained by two reasons: 1) JSP, ASP, and PHP interpret common language runtime codes whenever a user requests a page. On the other hand, ASP.NET 2.0 pre-compiles common language runtime codes before the user visits the web site, and thus it can display the result page immediately without any interpretation; 2) ASP.NET 2.0 employs a caching scheme to allocate frequently used code and data in main memory when a user visits a web site.

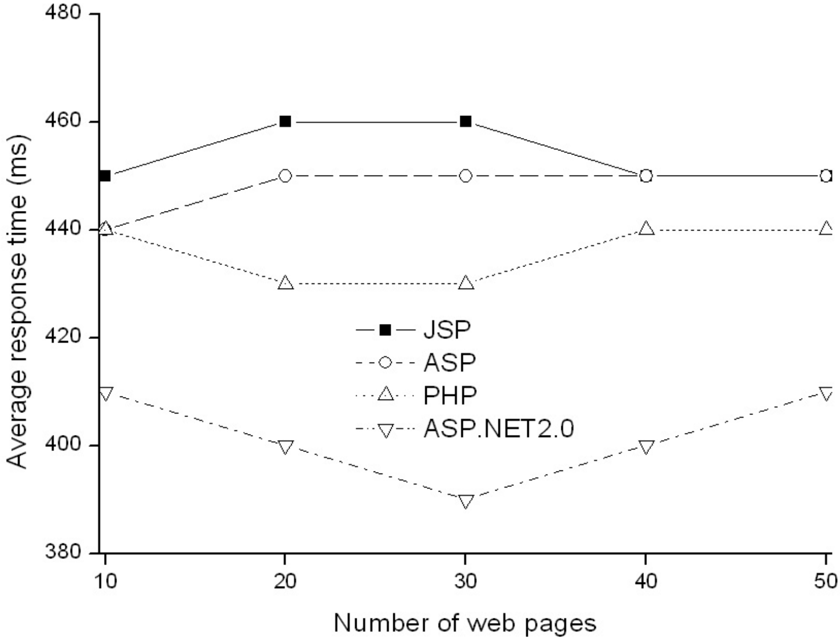


Fig. 5. Average response time not using VUI system

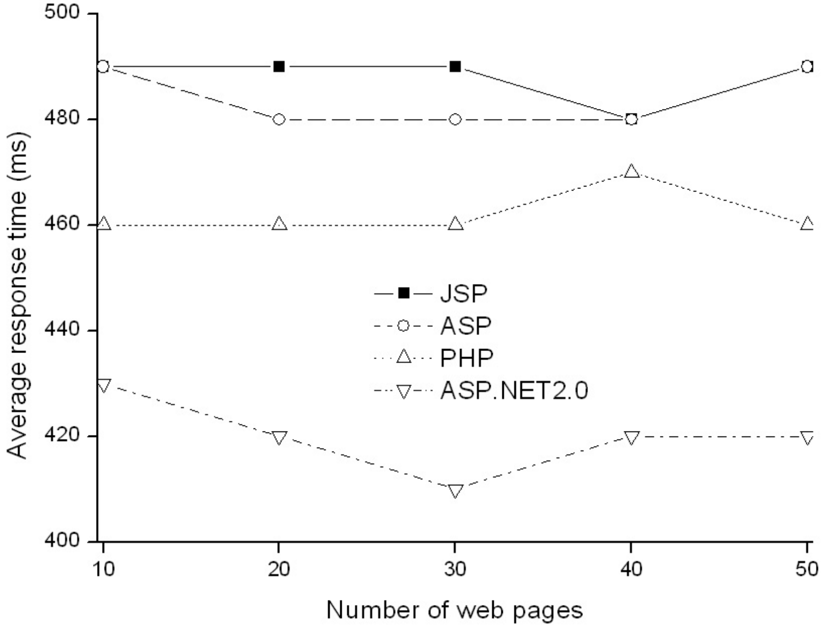


Fig. 6. Average response time using VUI system

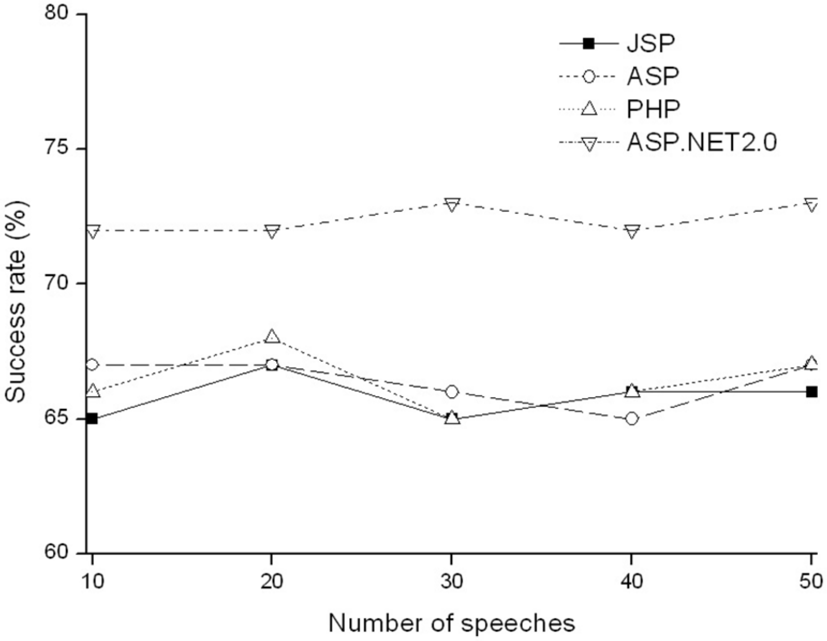


Fig. 7. Success rate

Figure 7 show the success rate as the number of speeches. It can be found that ASP.NET 2.0 has the highest success rate compared with other tools. All of these experimental results demonstrate that ASP.NET 2.0 is the most suitable programming language in VUI system.

5 Conclusion and Future work

This paper describes which web programming language reduces the response time with VXML. We design four other VXML systems. We also propose the guideline to use the VUI service without changing previous infra structure. Experimental results indicate that ASP.NET 2.0 can significantly reduce the average response time and provide higher success rate, compared with other tools, i.e., JSP, PHP, and ASP. In our future work, we will investigate VUI systems using VoiceXML 3.0 which is a new release from the W3C's voice browser working group [4].

References

1. Mecanovic, D., Shi, H.: Voice User Interface Design for a Telephone Application using VoiceXML. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 1058–1061. Springer, Heidelberg (2005)
2. Vankayala, R., Shi, H.: Dynamic Voice User Interface Using VoiceXML and Active Server Pages. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 16–18. Springer, Heidelberg (2006)

3. VoiceXML Forum (June 2007), http://www.voicexml.org
4. W3C Voice Brower Activity (June 2007), http://www.w3c.org/Voice
5. Tsai, M.-j.: The VoiceXML dialog system for the e-commerce ordering service. In: Proc. of the Ninth International Conference, May 24-26, 2005, vol. 1, pp. 95–100 (2005)
6. BeVocal Café, VoiceXML development environment (June 2007), http://www.bevocal.com
7. XML Web Services Created Using ASP.NET and XML Web Service Clients (June 2007), http://msdn2.microsoft.com/en-us/library/7bkzywba.aspx