

WebXone Development Guide

(2014-04-20)

AthTek Software

<http://www.athtek.com/webxone.html>

Directory

1. Overview	4
1.1 Objectives	4
1.2 Features.....	4
2. Instructions	4
2.1 Deploy environment of the Web server	4
2.2 Apply Delphi project to WebXone application.	6
2.3 Web access.....	9
2.4 Free trial to development mode	11
2.5 apply C++ Builder to develop WebXone application	12
2.6 apply Visual C++ to apply WebXone application	13
3. Support of the relevant system	14
3.1 The support of browsers.....	14
3.2 The support of the client-side operating system.....	15
3.3 support of development tools	15
3.4 support of the server operating system	15
3.5 support of the server service programs.....	15
3.6 Support of the database	15
3.7 Support of the middleware	15
4. Frequently Asked Questions	16
● Can registered user have access to further update?	16
● Is mobile platform enabled on client-side?	16
● Does the further update have development play supported by the mobile platform?.....	16
● Can it be deployed in Linux server?.....	16
● Does it support hundreds of people or more online together?	16
● Why does load fail and there is 126 error?	16
● Why cannot wxo be found when using IIS Web service?	16
● How to package many files by wxoConfig (including some configuration files and dynamic link library) ?.....	16
● How to pass parameters by URL?.....	17
● If the application is large, will run-up time be long because client-side has to load the application every time?.....	17
● Can WebXone have access to local port, and how about operating serial port or printer.....	17
● How to amend page title?	17
● How to close the page?	17
● Is MDI supported?	18
● How to shorten URL?	18
● Is it supported to customize directory installation?	18
● Is it supported to customize relevant information of setup.exe?.....	18
5. Recent update	18
6. Protocol	20

1.1 Free trial.....	20
1.2 Standard Edition	20
1.3 Enterprise Edition	20
Attachments:	21
1. Solutions to when WXP cannot be found when using IIS?.....	21
2. Win7 non-admin users close UAC.....	21
3. Brief introduction of the package deploy tool wxoConfig.....	21

1. Overview

WebXone is an advanced RIA-oriented (Rich Internet Applications) development environment. It is a new generation of Web application which combines the best UI of Desktop application, the popularity and fast, low-cost deployment of Web application and, real-time of interactive multimedia communication. It applies WebXone and can use common desktop application tools such as Delphi/C++Builder/VC/VB/PowerBuilder to develop web application system which is suitable for network.

WebXone is based on mature development platforms such as Delphi/C++Builder/VC/VB/PowerBuilder, and compatible with all the VCL components, and use unified system interface. It can quickly build a Web application system which can run efficiently.

WebXone enables Web development fast and efficient and enables web applications to run fast!

1.1 Objectives

It enables users to apply Delphi/C++Builder/VC/VB/PowerBuilder to achieve B/S programming easily.

1.2 Features

- Apply programming tools such as Delphi/C++Builder/VC/VB/PowerBuilder to develop web applications directly.
- Compatible with all the VCL components of Delphi/C++builder;
- Support to transform software project from C/S programs to B/S programs;
- High running efficiency which rivals C/S programs;
- Easy and quick Web deploying method;
- Compatible with all the mainstream browsers including IE(IE6~IE11), Chrome, FireFox, Safari, Opera, etc

2. Instructions

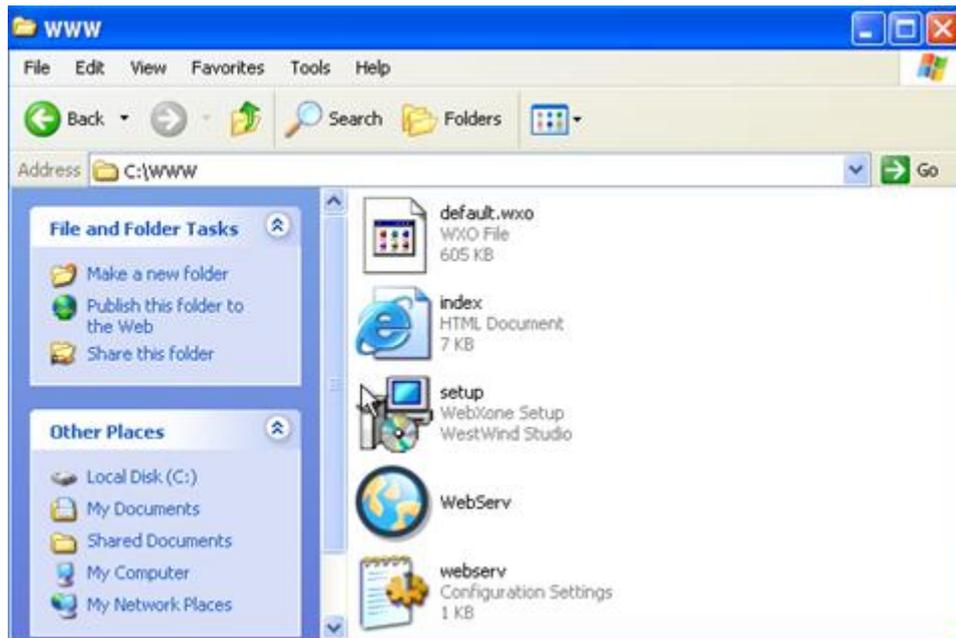
2.1 Deploy environment of the Web server

(1) Download SDK.

SDK download: <http://www.athtek.com/download/webxone.zip>

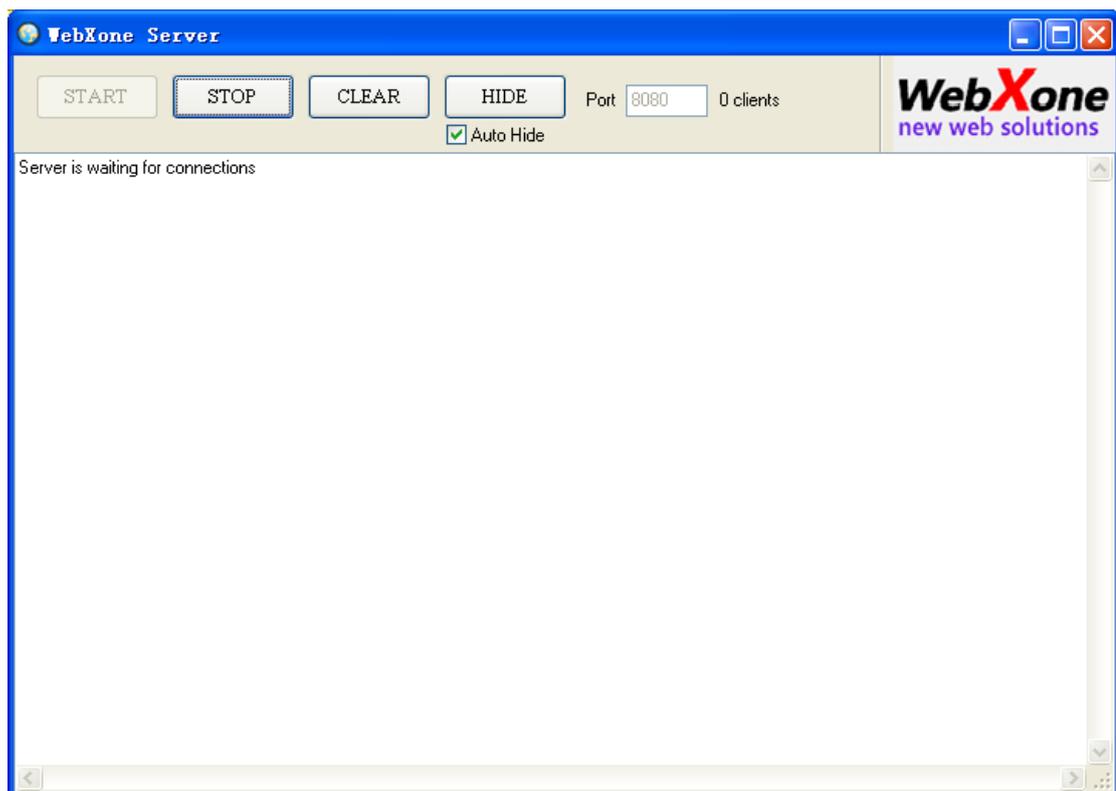
(2) Decompression and copy

Decompress webxone.zip, copy "WWW" directory to the root directory of C (it can be copied to any other directory, root directory of C is only an example), as shown in the picture



(3) Run web server program

Run web server program "WebServ.exe" in WWW directory, as shown in the picture



Now, the Web environment of WebXone is built.

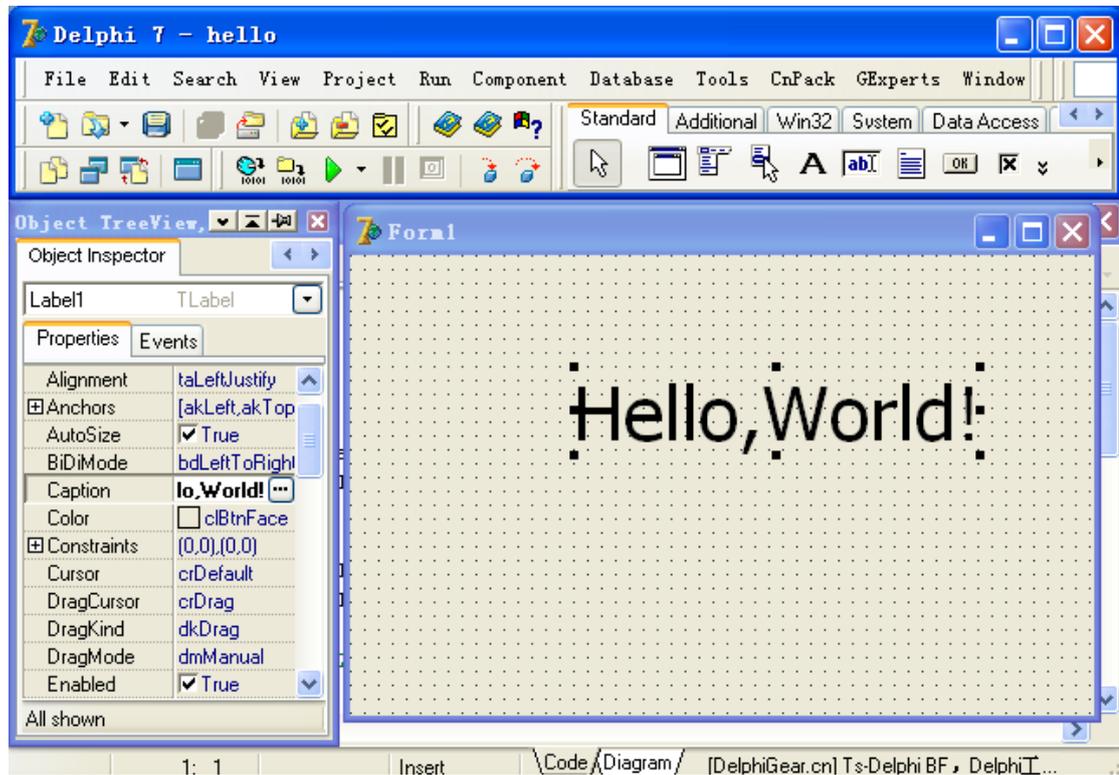
note: WWW directory is not in the root directory of C necessarily; IIS or Apache can be applied to replace WebServ. But to illustrate the question more clearly, and prevent ambiguity, the same configuration is suggested for beginners.

2.2 Apply Delphi project to WebXone application.

Take Delphi7 as an example to introduce development process

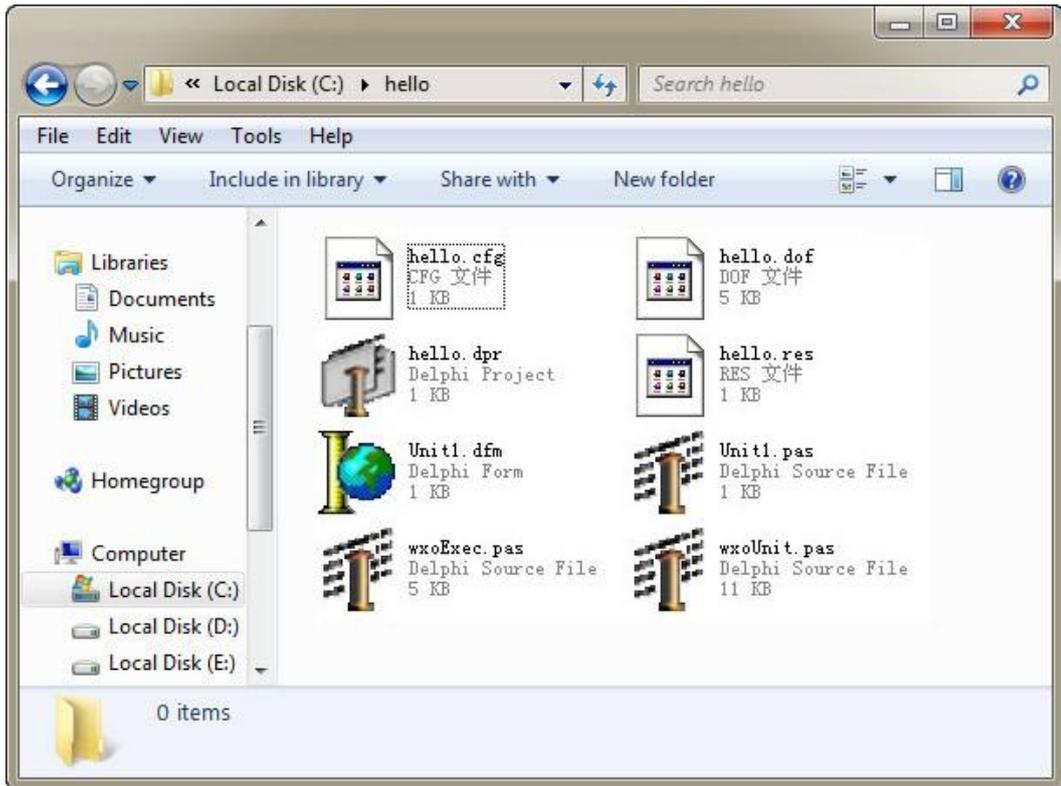
(1) New a common Delphi application

New an application, then put Label1 in form1, and set the “Caption” as “Hello, World!”. Then save the project as “hello.dpr” and the file directory is C:\hello

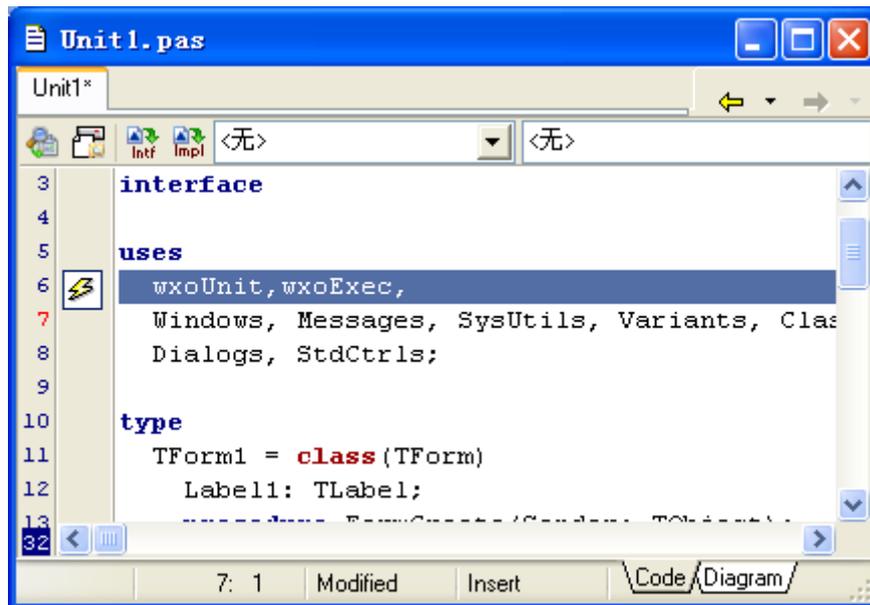


(2) Transform to WebXone application

Copy wxoUnit.pas and wxoExec.pas in the “public” directory in WebXone SDK to C:\hello, as shown in the picture.



Add references to wxoExec and wxoUnit in unit1.pas, as shown in the picture



Then add wxoPrepare in the OnCreate procedure such as

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    wxoPrepare(Handle,Application.Handle);
end;
```

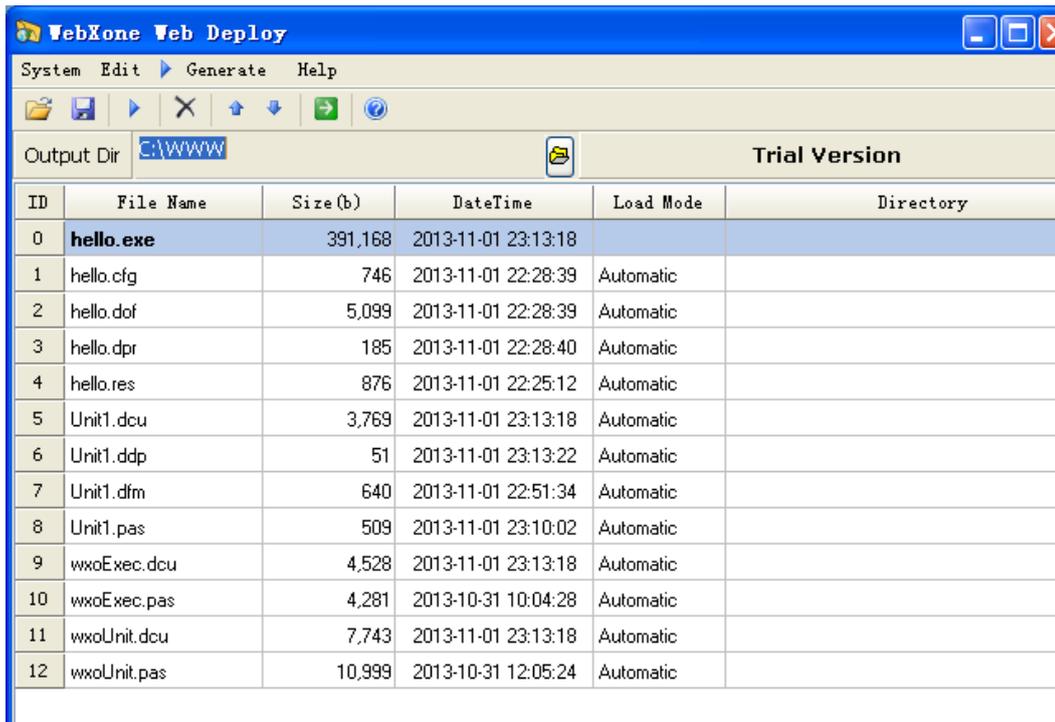
Then compile to EXE

(3) Deploy to server

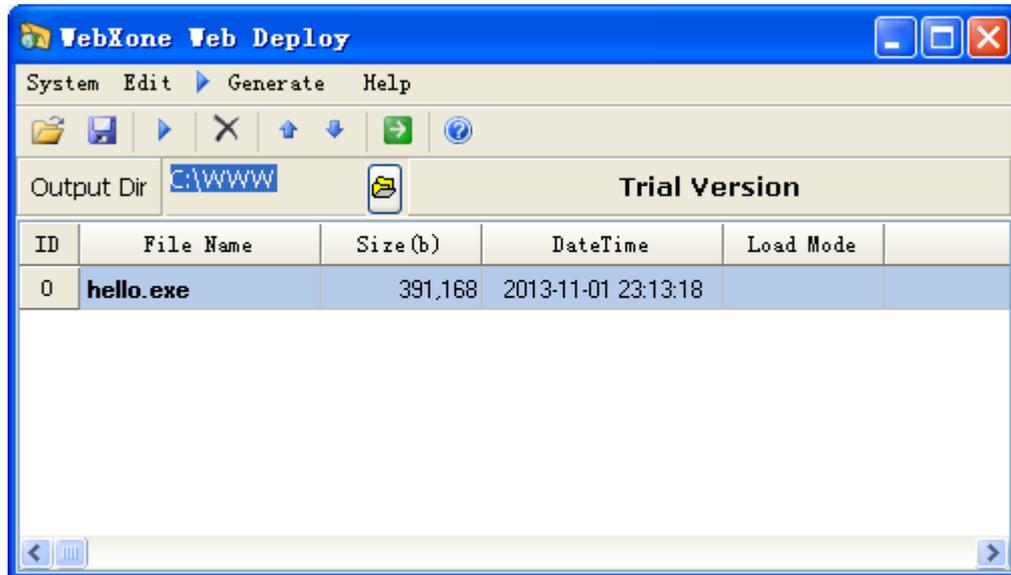
Run wxoConfig.exe in "Tools" directory in WebXone SDK

Choose output directory as C:WWW.

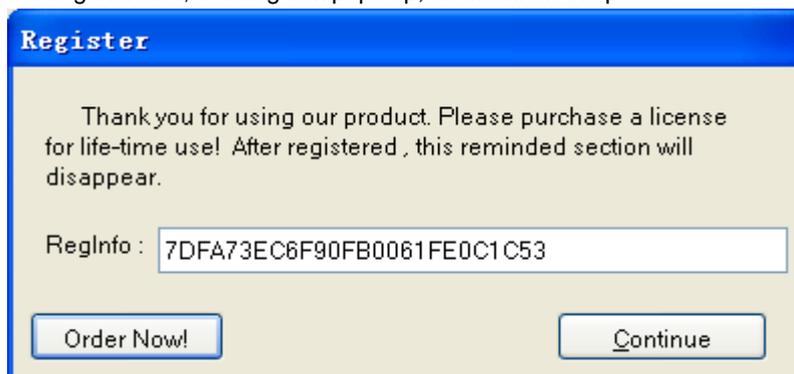
Open C:\hello\hello.exe, as shown in the picture



Selection menu: Edit->Select All, then: Edit->Remove, as shown in the picture



Then the menu is generated, a dialog box pops up, as shown in the picture



This is the dialog box for registration. Ignore it, wait for 5 seconds and click Continue. Now the deployment is finished.

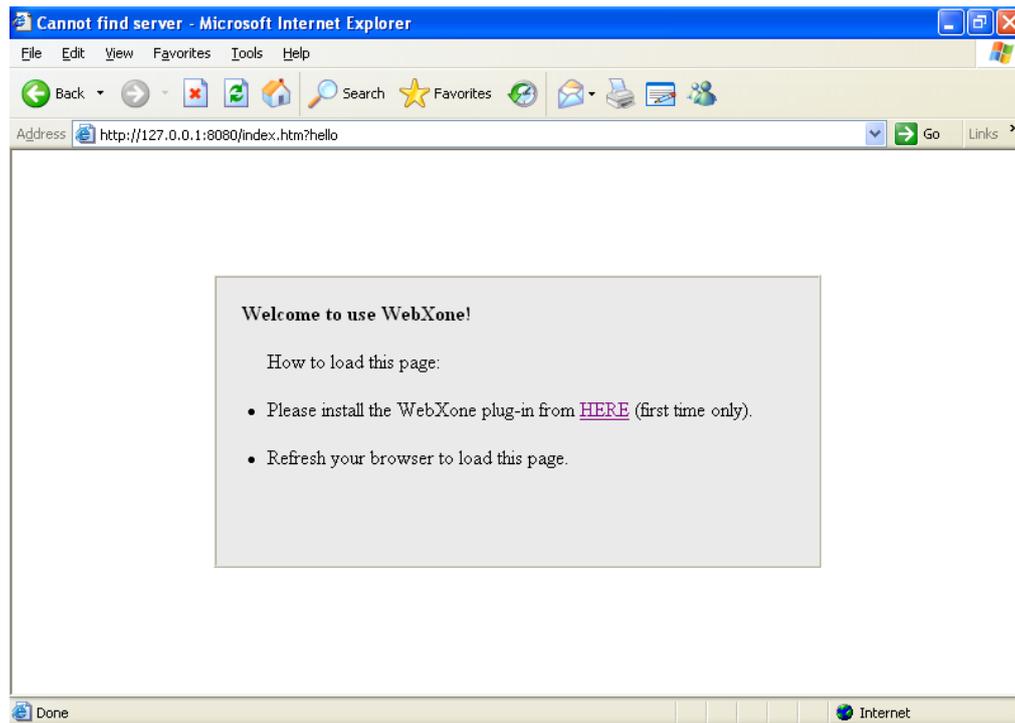
2.3 Web access

Open IE, and input the following in the address bar

<http://127.0.0.1:8080/index.htm?hello>

Note : The port is 8080

Now the installation interface of WebXone plug-in is supposed to appear.

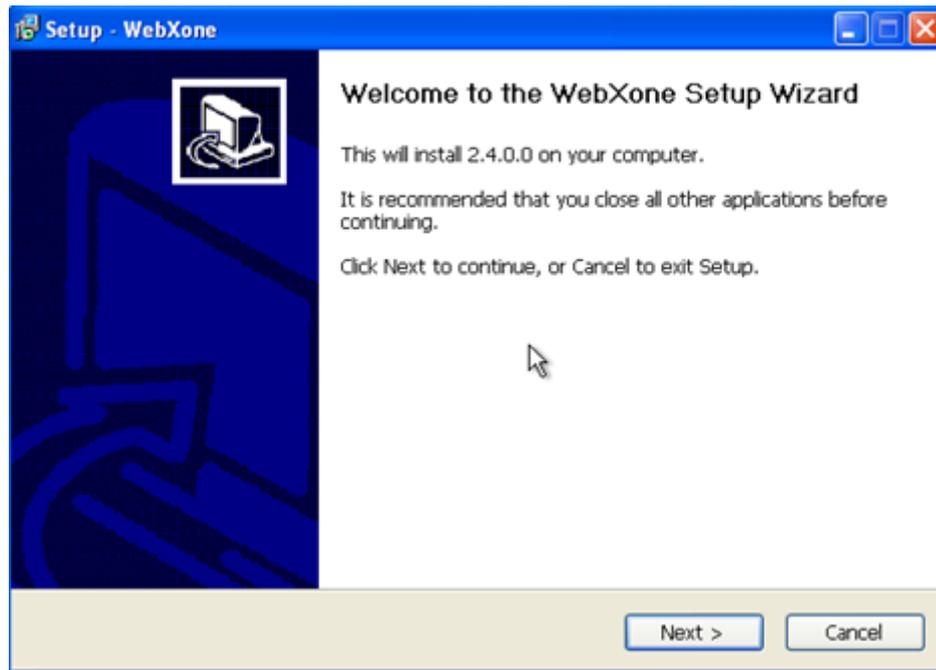


Click "here" download setup.exe and run it.

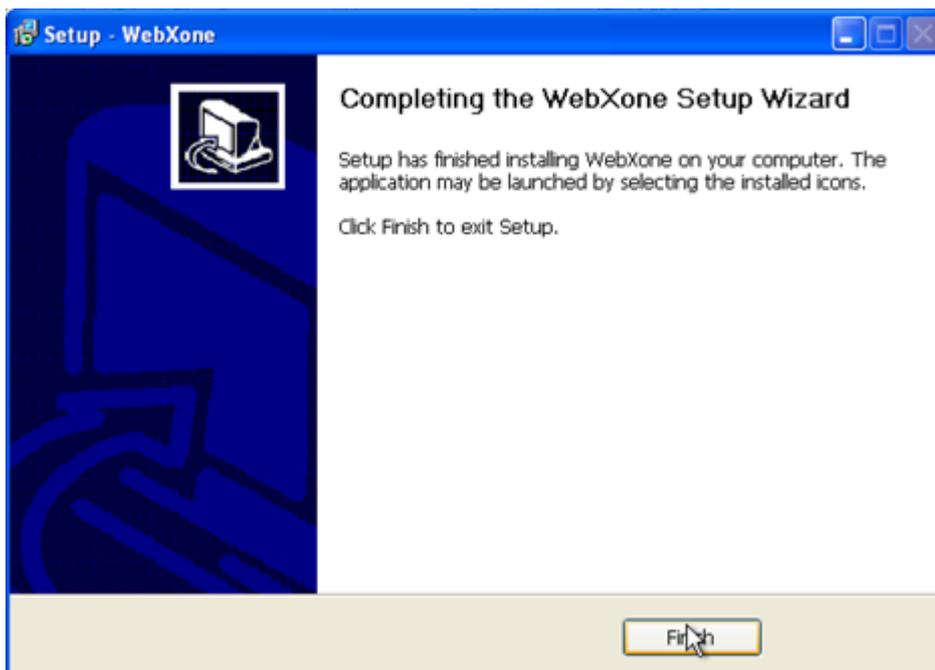
Note: Some systems has installed download tool such as Thunder, please find setup.exe in the download directory and run it.



Click "Run"

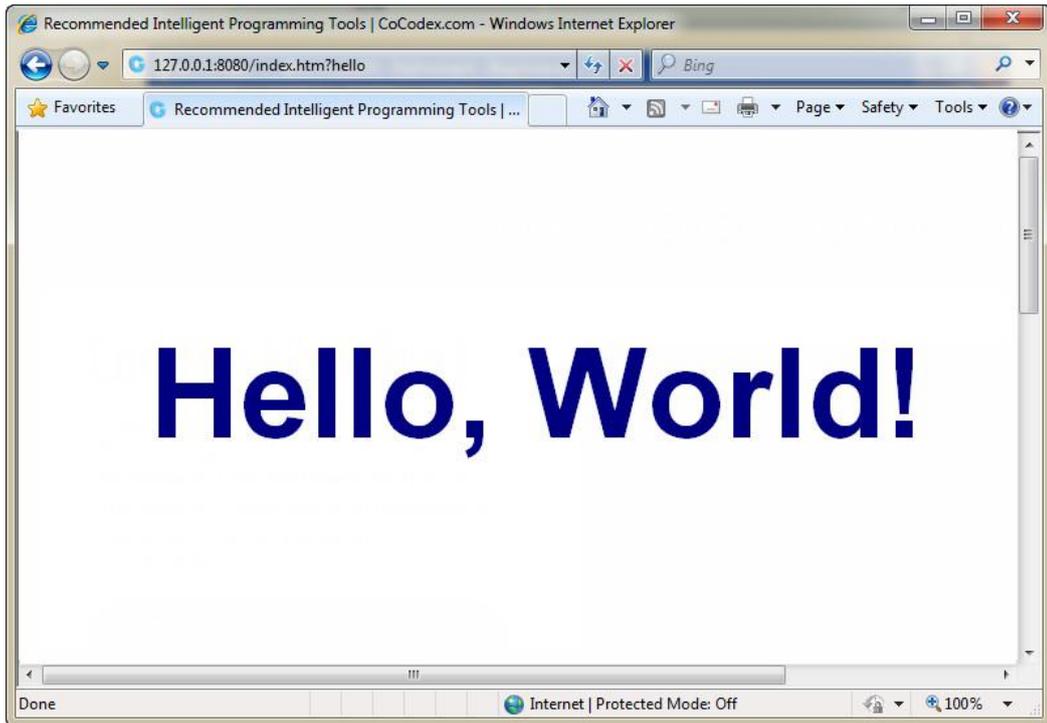


Click "Next"



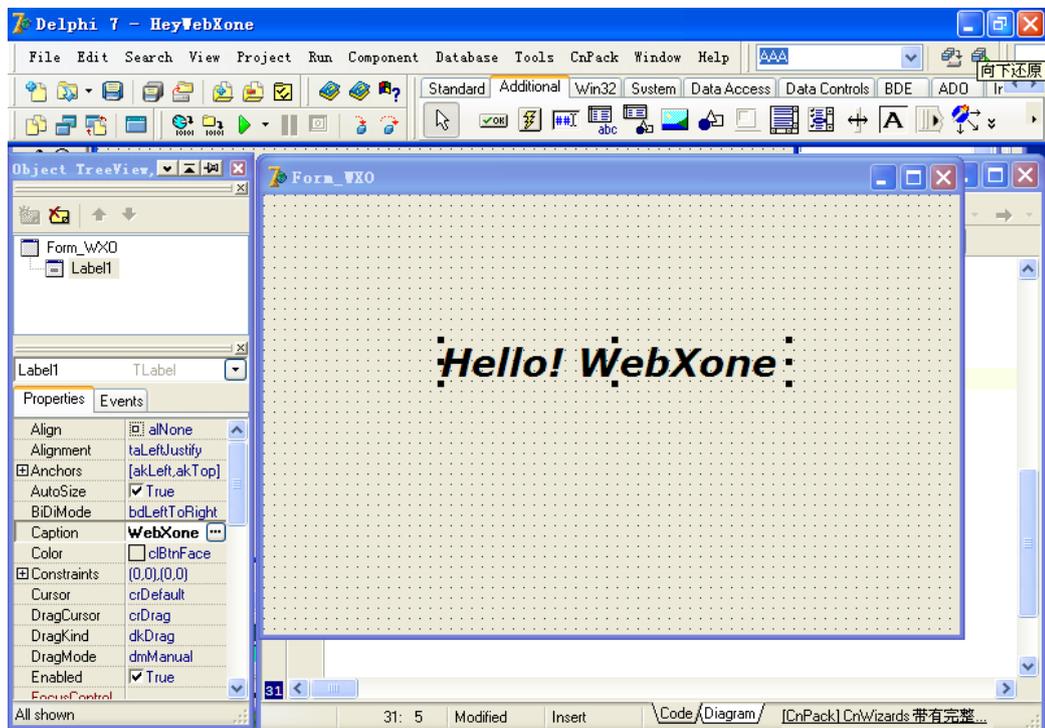
Installation is finished, click "Finish", and refresh the page.

Now the program "Hello, World" written just now can be seen, as shown in the picture.



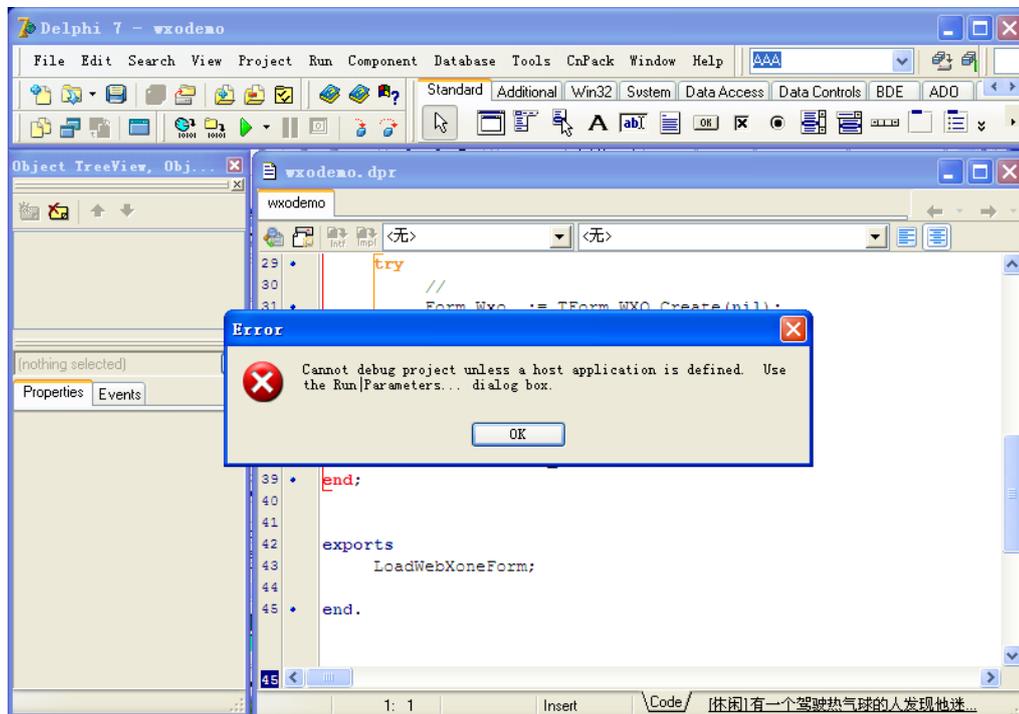
2.4 Free trial to development mode

1. Run Delphi 7, menu File → New → Application, set up an application, and add a Label control. Set the captions "llo!WebXone", then set suitable font. Save unit as Main.pas, the window name is Form_WXO, the name of the project is HeyWebXone.dpr.



Make sure it compiles successfully.

2. Open WebXone SDK, and decompress the public directory of the SDK, copy wxodemo.dpr and wxoUnit.pas to the directory which HeyWebXone.dpr is in.
3. Apply Delphi7 to open wxodemo.dpr, and compile directly, the following interface is supposed to be shown.



4. Now a file named wxodemo.wxo is supposed to be created in the directory which HeyWebXone.dpr is in. Copy the file to C:\WWW\
5. Open IE, input http://127.0.0.1/index.htm?wxodemo

2.5 apply C++ Builder to develop WebXone application

Take C++ Builder6 as an example to introduce the development process

- 1、 Run C++ Builder6, menu File → New → Application, create an application and add a Label control. Set the captions “Hello,World!”. Save the file as unit1.cpp, the name of the project is cb.dpr;
- 2、 Copy wxoExec.cpp of the public directory in WebXone SDK to the directory which cb.dpr is in.
- 3、 Menu :Project → Add to project, choose wxoExec.cpp;
Add references#include “wxoExec.h” in unit1.cpp
- 4、 Add codes in the OnCreate event

wxoPrepare((unsigned int)Handle,(unsigned int)Application->Handle);

As shown in the picture

```

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    wxoPrepare((unsigned int)Handle, (unsigned int)Application->Handle);
}
//-----

```

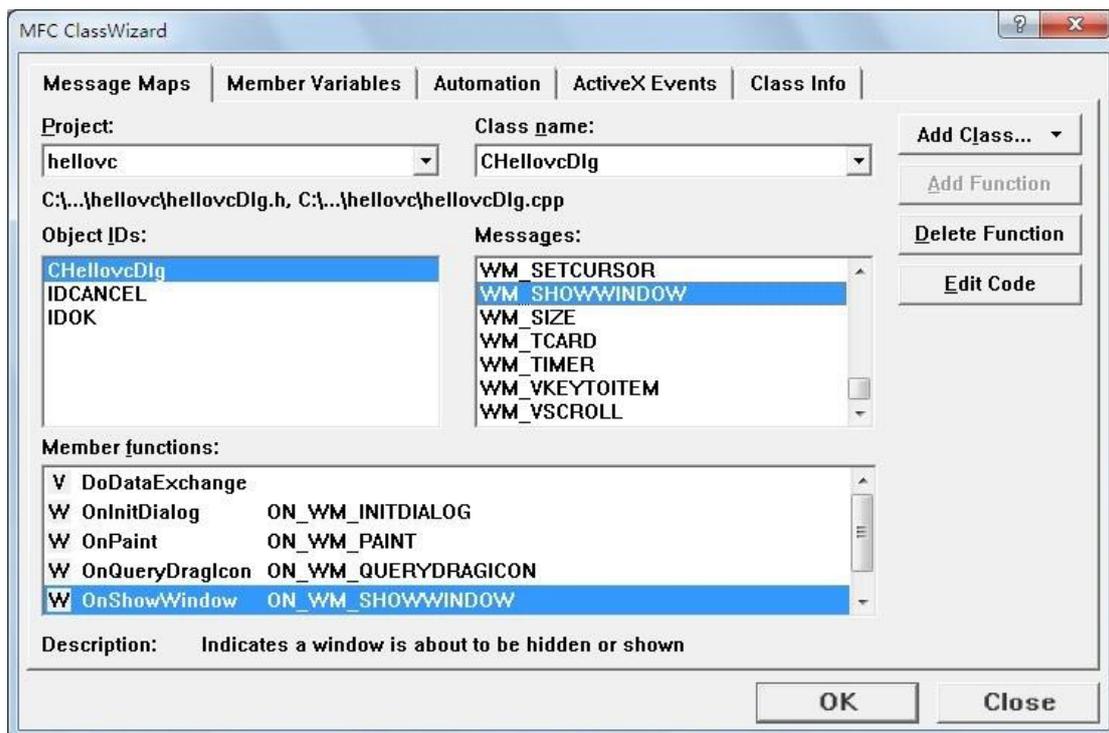
The compile to exe. (flash across by F9 is normal)

5. For further package deployment, please refer to relevant part in 2.2.

2.6 apply Visual C++ to apply WebXone application

Take Visual C++ 6 as an example to introduce development process

1. Run Visual C++ 6, choose menu "File" -> "New", choose "MFC AppWizard(exe)";
2. Choose basic dialog (dialog), then click "(finish)".
3. Copy wxoExec.cpp and wxoExec.h in public/vc to the project directory.
4. Click right button in the interface of the dialog box, and choose "Establish Class Wizard"
5. Double-click WM_SHOWWINDOW in the popup dialog box, and double-click OnShowWindow in the box



6. add #include "wzoExec.cpp" in hellovcDlg.cpp;
7. Add wxoPrepare ((unsigned int) this->GetSafeHwnd(), (unsigned int)this->GetSafeOwner()) in the OnShowWindow event;

```

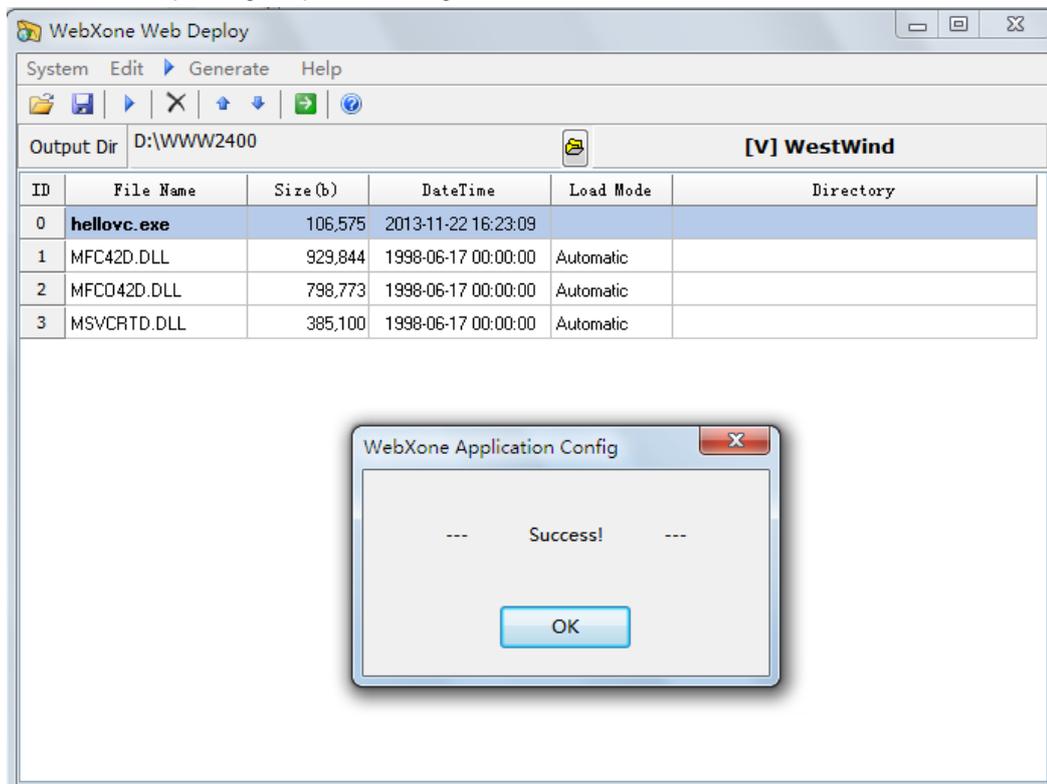
void CHellovcDlg::OnShowWindow(BOOL bShow, UINT nStatus)
{
    CDialog::OnShowWindow(bShow, nStatus);

    //
    wxoPrepare((unsigned int)this->GetSafeHwnd(),(unsigned int)this->GetSafeOwner());

    // TODO: Add your message handler code here
}

```

8. Compile to generate hellovc.exe, and copy this file to a void directory;
9. Copy the three basic runtime-support library of Visual C++6, MFC42D.dll, MFCO42D.dll and MSVCRTD.dll to the above directory.
10. Run wxoConfig.exe, and choose hellovc.exe in the above directory, and choose corresponding Output Dir, click generate



11. Open IE, input the following in the address bar
<http://127.0.0.1:8080/index.htm?hellovc>

3. Support of the relevant system

3.1 The support of browsers

Supports mainstream browsers like IE, Chrome, FireFox, Safari, Opera, etc.

3.2 The support of the client-side operating system

The client support Window 2000/Win 2003/Win XP/Win7 32/Win7 64/Win8

Now the client does not support iOS、Linux and Android.

note: Win7 non-administrator users should close UAC. The specific operation can be seen in the attachment.

3.3 support of development tools

Support all editions of Delphi

Support all editions of C++Builder

Support all editions of Visual C++

EPL is accepting the final test now.

Delphi7 are recommended.

3.4 support of the server operating system

Server operating system supports all the operating systems including Windows, Linux, Apple series.

3.5 support of the server service programs

The server supports all the service programs including IIS, Apache and other series, the small service program netbox is recommended. The customized WebServ in the above document is not a must, and it can be replaced. Mime can be added to IIS.

3.6 Support of the database.

Support all the databases.

3.7 Support of the middleware

Support all middleware including RTC, RO. Excellent domestic three-level plug-ins like RemoteADO and QuickBurro are recommended.

4. Frequently Asked Questions

- **Can registered user have access to further update?**

Answer: have lifelong free access to it.

- **Is mobile platform enabled on client-side?**

Answer: no

- **Does the further update have development play supported by the mobile platform?**

Answer: We plan to develop a edition like unigui: WebXoneJS. The specific content and time is to be determined.

- **Can it be deployed in Linux server?**

Answer: Yes! webXone demo site can be deployed in Linux server.

- **Does it support hundreds of people or more online together?**

Answer: Yes! Someone has developed a system supporting tens of thousands of people based on WebXone.

- **Why does load fail and there is 126 error?**

Answer: Because some subsidiary file are used in the application (such as DLL), but it does not exist on the client-side and requires to add this file when users package using WXOConfig.

- **Why cannot wxo be found when using IIS Web service?**

For details please see annex.

- **How to package many files by wxoConfig (including some configuration files and dynamic link library) ?**

Answer: Copy WebXone application (wox or exe) as well as other relevant files (such as configuration files, dll, etc) to a void directory and open the application (wox or exe) by wxoConfig, and generate it directly.

Note: Output directory is supposed to be web directory

- **How to pass parameters by URL?**

Answer: You can pass parameters refer to the following website (just an example and the website cannot be used) www.xxx.com/index.htm?id=main&&abcde. The website will open the main application. The program gives the website to the variable `gsUrl` and parses it in the program.

Note: separator is "&&"

- **If the application is large, will run-up time be long because client-side has to load the application every time?**

Answer: The Intelligent compression loading mechanism `WebXone` applies load compressed programs and compresses it automatically when client-side have access to it for the first time. Afterwards if client-side has access to it, client-side compares the local files with server-side files automatically. If server-side files are not updated, there is no second upload.

- **Can WebXone have access to local port, and how about operating serial port or printer**

Answer: You can operate the local port including serial port, parallel port, USB, printer, etc.

- **How to amend page title?**

Answer: Through `wxoSendMessage` in `wxoFeec`. Through invoking Javascript to achieve it.

```
//Delphi
    wxoSendMessage(giApp,
        Handle,
        'RNJS',
        Handle,
        'window.document.title = '"+Edit1.Text+"'");
//C
    wxoSendMessage(giApp,_WXO_MAINFORMHANDLE,"RNJS",_WXO_MAINFORMHANDLE,
        "window.document.title = 'new title'");
```

- **How to close the page?**

Answer: Through `wxoSendMessage` in `wxoFeec`. Through invoking Javascript to achieve it. Generally you can use `wxoClose` directly.

- **Is MDI supported?**

Answer: Yes

- **How to shorten URL?**

Answer: If the application name is default.wxo or default.exe ,“?” as well as the following application name are not required to be added after employment and WebXone web application can be visited directly, such as <http://athtek/web-application-kit/demos.html>. If you want to shorten it further, you can have access to it through the configuration of IIS and Apache like www.athtek.com/demo/.

- **Is it supported to customize directory installation?**

Answer: It is not supported now, but future editions may add this function.

- **Is it supported to customize relevant information of setup.exe?**

Answer: Yes. Note that users can get innosetup-format source code of setup.exe. And users can amend information such as the names of the plug-ins, developers, version number, etc.

5. Recent update

2014-05-01 version 2.4.0.0

- 1 Add the ability to implant EXE directly, and compatible with all the VCL components completely (highly recommended).
- 2 Add the ability to support MDI.
- 3 Add the ability to exit automatically for WXO in exceptions (mainly aim at some browsers for the minority)
- 4 Update setup.exe, solve the problem of forgetting to exit the browser when updating.

2013-05-23 version 2.3.6.0

- 1 Add the ability to deal with exceptions when exit and compatible with VCL components better.
- 2 Add the ability to run JavaScript.
- 3 Add the ability to deal with main window such as OnClose.

- 4 Add digital signature.
- 5 Solve the problem of 360 reporting virus.
- 6 Solve the bug that it is easily blocked when using FireFox.
- 7 Apply "MD5+size of the file+time" for duplication check and it is more stable.

2012-12-18 version 2.3.0.2

- 1 Be Compatible with all the mainstream browsers including IE, Chrome, FireFox, Safari and Opera.
- 2 Add the ability to embed automatically of non-main windows.
- 3 Cancel the registration restriction and all developers have free access to it and have no restrictions.
- 4 Add the lightweight server program WebServ to debug systems fast.
- 5 Add the attachment loading mode and add the "load not update" mode on the basis of original model;
- 6 Redesign wxoUnit, and make it more convenient and quick for transition from ordinary procedure to WebXone.
- 7 Apply MD5 to update verification and it is more scientific than the original size+time.

2012-06-20 version 2.2

1. Apply modular construction, and the system can be updated gradually.
2. Amend the loading directory to solve the requests of Oracle have for directories.

2012-03-27 version 2.1

1. Have free access to single WXO mode.
2. Have free access to applications which have less than 5 subsidiary files.
3. Amend the registration mode and users can release applications for infinite times and development and deployment don not bind to computers.
4. Support customized installation interface (index.html source codes are supported)

2012-03-19 version 2.0

1. Add multi-annex file deployment.
2. Add the ability to compressing load and increase the loading speed greatly.
3. Apply the intelligent loading technique and decrease the loading time for the second load to almost 0;
4. Test the third party control (such as cxGrid) further.

2011-12-10 Version 1.1

1. Add the demo such as Hello, World!
2. Solve the bug that directory can be generated.
3. Solve some unsuccessful LoadLibray bugs.

2011-11-21 Version 1.0

Achieve WebXone 1.0, and users can apply tools like Delphi generate web applications.

6. Protocol

1.1 Free trial

Users have free access to development.

Feature:

- Support simple file(.wxo) deployment

1.2 Standard Edition

An application including many files at one time can be released.

- Support simple file(.wxo) deployment;
- Support multi-file intelligent compress deployment
- Deploy package operation(wxoConfig.exe) is restricted to run in specific computers;
- Have free access to the Delphi source code and controls of WebServ.exe;
- Have free access to innosetup-format source code of setup.exe;
- Have free access to one set of CRM source code;

1.3 Enterprise Edition

An application including many files at one time can be released.

- Support simple file(.wxo) deployment;
- Support multi-file intelligent compress deployment
- Deploy package operation(wxoConfig.exe) can run in many computers;
- Have free access to the Delphi source code and controls of WebServ.exe;
- Have free access to innosetup-format source code of setup.exe;
- Have free access to one set of CRM source code;
- **A complimentary standard edition of RemoteADO or QuickBurro (primary developer are responsible for after-sales service)**

Attachments:

1. Solutions to when WXO cannot be found when using IIS?

Reason: IIS restrict to download WXO files to local, so WXO files can be found in local.

Solutions: Amend HTTP header information in IIS to achieve this, mainly by amending MIME information to achieve it and you just need to add four types (.wx,.wa,.xml,.wxo) and choose application/vnd.rn-realmedia for Mime type.

2. Win7 non-admin users close UAC

Input the following directly in operation:

“C:\Windows\System32\UserAccountControlSettings.exe”. The settings for UAC pops up directly and then choose “Never Notify”. Some users are supposed to restart the computers.

3. Brief introduction of the package deploy tool wxoConfig

wxoConfig is a tool to package and deploy WebXone application(.wxo files or processed .exe files) as well as its annex files. WebXone packaged and deployed by wxoConfig are fast when having access to client-side and does not need to load aging when visiting it for the second time.

Basic operations as followed:

- (1) Preparations: New a directory and package WebXone application (.wxo files or processed .exe files) as well as its annex files and copy it to the new directory.
- (2) Start the program, and run wxoConfig.
- (3) Choose the Output Dir you plan to output to including the directory index.htm, default , setup.exe;
- (4) Choose WXO. Choose the .wxo file or processed .exe file.
- (5) Manage subsidiary files: Through menu Edit--->Delete to delete the chosen subsidiary files.
- (6) Configure subsidiary files: Through menu Edit--->Set current file Load... To configure the current subsidiary files.

Of it

Load Automatic is the default option, indicates that client-side loads the file from server-side automatically and loads again after updating from server-side.

Load Manual indicates that client-side cannot load the file from server-side automatically, and users need to load the file from client-side manually, like GetAttachFile(Url,2). The first parameter is current URL, and the second is ID num of the subsidiary file.

Load and No update indicate that client-side load the file from server-side automatically and not load again after updating from server-side.

6、package and deploy: Use menu “Generate” to generate the directory after packaging, and copy it to the directory you plan to deploy in.