OnTime 2005
(V5.0.1)

Web Services SDK

Programmer's Guide

Table of Contents

Introduction

The OnTime Web Services SDK is a set of web services that are designed to allow developers to integrate OnTime Defect Tracking capabilities into their own windows and web based applications.

Here is a sampling of what you can do with the OnTime Web Services SDK:

- Create web-based capabilities for adding or updating defects
- Create web-based capabilities for viewing or reporting on defects in your OnTime database
- Create custom web-based or windows-based user interfaces to view, update or list defects and/or features.
- Integrate defect reporting capabilities into your own applications so that your users can report a defect from anywhere on the Internet.

The possibilities are truly endless, so lets get started.

Installation & Setup

The distribution zip file contains the following files:

- **OnTime 2005 SDK Programmer's Guide.Doc** – This document

- **OnTimeSDKWebServicesV50Setup.msi** – This is the set of SOAP-Based Web Services that make up the OnTime Software Development Kit.

- **OnTimeSDKWebSamplesV50Setup.msi** – This is a sample C# web project that uses the SDK to demonstrate adding a defect through web forms and listing all the defects and features in the OnTime Database.

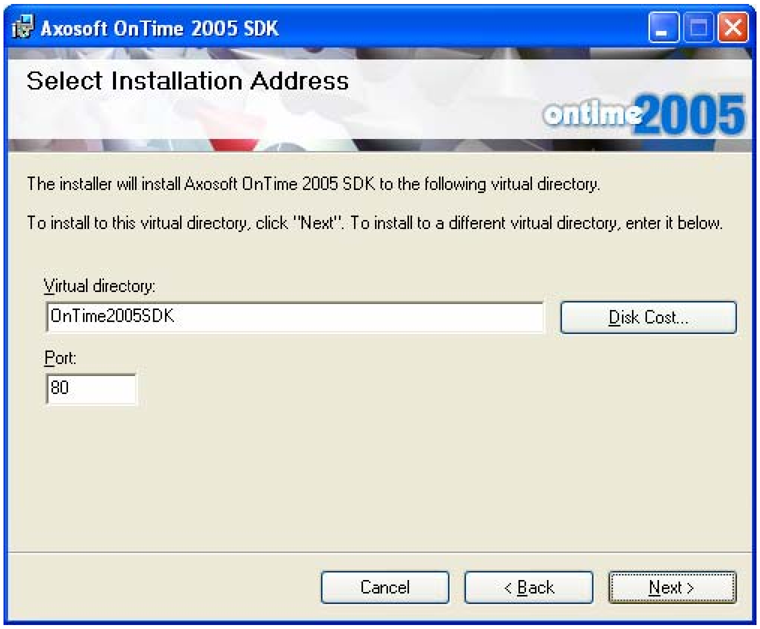
- **OnTimeSDKWinSamplesV50Setup.msi** – This is a sample C# windows project that uses the SDK to demonstrate adding a defect through a Windows Form and listing all the defects in the OnTime Database.

Before you can use any of the samples, you must first install the OnTime Web Services SDK using the "OnTimeSDKWebServicesV50Setup.msi" file.

Installing OnTime Web Services

To install OnTime Web Services, follow these steps:

1. Run OnTimeSDKWebServicesV50Setup.msi to start the Web Services installation wizard.
2. Click "Next" on the welcome screen
3. If you agree with the licensing agreement, click on the "I Agree" radio box and click "Next". (if you don't agree with the licensing, please contact sales@axosoft.com)
4. On the "Select Installation Address" step, choose a virtual directory for the installation of the OnTime Web Services



Select Installation Address

The installer will install Axosoft OnTime 2005 SDK to the following virtual directory.

To install to this virtual directory, click "Next". To install to a different virtual directory, enter it below.

Virtual directory: OnTime2005SDK

Port: 80

> ***Note:*** The Samples that are included in the SDK assume that the default virtual directory of "OnTime2005SDK" is the installation address that is selected. If you change this location, you will have to update the Web References in the included samples.

5. Click "Next" to start the installation
6. After the installation completes successfully, using a text editor (such as Notepad), open the "web.config" file (this file is normally located in the "\inetpub\wwwroot\OnTime2005SDK" directory by default). This should look something like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="ConnectionString" value="ENTER YOUR SQL SERVER CON STRING HERE"
/>

    <!--  SECURITY TOKEN
               Generate a GUID and replace the value below with your newly
generated GUID for security.
    -->
    <add key="SecurityToken" value="ENTER A GUID HERE" />

    <!--  WEB SERVICES USER
               The Id of the OnTime user that the web services will run as.
Any actions performed on defects and features through the web services will
appear as done by this user.
    -->
    <add key="WebServicesUser" value="1" />
  </appSettings>

  <system.web>
  ...
```

7. At the top of the file, locate the "ConnectionString" AppSettings key. Replace the "`ENTER YOUR SQL SERVER CON STRING HERE`" string with a valid connection string to the OnTime

database which you would like to work with. A valid connection string can typically take the following form:

`server=(local);database=ontimedb;uid=ontimeuser;pwd=otdbpwd`

Where (local) is the SQL Server's name, "ontimedb" is the name of your OnTime database, "ontimeuser" is the SQL Server user name to connect as and "otdbpwd" is the password for that user.

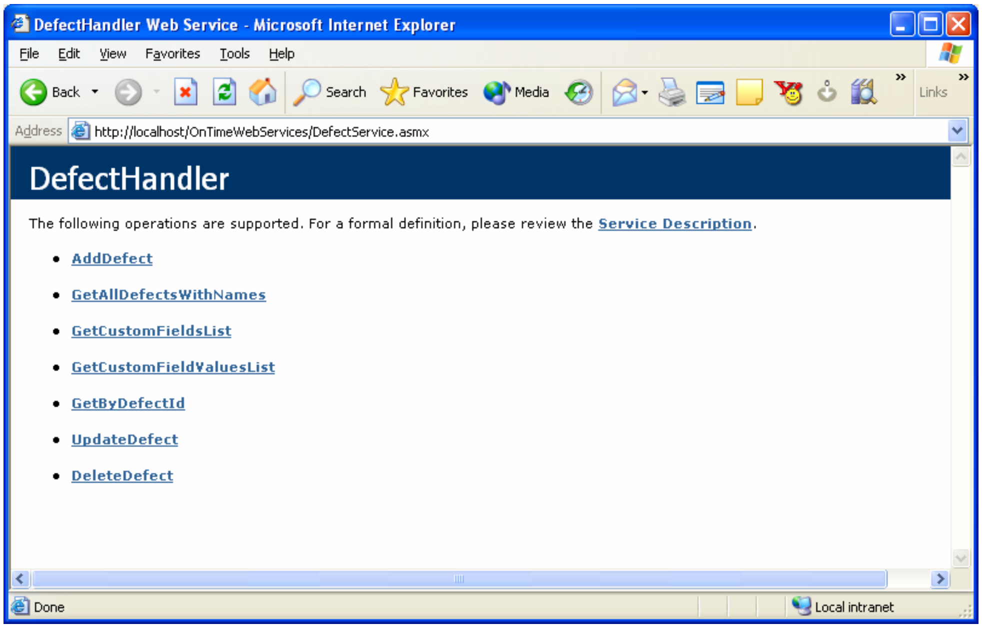
8. Next locate the "SecurityToken" key. Generate a new GUID using Visual Studio (or any other tool) by going under Tools -> Create GUID and replace the "`ENTER A GUID HERE`" string with the GUID you generated. The GUID should look something like this: `3060C5DE-7398-4e70-909B-2F7209D6FF08`. This GUID is used as a security measure and is required for every method call in the SDK.

9. Next locate the "WebServicesUser" key. This is the ID of the OnTime user that the SDK will 'impersonate'. The default is '1' which is the ID of the Administrator account. You may create a separate user for this specific use and enter its ID instead.

Test to make sure that everything is working properly by opening an Internet Explorer browser window and browse to the following address:

http://localhost/OnTime2005SDK/DefectService.asmx

Replace "OnTime2005SDK" with the virtual directory name that you chose above. When browsing to this address, you should see a screen that looks like this:


DefectHandler Web Service - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Media
Links
Address
http://localhost/OnTimeWebServices/DefectService.asmx
Done
Local intranet

DefectHandler

The following operations are supported. For a formal definition, please review the Service Description.

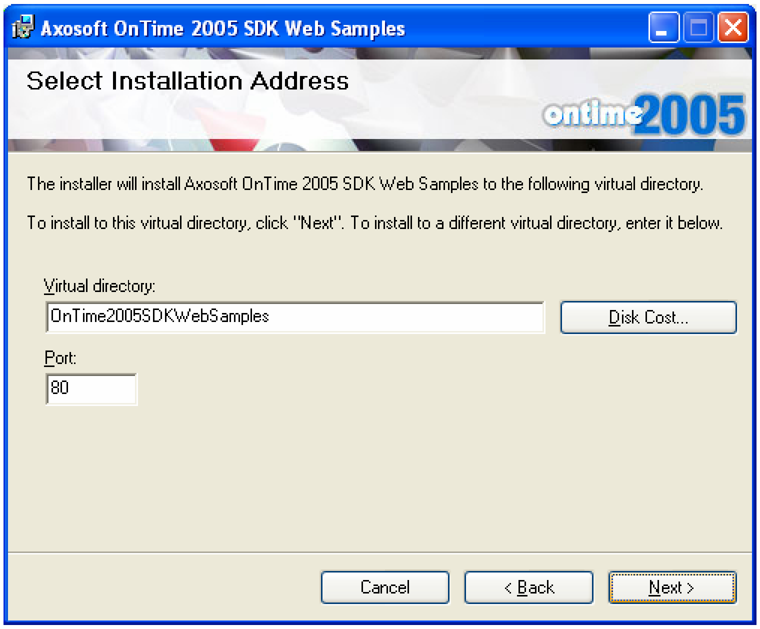
- AddDefect
- GetAllDefectsWithNames
- GetCustomFieldsList
- GetCustomFieldValuesList
- GetByDefectId
- UpdateDefect
- DeleteDefect

The OnTime Web Services and web methods are now ready to be used in your own development projects.

Installing OnTime SDK Web Samples

To install the OnTime SDK Web Samples, follow these directions:

1. Run OnTimeSDKWebSamplesV50Setup.msi to start the Web Services installation wizard.
2. Click “Next” on the welcome screen
3. On the “Select Installation Address” step, choose a virtual directory for the installation of the OnTime SDK Web Samples


Axosoft OnTime 2005 SDK Web Samples
Select Installation Address
ontime2005
Disk Cost...
Cancel
< Back
Next >

The installer will install Axosoft OnTime 2005 SDK Web Samples to the following virtual directory.

To install to this virtual directory, click "Next". To install to a different virtual directory, enter it below.

Virtual directory: OnTime2005SDKWebSamples

Port: 80

4. Click “Next” to start the installation
5. After the installation completes successfully, using a text editor (such as Notepad), open the “web.config” file (this file is normally located in the “\inetpub\wwwroot\OnTime2005SDKWebSamples” directory by default). This should look something like this:

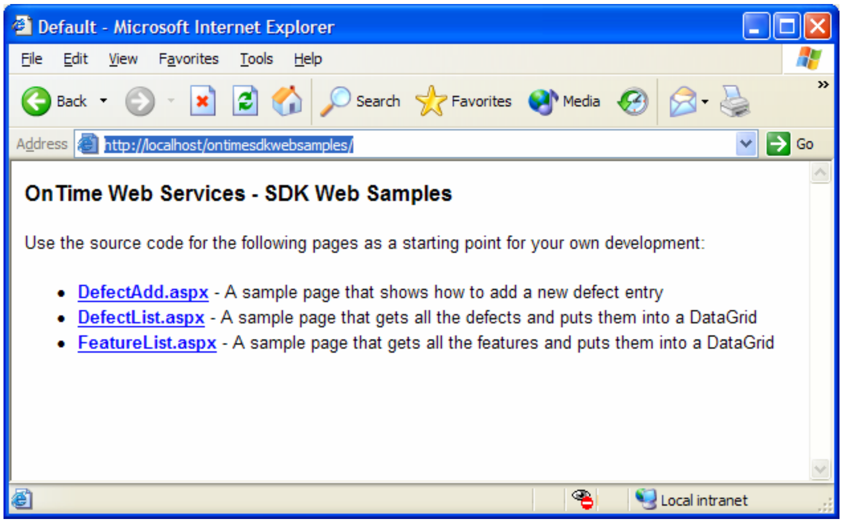
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <!--  SECURITY TOKEN
               Generate a GUID and replace the value below with your newly
generated GUID for security.
    -->
    <add key="SecurityToken" value="ENTER A GUID HERE" />

  ...
```

6. Replace the “`ENTER A GUID HERE`“ string with the GUID created when you installed the OnTime web services. It should match the GUID in the web.config file of the web services.

To test that the installation worked successfully, browse to the following page:

http://localhost/ontime2005sdkwebsamples/

Be sure to replace the “ontime2005sdkwebsamples” with the installation address that you chose above. You should see a screen that looks similar to this:

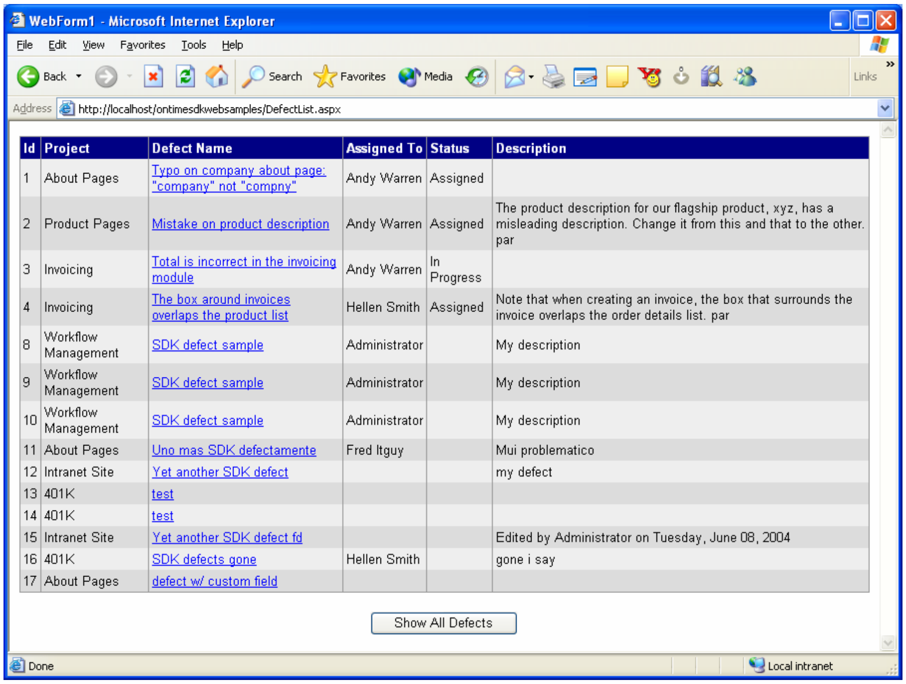

Default - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://localhost/ontimesdkwebsamples/
Go
Local intranet

OnTime Web Services - SDK Web Samples

Use the source code for the following pages as a starting point for your own development:

- DefectAdd.aspx - A sample page that shows how to add a new defect entry
- DefectList.aspx - A sample page that gets all the defects and puts them into a DataGrid
- FeatureList.aspx - A sample page that gets all the features and puts them into a DataGrid

Browse to the “DefectList.aspx” page and click on the “Show All Defects” button. If everything worked as expected, you should see a page that resembles the following (with defects from your own OnTime database being displayed):


WebForm1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Links
Address
http://localhost/ontimesdkwebsamples/DefectList.aspx
Show All Defects
Done
Local intranet

Id	Project	Defect Name	Assigned To	Status	Description
1	About Pages	Typo on company about page: "company" not "compny"	Andy Warren	Assigned	
2	Product Pages	Mistake on product description	Andy Warren	Assigned	The product description for our flagship product, xyz, has a misleading description. Change it from this and that to the other. par
3	Invoicing	Total is incorrect in the invoicing module	Andy Warren	In Progress	
4	Invoicing	The box around invoices overlaps the product list	Hellen Smith	Assigned	Note that when creating an invoice, the box that surrounds the invoice overlaps the order details list. par
8	Workflow Management	SDK defect sample	Administrator		My description
9	Workflow Management	SDK defect sample	Administrator		My description
10	Workflow Management	SDK defect sample	Administrator		My description
11	About Pages	Uno mas SDK defectamente	Fred Itguy		Mui problematico
12	Intranet Site	Yet another SDK defect			my defect
13	401K	test			
14	401K	test			
15	Intranet Site	Yet another SDK defect fd			Edited by Administrator on Tuesday, June 08, 2004
16	401K	SDK defects gone	Hellen Smith		gone i say
17	About Pages	defect w/ custom field			

Important Note: The web samples have been set up to reference web services that are located at http://localhost/ontime2005sdk. If you chose to install the OnTime Web Services at a different location, you will need to open the web.config file and change the address of the web references to the appropriate location.

Locate the following section from the web.config file:

```
<appSettings>
        <add key="OnTimeSDKWebSamples.AttachmentService.AttachmentHandler"
value="http://localhost/OnTime2005SDK/AttachmentService.asmx"/>
```

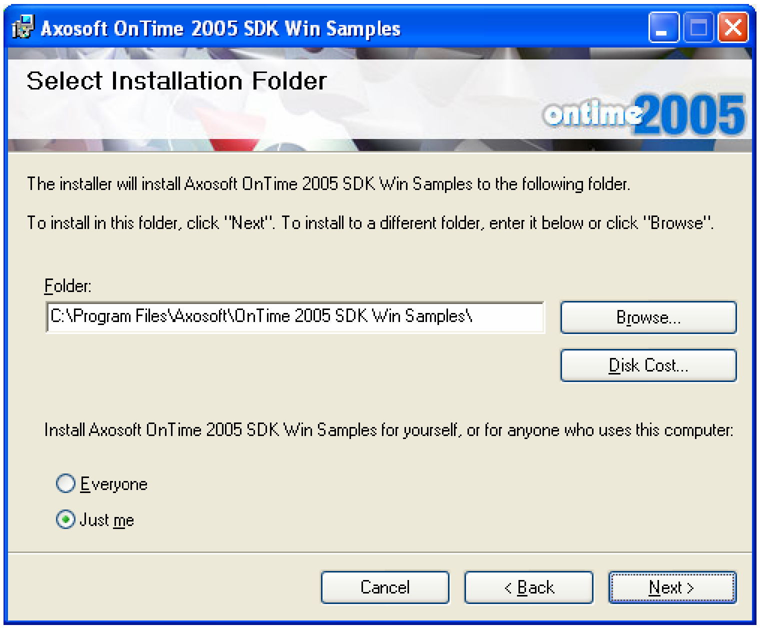
```
      <add key="OnTimeSDKWebSamples.TypeService.TypeHandler"
value="http://localhost/OnTime2005SDK/TypeService.asmx"/>
      <add key="OnTimeSDKWebSamples.ProjectService.ProjectHandler"
value="http://localhost/OnTime2005SDK/ProjectService.asmx"/>
      <add key="OnTimeSDKWebSamples.PickListService.PickListHandler"
value="http://localhost/OnTime2005SDK/PickListService.asmx"/>
      <add key="OnTimeSDKWebSamples.FeatureService.FeatureHandler"
value="http://localhost/OnTime2005SDK/FeatureService.asmx"/>
      <add key="OnTimeSDKWebSamples.UserService.UserHandler"
value="http://localhost/OnTime2005SDK/UserService.asmx"/>
      <add key="OnTimeSDKWebSamples.DefectService.DefectHandler"
value="http://localhost/OnTime2005SDK/DefectService.asmx"/>
</appSettings>
```

Change the values in bold to reflect the location of the OnTime Web Services installation.

Installing OnTime SDK Windows Samples

To install the OnTime SDK Windows Samples, follow these directions:

7. Run OnTimeSDKWinSamplesV50Setup.msi to start the Web Services installation wizard.
8. Click “Next” on the welcome screen
9. On the “Select Installation Folder” step, choose a directory for the installation of the OnTime SDK Windows Samples


Axosoft OnTime 2005 SDK Win Samples
Select Installation Folder
ontime2005
Browse...
Disk Cost...
Cancel
< Back
Next >

The installer will install Axosoft OnTime 2005 SDK Win Samples to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

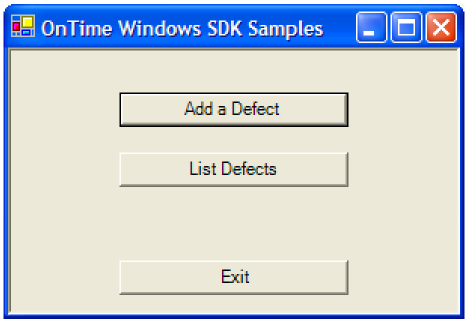
Folder: C:\Program Files\Axosoft\OnTime 2005 SDK Win Samples\

Install Axosoft OnTime 2005 SDK Win Samples for yourself, or for anyone who uses this computer:

○ Everyone
● Just me

10. Click “Next” to start the installation

To verify the installation run the “OnTimeSDKWinSamples.exe” file from the installation directory (by default, the installation directory is “C:\Program Files\Axosoft\OnTime 2005 SDK Win Samples”). You should see a program that looks like this:



Click on “List Defects” to open the list defects window and then click “Get Defects” to ensure the sample is able to reach the OnTime web services.

Important Note: The Windows samples have been set up to reference OnTime web services that are located at http://localhost/ontime2005sdk. If you chose to install the OnTime Web Services at a different location, you will need to open the “OnTimeSDKWinSamples.exe.config” file and change the address of the web references to the appropriate location.

Locate the following section inside the OnTimeSDKWinSamples.exe.config file:

```
<appSettings>
       <add key="OnTimeSDKWinSamples.DefectService.DefectHandler"
value="http://localhost/ontime2005sdk/defectservice.asmx"/>
</appSettings>
```

Change the value in bold to reflect the location of the OnTime Web Services installation.

SDK Samples

Included with the OnTime Web Services SDK is 2 sample projects:

- **OnTime SDK Web Samples** – This set of samples demonstrate the usage of the OnTime Web Services through a set of ASP.NET web pages using the C# Programming Language.

- **OnTime SDK Win Samples** – This sample demonstrates the use of OnTime Web Services through WinForms using the C# Programming Language.

Each of the sample projects is covered in one of the sections below. However, before continuing, be sure to install the samples using the instructions provided earlier in this guide.

Web Samples

Introduction

Using Visual Studio.net, open the “OnTimeSDKWebSamples.csproj” project file (by default, it’s located at “C:\Inetpub\wwwroot\OnTime2005SDKWebSamples”). This project contains 4 web pages:

- **Default.aspx** – This page is a simple HTML page that provides quick links to the other pages.
- **DefectAdd.aspx** – This is a page that demonstrates “Adding” and “Editing” a new defect to the OnTime database using the OnTime Web Services SDK.
- **DefectList.aspx** – This page demonstrates how to obtain a list of defects and display them using an ASP.NET DataGrid.

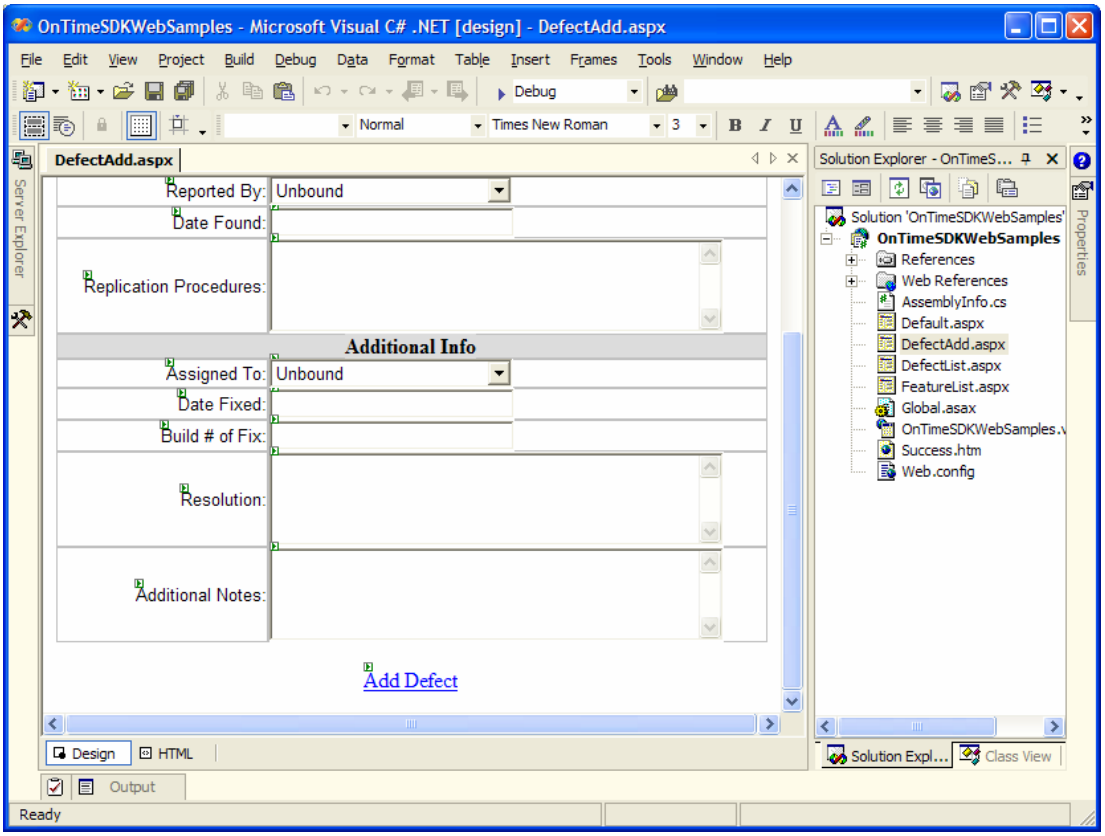
- **FeatureList.aspx** – This page demonstrates how to obtain a list of features and display them using an ASP.NET DataGrid.

Security

In order to limit who can use the OnTime web services on your server, a GUID is defined in the web services ‘web.config’ file under the ‘SecurityToken’ key. This GUID is needed by anyone who wants to call the OnTime SDK methods, and used in conjunction with SSL encryption should provide a secure way of accessing the OnTime data through your server.
Thus, every method call takes in one parameter which is a GUID that will have to match the GUID defined in the web services ‘web.config’ file.

Setting up the “Add Defect” WebForm

To start with, open the “DefectAdd.aspx” page in the designer. You should have a screen that looks similar to the one below:



The designer shows a web form that contains input fields for most of the information that can be stored about a defect. Notice that the designer also provides 3 unbound drop-down list boxes to allow the user to select which project this defect belongs to, which user reported the defect and to which user to assign the defect. There are also 3 drop-down list boxes for setting the status, priority, and severity of a defect. In order to fill these drop-down “pick-lists”, we use three web services provided by OnTime:

- ProjectService.asmx
- UserService.asmx
- TypeService.asmx

These services are already added to the project, but to see how this sample uses the services to fill the pick lists, view the code-behind for the “DefectAdd.aspx” page. Locate the code for the “Page_Load” method and find the following code section:

```
       // Fill up the projects list box
       ProjectService.ProjectHandler projectHandler = new
ProjectService.ProjectHandler();

       projectDropDown.DataSource = projectHandler.GetAllProjects(new
Guid(System.Configuration.ConfigurationSettings.AppSettings["SecurityToken"]));
       projectDropDown.DataTextField = "Name";
       projectDropDown.DataValueField = "ProjectId";
       projectDropDown.DataBind();
```

This code fills up the project drop-down by first creating a “ProjectHandler” object which serves as a proxy to the OnTime Project Web Service. It then sets the drop down control’s “DataSource” property to the value returned by the “GetAllProjects()” method of the Project Web Service. The GUID defined in the web.config file is passed in as a parameter to this call. The field that contains each project’s name in the DataSet that is returned is called “Name”, so the “DataTextField” for the drop-down control is set to the “Name” column. To uniquely identify each Project, the “ProjectId” column is used for the “DataValueField” of the drop-down control.

In a similar way, the “Reported By” and “Assigned To” drop-down controls are filled using the User Web Service.

Adding a Defect

In the “`Page_Load`” event of the DefectAdd.aspx file, you will see the following code:

```
if(_defect == null)
{
       _defect = new DefectService.Defect();
       _defect.CustomFieldValues = new
DefectService.DefectHandler().GetCustomFieldValuesList(new
Guid(System.Configuration.ConfigurationSettings.AppSettings["SecurityToken"]));
}
```

This code creates a new Defect object and calls a method to get a list of empty values for all custom fields associated with defects.

When the “Add Defect” link is clicked on the Add Defect web form, the “`addDefectLink_Click()`” method is called. We will walk through this code to explain how the Defect Web Service is used to add a new defect to the OnTime database:

```
_defect.AssignedToId = Convert.ToInt32(assignedToDropDown.SelectedItem.Value);
_defect.BuildNumber = defectBuildTextBox.Text;
_defect.BuildNumberOfFix = fixBuildTextBox.Text;

//In this sample, Date fields currently don't do validation of date.
// Also, "DateTime.MinValue" in OnTime means that this date field is not set
_defect.DateFound = (dateFoundTextBox.Text == "" ? DateTime.Now :
Convert.ToDateTime(dateFoundTextBox.Text));
_defect.DateFixed = (dateFixedTextBox.Text == "" ? DateTime.MinValue :
Convert.ToDateTime(dateFixedTextBox.Text));

_defect.Description = descriptionTextBox.Text;
_defect.Name = nameTextBox.Text;
_defect.Notes = notesTextBox.Text;
_defect.ProjectId = Convert.ToInt32(projectDropDown.SelectedItem.Value);
_defect.ReplicationProcedures = replicationTextBox.Text;
_defect.ReportedById = Convert.ToInt32(reportedByDropDown.SelectedItem.Value);
_defect.Resolution = resolutionTextBox.Text;
_defect.StatusTypeId = Convert.ToInt32(statusDropDown.SelectedItem.Value);
_defect.PriorityTypeId = Convert.ToInt32(priorityDropDown.SelectedItem.Value);
_defect.SeverityTypeId = Convert.ToInt32(severityDropDown.SelectedItem.Value);
```

The above lines start assigning values to the defect object based on the selections from the WebForm that was filled by the user. In order to keep the example as simple as possible, there is no form validation done on either the client or the server.

```
DefectService.DefectHandler defectHandler = new DefectService.DefectHandler();
if(_defect.DefectId == 0)
       defectHandler.AddDefect(new
Guid(System.Configuration.ConfigurationSettings.AppSettings["SecurityToken"]),
_defect, _defect.CustomFieldValues);
```

The above 2 lines are the meat of this method. The first line creates a “DefectHanlder” object which is nothing more than a proxy class that points to the OnTime Defect Web Service. The second line uses the “AddDefect” method of the defectHandler object to add the defect. The GUID, the defect object, and the list of custom field values are passed in to the method.
It’s that simple!

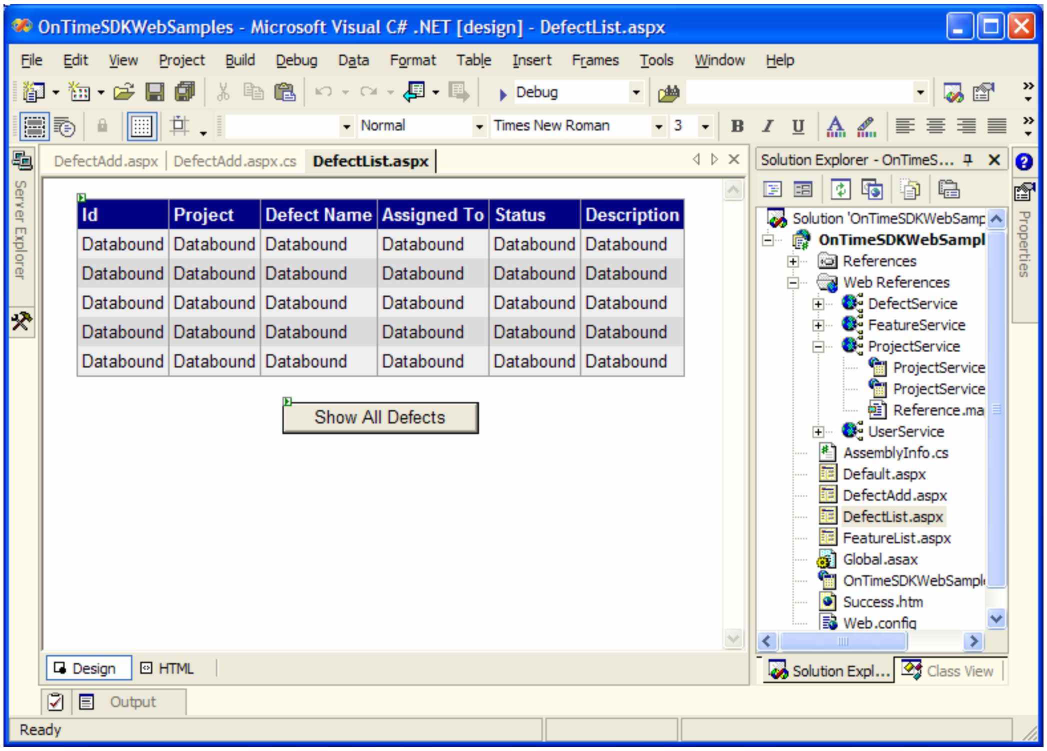
```
Response.Redirect("DefectList.aspx");
```

Finally, the page is redirected back to the list of defects.

You will notice that the code is set up to handle editing of defects as well. You can do this by clicking on the defect name link from the defects listing page. This will populate the edit page with the existing defect data and will update the data once the ‘Update Defect’ link is clicked.

Obtaining a List of Defects

The defect list web form is contained in the “DefectList.aspx” page. Open this page now in the designer view. Visual Studio.net should look something similar to this:



Id	Project	Defect Name	Assigned To	Status	Description
Databound	Databound	Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound	Databound	Databound
Databound	Databound	Databound	Databound	Databound	Databound

This page contains a DataGrid with 6 bound columns. The columns have been bound to the column names of a DataSet that is returned by the “GetAllDefectsWithNames()” method of the Defect Web Service. To see the code that calls the web service and obtains the defects DataSet, switch to code view and locate the code for “`showDefectsButton_Click()`” method. We will walk through that code here:

```
DefectService.DefectHandler defectHandler = new DefectService.DefectHandler();
DataSet ds = defectHandler.GetAllDefectsWithNames(new
Guid(System.Configuration.ConfigurationSettings.AppSettings["SecurityToken"]));
```

In the above lines of code, we first create a DefectHanlder object and a DataSet. We set the DataSet to the DataSet that is returned by the “GetAllDefectsWithNames()” method of the DefectHandler object. After these two lines, we have access to all the defects in our OnTime database. The defects are now stored in the locally available DataSet called “ds”.

```
defectsDataGrid.DataSource = ds;
defectsDataGrid.DataBind();
```

Finally, we set the DataSource of the DataGrid property to the DataSet that we just obtained.

Obtaining a List of Features

To obtain and list all the features in the OnTime database, the process is nearly identical to the process of obtaining and displaying a list of defects. The sample code is provided on how to perform this task (in the FeaturesList.aspx file), but will not cover the code in detail. For more information, see the section on “Obtaining a List of Defects.”

Windows Samples

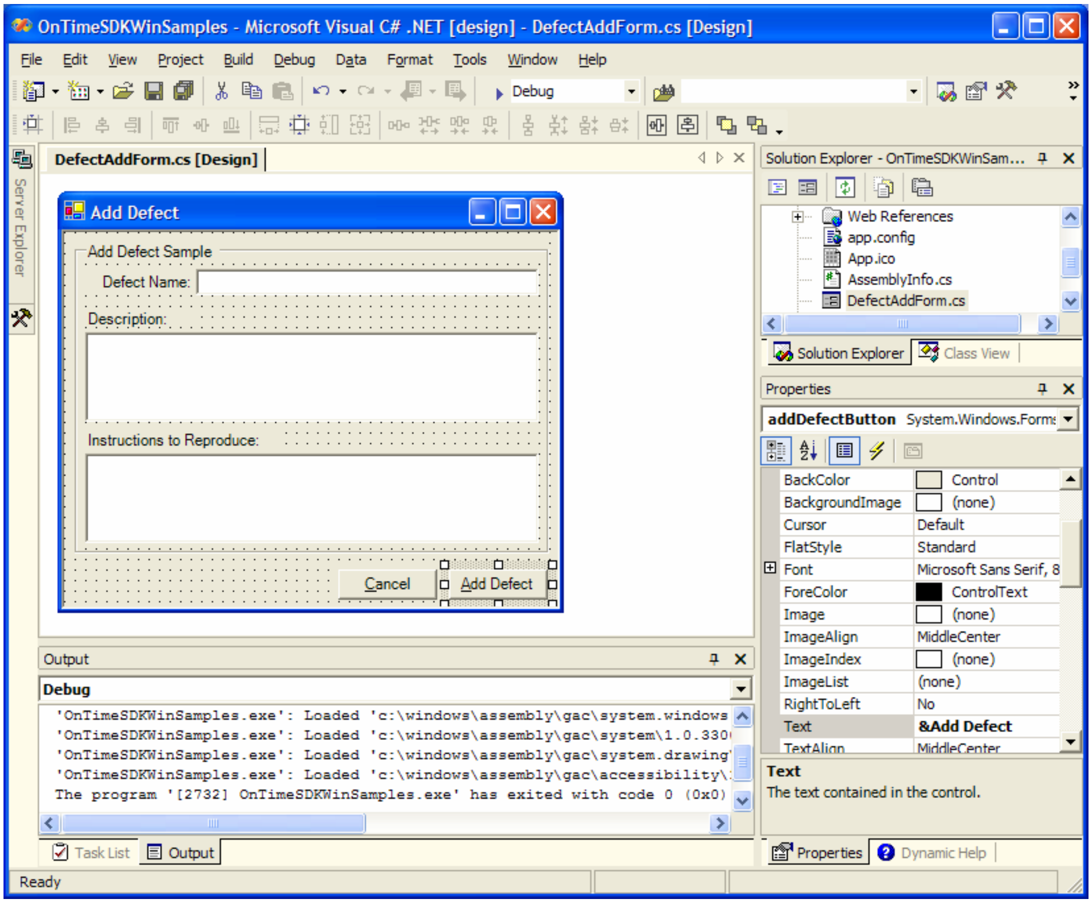
Introduction

Using Visual Studio.net, open the “OnTimeSDKWinSamples.csproj” project file (by default, it’s located at “C:\Program Files\MCPCentral\OnTime 2005 SDK Win Samples”). This project contains 3 Windows Forms:

- **MainForm** – This is a simple page with buttons to execute the other two pages.
- **DefectAddForm** – This form allows a user to quickly add a new defect with a small amount of information.
- **DefectListForm** – This form lists all the defects in the OnTime database and displays it using a DataGrid.

Adding a Defect

To start, open the “DefectAddForm.cs” file in design mode. Visual Studio.net should look something like this:


OnTimeSDKWinSamples - Microsoft Visual C# .NET [design] - DefectAddForm.cs [Design]
File Edit View Project Build Debug Data Format Tools Window Help
Debug
Server Explorer
DefectAddForm.cs [Design]
Add Defect
Add Defect Sample
Defect Name:
Description:
Instructions to Reproduce:
Cancel
Add Defect
Solution Explorer - OnTimeSDKWinSam...
Web References
app.config
App.ico
AssemblyInfo.cs
DefectAddForm.cs
Solution Explorer
Class View
Properties
Output
Task List
Output
Properties
Dynamic Help
Ready

addDefectButton System.Windows.Forms

BackColor	Control
BackgroundImage	(none)
Cursor	Default
FlatStyle	Standard
Font	Microsoft Sans Serif, 8
ForeColor	ControlText
Image	(none)
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageList	(none)
RightToLeft	No
Text	**&Add Defect**
TextAlign	MiddleCenter

Text
The text contained in the control.

Debug

```
'OnTimeSDKWinSamples.exe': Loaded 'c:\windows\assembly\gac\system.windows
'OnTimeSDKWinSamples.exe': Loaded 'c:\windows\assembly\gac\system\1.0.330
'OnTimeSDKWinSamples.exe': Loaded 'c:\windows\assembly\gac\system.drawing'
'OnTimeSDKWinSamples.exe': Loaded 'c:\windows\assembly\gac\accessibility\
The program '[2732] OnTimeSDKWinSamples.exe' has exited with code 0 (0x0)
```

The add defect form has been created with very few fields to keep the example short and simple. To see what happens when the “Add Defect” button is clicked, locate the code for the “`addDefectButton_Click()`” method. We will walk through this code now:

```
DefectService.Defect defect = new DefectService.Defect();
```

The above line creates a new Defect object that will be used to hold the defect information.

```
defect.Description = descriptionTextBox.Text;
defect.ReplicationProcedures = reproduceTextBox.Text;
defect.Name = defectNameTextBox.Text;
// There are many other defect properties which we are not setting
```

As indicated in the above comments, only 3 properties of the defect object are set. This simple example also shows that you don't need to set every property in order to add a new defect. In practice, this type of simple form might be used to allow customers to report a defect without forcing the customer to provide too much information.

```
DefectService.DefectHandler defectHandler = new DefectService.DefectHandler();

defectHandler.AddDefect(new
Guid(System.Configuration.ConfigurationSettings.AppSettings["SecurityToken"]),
defect, defect.CustomFieldValues);
this.Close();
```

Finally, using the "DefectHandler" proxy class we add the defect to the OnTime database by calling the "AddDefect" web method.

Web Service APIs

See the OnTime SDK help file for reference information.