



iSite Enterprise SDK Tutorials

iSite PACS 3.6 and 4.1

Rev. 1.02
2008 Oct 28



Enterprise Imaging

iSite Enterprise SDK Tutorials

iSite PACS 3.6 and 4.1

Rev. 1.02

2008 Oct 28

Philips Healthcare

is part of
Royal Philips Electronics
www.medical.philips.com
medical@philips.com

North America

Philips Healthcare Informatics
Radiology Informatics Business Unit
4100 East Third Ave., Suite 101
Foster City, CA 94404
USA

Europe

Philips Medical Systems Nederland B.V.
PMS Quality & Regulatory Affairs Europe
Veenpluis 4-6
5684 PC Best
The Netherlands

Copyright © 2007 Koninklijke Philips Electronics N.V.
All rights reserved. iSite and iSyntax are registered trademarks of Koninklijke Philips Electronics N.V.

TMP4.05-02 2008 Apr 28

Table of Contents

Overview	2
Related Documents.....	2
1 iSite Enterprise ActiveX API C++ Tutorial.....	3
1.1 Adding the Controls to a Dialog-Based Application.....	3
1.1.1 Using the Control in a Dialog-Based Application	4
1.1.2 Initializing and Using the Controls	7
1.1.3 Logging in and Displaying an Exam	8
2 iSite Enterprise ActiveX API Tutorial Visual Basic Sample.....	13
2.1 Adding the Controls to a Visual Basic Application in Visual Studio 6.....	13
2.1.1 Using the Control in the Application.....	13
2.1.2 Initializing and Using the Controls	14
2.1.3 Logging in and Displaying an Exam	16
3 iSite Enterprise ActiveX API Tutorial VBScript Sample.....	20
3.1 Adding the Controls to a Web Page.....	20
3.1.1 Initializing and Using the Controls	21
3.2 Programmatically Logging in and Displaying an Exam	22
3.2.1 Programmatically Logging in to the iSite PACS Server	23
4 iSite Enterprise ActiveX API Tutorial JavaScript Sample.....	26
4.1 Adding the Controls to a Web Page.....	26
4.1.1 Initializing and Using the Controls	27
4.1.2 Programmatically Logging in and Displaying an Exam.....	29

Overview

The iSite Enterprise application consists of three ActiveX controls packaged as a single ocx. The first control is a non-visual control and allows the user to interact directly with the iSite PACS Server. The second control is a visual control that contains the user interface for the iSite Enterprise application. The third and final control is a single image control that contains a user interface for single image display.

Note: The **iSiteEnterprise** control depends upon the non-visual control and will not operate until the non-visual control has been initialized. You must first initialize the non-visual control before initializing and using the **iSiteEnterprise** control.

StentorBackEnd Option: Refer to the *iSite PACS Client and API Overview* document for information on how to properly use the **StentorBackEnd** option.

The following tutorials provide information on how to embed iSite Enterprise in an application:

- ActiveX API C++
- ActiveX API Visual Basic Sample

The following tutorials provide information on how to embed iSite Enterprise in a Web page:

- ActiveX API VBScript Sample
- ActiveX API JavaScript Sample

Related Documents

Please read this and supporting documentation before beginning development with the API. See the following documents for detailed information for all API events and methods:

- *iSite Enterprise ActiveX Reference Guide* for version 4.1/3.6
- *iSite Radiology ActiveX Reference Guide* for version 4.1/3.6

See the following document for details concerning API compatibility between different versions of iSite PACS:

- *iSite PACS API Version Information* for version 4.1/3.6
- *iSite PACS Client and API Overview* for version 4.1/3.6

1 iSite Enterprise ActiveX API C++ Tutorial

This tutorial will show how to add the **iSiteEnterprise** ActiveX controls to a MFC/C++ based project and to perform the basic functions provided by the controls. Knowledge of Visual Studio, MFC and C++ is assumed. This tutorial is intended to help you get started using the iSite Enterprise ActiveX controls. Please use the reference guides and the sample applications provided with the SDK for more details. This is the Visual C++ example.

1.1 Adding the Controls to a Dialog-Based Application

Create a dialog-based application using Visual Studio. See the Visual Studio documentation for more details if you are unfamiliar with dialog-based applications.

Adding active x controls to the project:

1. On the **Project** menu, click **Add to Project**. A shortcut menu appears.
2. Click **Components and Controls**. The **Components and Controls Gallery** dialog box appears.
3. Click on the **iSiteNonVisual** control. Visual C++ displays a dialog box from which you can select a subset of the components classes to add to the project.
4. Click **OK** to add the component.
5. Visual C++ generates the **iSiteNonVisual.h** and **iSiteNonVisual.cpp** file for the component and adds them to the project.
6. Repeat the above for the **iSiteEnterprise** control.

Two new classes added to the project, **iSiteNonVisual** and **iSiteEnterprise**. Two new icons have also been added to the controls toolbar of the dialog editor:



1.1.1 Using the Control in a Dialog-Based Application

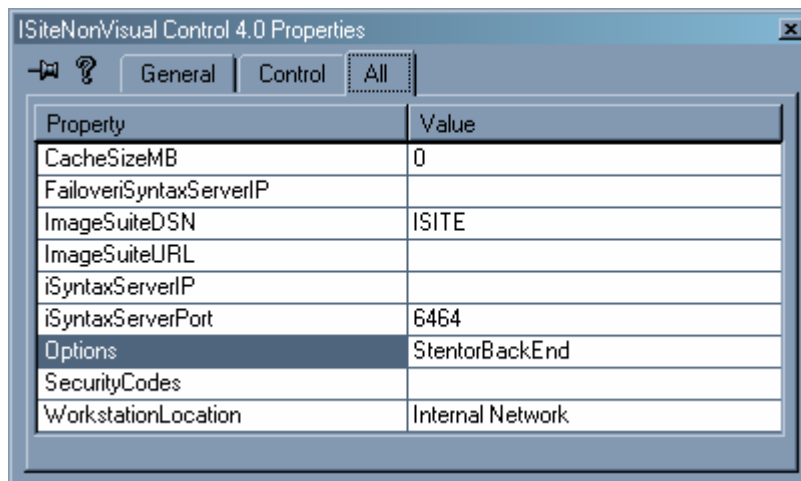
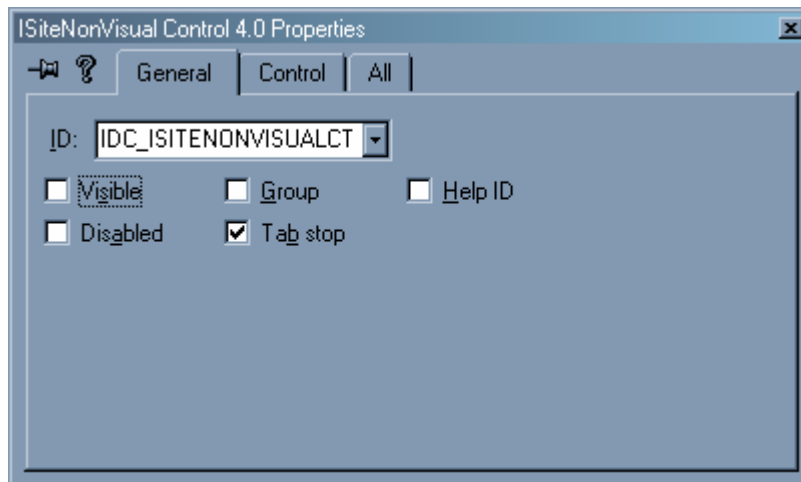
Use the iSite PACS ActiveX controls as you would any other control. Simply select the control on the toolbar and add it to the dialog as shown below. You must add the **iSiteNonVisual** control before adding the **iSiteEnterprise** control.

Important: The **iSiteNonVisual** control must be added first before adding the **iSiteEnterprise** control. Otherwise initialization errors may result.

The screenshot shows a Windows-style dialog box titled "iSite Enterprise Sample Application". It contains two main sections. The top section, "Initialization Properties", has six input fields: "Server IP:", "ImageSuite URL:", "Server Port:", "User Name:", "Password:" (with masked characters "xxxx"), and "AuthSource:". Each field has an "Edit" button next to it. To the right of this section is a white rectangular area labeled "iSiteNonVisualCtrl (Design Mode)". The bottom section, "iSite Enterprise", has three input fields: "Patient ID:", "Accession ID:", and "Organization:", each with an "Edit" button. To the right of these fields is a "Show Exam" button.

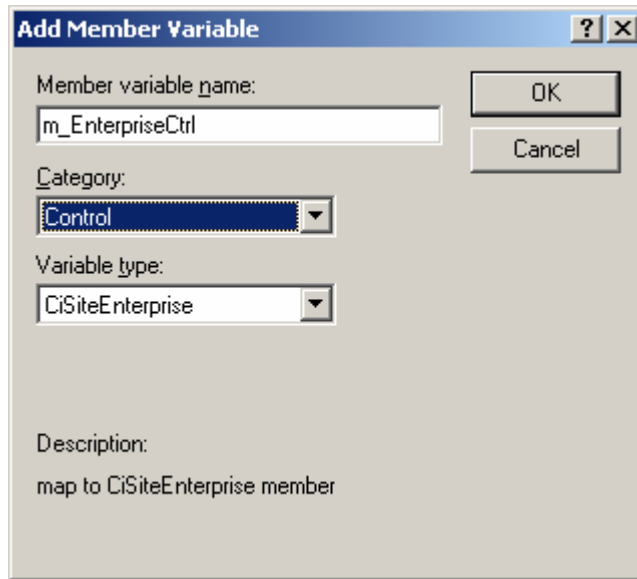
Remember that there can only be one **iSiteNonVisual** control per process.

Note: It is best to make the **iSiteNonVisual** control not visible by unselecting the **Visual** checkbox in the control.

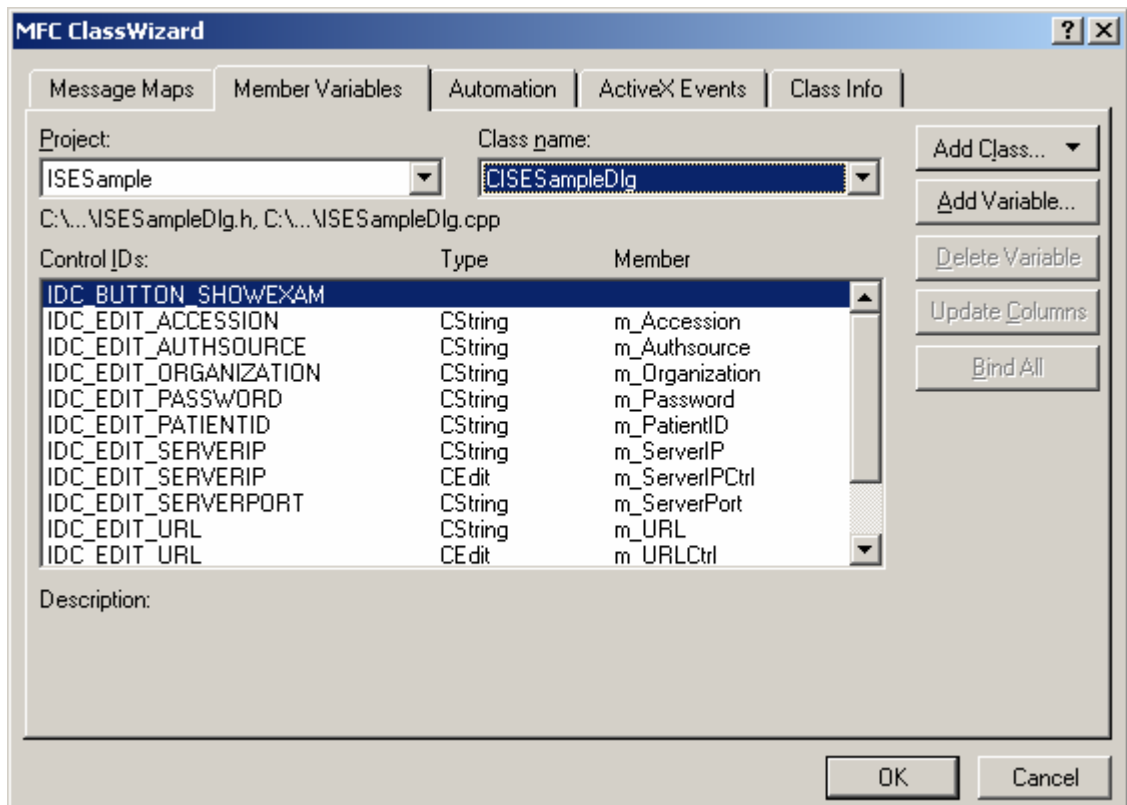


Use the class wizard to add member variables to connect the class to the control:

1. Open the **ClassWizard** by selecting the **ClassWizard** menu item in the **View** menu.
2. Highlight the Control ID for the **iSiteEnterprise** control.
3. Click the **Add Variable...** button.
4. Set the variable name, category and type as follows. Click **OK**.



Repeat the above for the **iSiteNonVisual** control. You should have two member variables similar to what is show below. Click **OK** to accept the additions.



1.1.2 Initializing and Using the Controls

The **iSiteNonVisual** control must be initialized before the **iSiteEnterprise** control may be used. In order to initialize the control you must have the following information set beforehand.

Property	Function
Server IP address	void SetISyntaxServerIP(LPCTSTR);
Server Port Number	void SetISyntaxServerPort(LPCTSTR);
ISuite URL	void SetImageSuiteURL(LPCTSTR);

Notice the following code in the `InitializeNonVisualCtrl()` function:

```
BOOL CISESampleDlg::InitializeNonVisualCtrl()
{
    BOOL bRet = false;

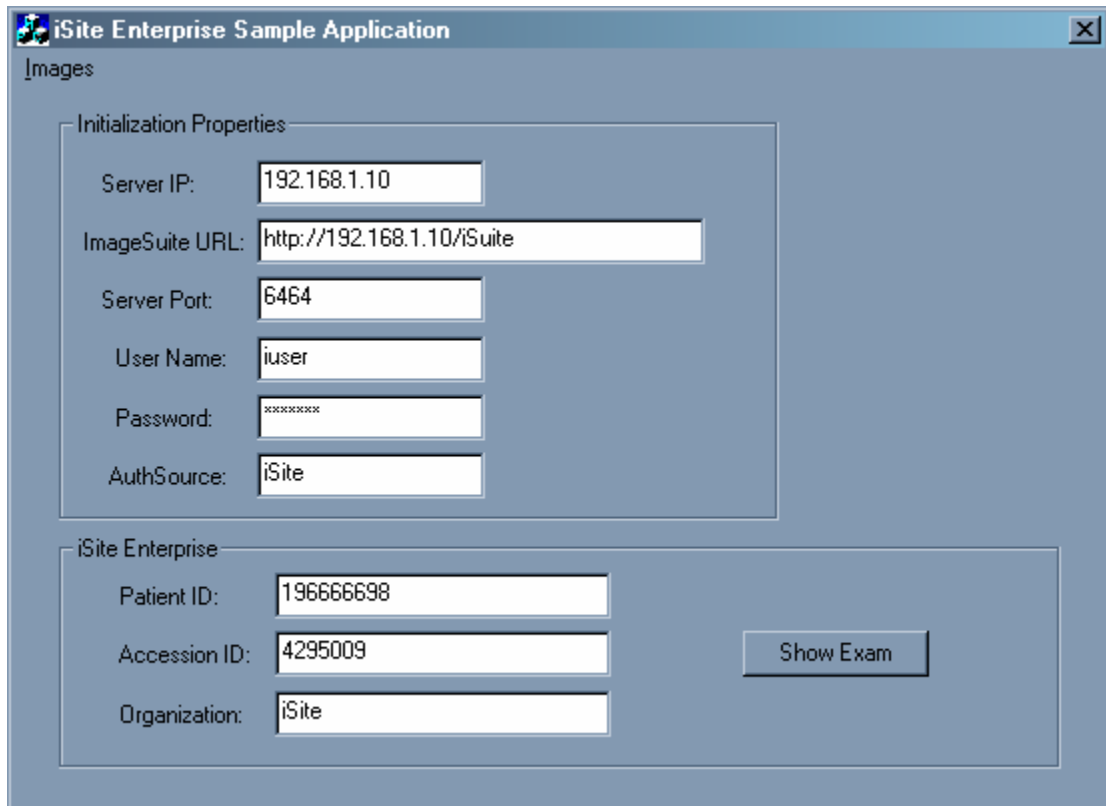
    // already initialized?
    if (m_NonVisualCtrl.GetInitialized())
        bRet = true;
    else
    {
        // Set the required parameters
        m_NonVisualCtrl.SetISyntaxServerIP(m_ServerIP);
        m_NonVisualCtrl.SetISyntaxServerPort(m_ServerPort);
        m_NonVisualCtrl.SetImageSuiteURL(m_URL);
        m_NonVisualCtrl.SetImageSuiteDSN("iSite");

        // Give the initialize a try
        bRet = m_NonVisualCtrl.Initialize();

        if(!bRet)
        {
            long lError = m_NonVisualCtrl.GetLastErrorCode();
            AfxMessageBox("Error Initializing NonVisual Control");
        }
    }

    return bRet;
}
```

Build and run the application. You should see the following:



The screenshot shows a window titled "iSite Enterprise Sample Application" with a standard Windows-style title bar. The window is divided into two main sections. The top section, labeled "Images", contains a group box titled "Initialization Properties". This group box has six text input fields: "Server IP:" with the value "192.168.1.10", "ImageSuite URL:" with the value "http://192.168.1.10/iSuite", "Server Port:" with the value "6464", "User Name:" with the value "iuser", "Password:" with the value "xxxxxxx", and "AuthSource:" with the value "iSite". The bottom section contains a group box titled "iSite Enterprise". This group box has three text input fields: "Patient ID:" with the value "196666698", "Accession ID:" with the value "4295009", and "Organization:" with the value "iSite". To the right of these fields is a button labeled "Show Exam".

1.1.3 Logging in and Displaying an Exam

Once the controls have been successfully integrated you may now start using the API's.

The **iSiteNonVisual** control provides access to the iSite PACS Server application and must be initialized before using the **iSiteEnterprise** control. Only one instance of the **iSiteNonVisual** control is allowed per process. Attempting to initialize more than one instance will fail. It is recommended to create a single instance of the **iSiteNonVisual** control when your application starts up, and destroy that control when your application terminates.

The **iSiteEnterprise** control contains the user interface for the application. It depends upon the **iSiteNonVisual** control for operation. Before interacting with the **iSiteEnterprise** control, you must initialize the **iSiteNonVisual** control. If a user has not logged in programmatically, (via the **iSiteNonVisual::Login** method), a login screen will be presented to the user.

1.1.3.1 Logging in to the iSite PACS Server

Use these functions to log into and out of the iSite PACS Server:

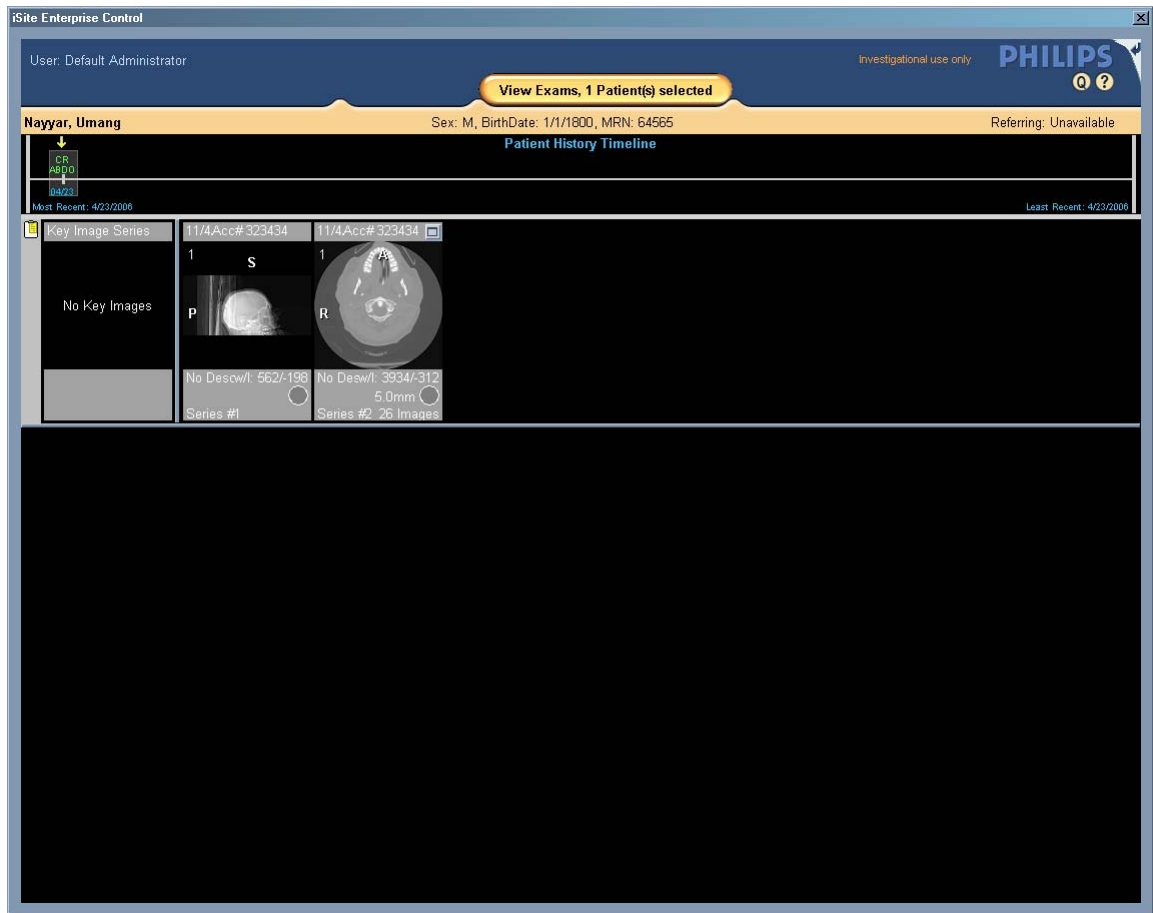
```
boolean Login(BSTR UserName, BSTR Password,  
             BSTR AuthSource,  
             BSTR Token,  
             BSTR Mnemonic);  
  
boolean Logout();
```

To log out is very simple. Just call the `Logout()` function. You may create another button to perform this task

For Example.

```
void CMyDlg::OnButtonLogout()  
{  
    if(m_NonVisual.Logout() == FALSE)  
    {  
        AfxMessageBox("Error logging out");  
    }  
}
```

Build and run the application. Click **Show Exam** and you should see the following:



Note: You may prevent the UI components from appearing by setting the Option property with the `SetOptions()` function. You must set this option before initializing the **iSiteNonVisual** control.

For example, `m_NonVisual.SetOptions("HideFullUserName");`

See the "iSiteEnterprise" Control section in the *API Reference Guide* for more details on how to set these options.

1.1.3.2 Displaying an Exam

Displaying an exam is easy if you know the exam ID. Most of the time, you will only know the patient and accession IDs. The **iSiteNonVisual** control has functions to help you find the exam IDs.

We will use these functions to find and open an exam given the patient and accession ID:

```
iSiteNonVisual::
CString FindExam(LPCTSTR Accession, LPCTSTR PatientID, LPCTSTR
Organization)

iSiteEnterprise::OpenCanvasPage(BSTR IntExamID,
                                BSTR IntExceptionID,
                                boolean Reveal,
                                boolean Lock,
                                boolean OpenNew);
```

For example:

```
void CISESampleDlg::OpenExam()
{
    CString ExamID;

    // find the exam id based on patient id & accession
    ExamID = m_NonVisualCtrl.FindExam(m_Accession, m_PatientID,
m_Organization);

    // Did we found it?
    if (!ExamID.IsEmpty())
    {
        CISEDisplayDlg ISEDlg;

        // pass the exam id & nonvisual ctrl over to the dialog and
display the exam
        // the dialog will take care of opening and closing it.
        ISEDlg.m_ExamID = ExamID;
        ISEDlg.m_pNonVisualCtrl = &m_NonVisualCtrl;
        ISEDlg.DoModal();
    }
    else
    {
        AfxMessageBox("Error Finding Exam");
    }
}
```

```
BOOL CISEDdisplayDlg::OpenCanvasPage()
{
    BOOL bRet = false;

    // just a sanity check
    if (!m_ExamID.IsEmpty())
    {
        m_canvasPageID = m_EnterpriseCtrl.OpenCanvasPage(m_ExamID, "",
TRUE,
FALSE, TRUE);
        if (m_canvasPageID.IsEmpty())
        {
            AfxMessageBox("Error Opening CanvasPage");
        }
        else
            bRet = true;
    }

    return bRet;
}
```

Build and run the application, and walk through the member functions in the debugger to help you understand how the application and the API works. After familiarizing yourself with the API, you will find that you can create powerful custom applications to suit your own needs.

2 iSite Enterprise ActiveX API Tutorial Visual Basic Sample

This tutorial will show how to add the iSite Enterprise ActiveX controls to a Visual Basic project and to perform the basic functions provided by the controls. Knowledge of Visual Basic is assumed. This tutorial is intended to help you get started using the iSite Enterprise ActiveX controls. Please use the reference guides and the sample applications provided with the SDK for more details. This example is in the ISESample_VisualBasic directory in the SDK. It is assumed that the iSite Enterprise client has already been installed on the development workstation.

2.1 Adding the Controls to a Visual Basic Application in Visual Studio 6

Create a Standard EXE application using Visual Basic.

Adding the ActiveX controls to the project:

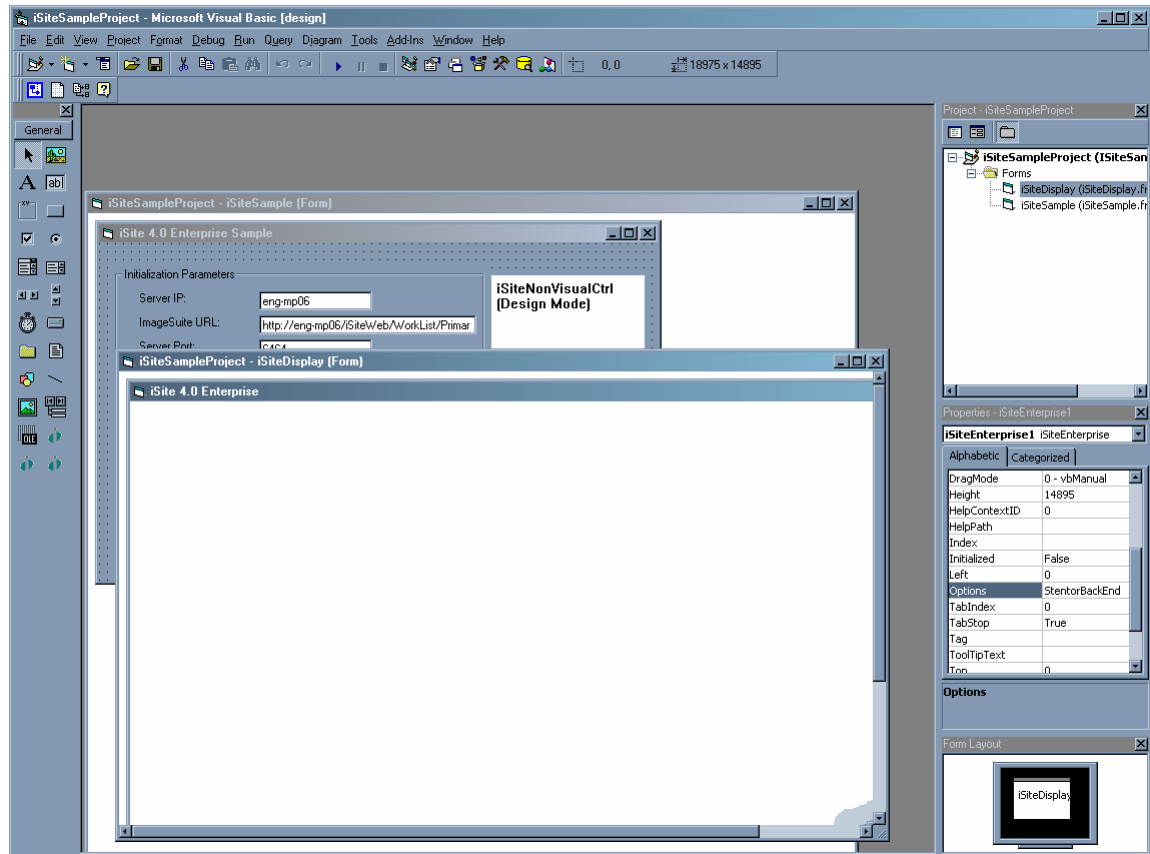
1. On the **Project** menu, click **Components....** The Components dialog will appear.
2. Check the appropriately versioned iSite PACS control. The iSite PACS Control version must match the Major/Minor iSite PACS version installed on the server that will be connected to.
3. Click **OK** to add the component.

Two new classes added to the project, **iSiteNonVisual** and **iSiteEnterprise**. Two new icons have also been added to the controls toolbar.

2.1.1 Using the Control in the Application

Use the iSite ActiveX controls as you would any other control. Simply select the control on the toolbar and add it to the form as shown below. You must add the **iSiteNonVisual** control before adding the **iSiteEnterprise** control.

Important: The **iSiteNonVisual** control must be added first before adding the **iSiteEnterprise** control. Otherwise, there may be initialization errors.



Remember that there can only be one **iSiteNonVisual** control per process.

2.1.2 Initializing and Using the Controls

The **iSiteNonVisual** control must be initialized before the **iSiteEnterprise** control may be used. In order to initialize the control you must have the following information set beforehand:

Property	Attribute
Server IP address	iSyntaxServerIP
Server Port Number	iSyntaxServerPort
iSuite URL	ImageSuiteURL

Place the following code into the Form_Load():

```
' already initialized?
If iSiteNonVisual1.Initialized = True Then
    InitializeNonVisualCtrl = True
Else
    ' Set the required parameters
    iSiteNonVisual1.iSyntaxServerIP = ServerIP.Text
    iSiteNonVisual1.iSyntaxServerPort = ServerPort.Text
    iSiteNonVisual1.ImageSuiteURL = URL.Text
    iSiteNonVisual1.ImageSuiteDSN = "iSite"

    ' Set these default options
    iSiteNonVisual1.Options =
    "StentorBackEnd,HideLogoutButton,HideCloseTabButton "

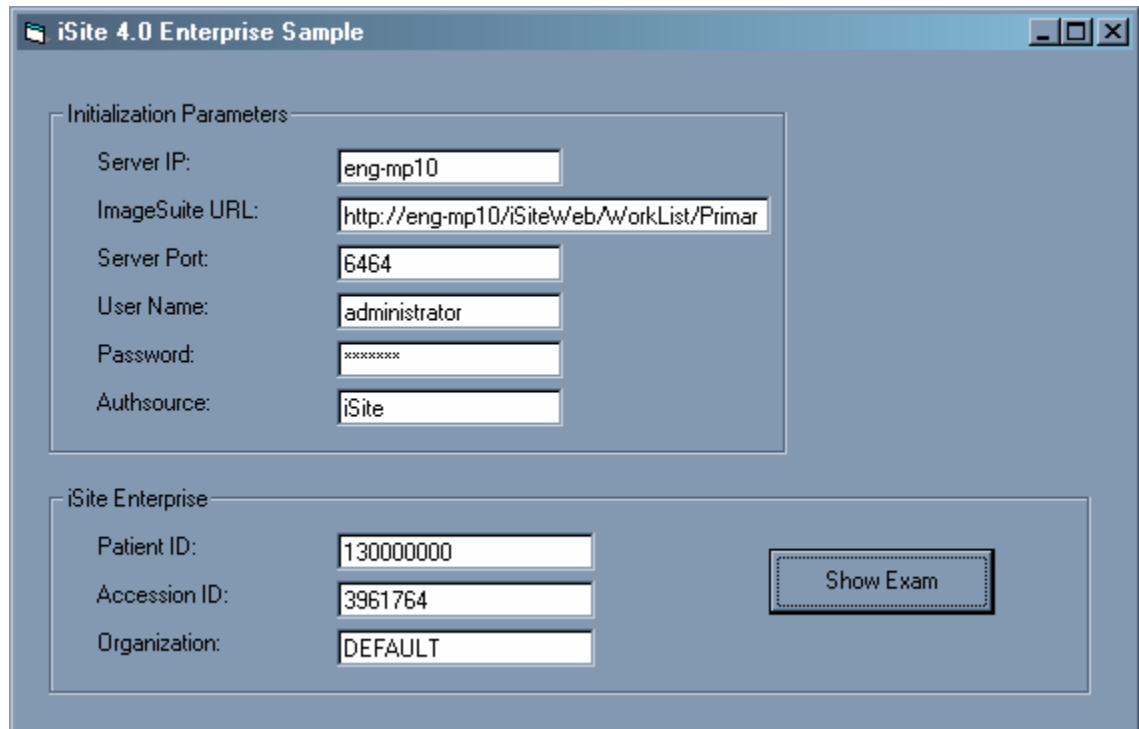
    ' Initialize
    InitializeNonVisualCtrl = iSiteNonVisual1.Initialize()

    If InitializeNonVisualCtrl <> True Then
        MsgBox "Error Initializing NonVisual Control"
    End If
End If

' Set these default options
iSiteEnterprise1.Options =
"StentorBackEnd,DisableAutoLogout,HideFolders,HideCloseTabButton,DisablePr
eferences"

' Initialize the visual control
InitializeEnterpriseCtrl = iSiteEnterprise1.Initialize(iSiteNonVisual1)
If InitializeEnterpriseCtrl = False Then
    MsgBox "Error Initializing Enterprise Control"
End If
```

Run the application. You should see the following:



The screenshot shows a Windows-style application window titled "iSite 4.0 Enterprise Sample". The window is divided into two main sections. The top section, titled "Initialization Parameters", contains six text input fields: "Server IP:" with the value "eng-mp10", "ImageSuite URL:" with the value "http://eng-mp10/iSiteWeb/WorkList/Primar", "Server Port:" with the value "6464", "User Name:" with the value "administrator", "Password:" with masked characters "xxxxxxxx", and "Authsource:" with the value "iSite". The bottom section, titled "iSite Enterprise", contains three text input fields: "Patient ID:" with the value "130000000", "Accession ID:" with the value "3961764", and "Organization:" with the value "DEFAULT". To the right of the "Patient ID:" field is a button labeled "Show Exam".

2.1.3 Logging in and Displaying an Exam

Once the controls have been successfully integrated you may now start using the API's.

The **iSiteNonVisual** control provides access to the iSite PACS Server application and must be initialized before using the **iSiteEnterprise** control. Only one instance of the **iSiteNonVisual** control is allowed per process. Attempting to initialize more than one instance will fail. It is recommended to create a single instance of the **iSiteNonVisual** control when your application starts up, and destroy that control when your application terminates.

The **iSiteEnterprise** control contains the user interface for the application. It depends upon the **iSiteNonVisual** control for operation. Before interacting with the **iSiteEnterprise** control, you must initialize the **iSiteNonVisual** control. When a user is not logged in programmatically (via the **iSiteNonVisual::Login** method) a login screen will appear.

2.1.3.1 Logging in to the iSite PACS Server

Use these functions to log into and out of the iSite PACS Server:

```
boolean Login(UserName, Password, Authsource, Token, Mnemonic)
```

```
boolean Logout()
```

For example, add a button to the form and add the following code:

```
Private Sub Login_Click()  
    Dim SecurityCodes  
    Login = False  
    ' test if we are logged in already  
    If iSiteNonVisual1.SecurityCodes = "" Then  
        Login = iSiteNonVisual1.Login(UserName.Text, Password.Text,  
"iSite", "", "")  
        If Login = False Then  
            MsgBox "Error logging in"  
        End If  
    Else  
        Login = True  
    End If  
End Sub
```

To log out is very simple. Just call the Logout() function. You may create another button to perform this task.

For example:

```
Function Logout()  
    Logout = iSiteNonVisual1.Logout()  
    If Logout = False Then  
        MsgBox "Error logging out"  
    End If  
End Function
```

Note: You may control the appearance of the iSite Enterprise control by setting the Options property.

See the “iSite Enterprise Control” section in the *API Reference Guide* for more details on how to set these options.

2.1.3.2 Displaying an Exam

Displaying an exam is easy if you know the exam ID. Most of the time, you will only know the patient and accession IDs. The **iSiteNonVisual** control has functions to help you find the exam ids.

We will use these functions to find and open an exam given the patient and accession ID:

iSiteNonVisual Control:

```
FindExam(Accession, PatientID, Organization)
```

iSiteEnterprise Control:

```
OpenCanvasPage(IntExamID, IntExtceptionID, Reveal, Lock, OpenNew)
```

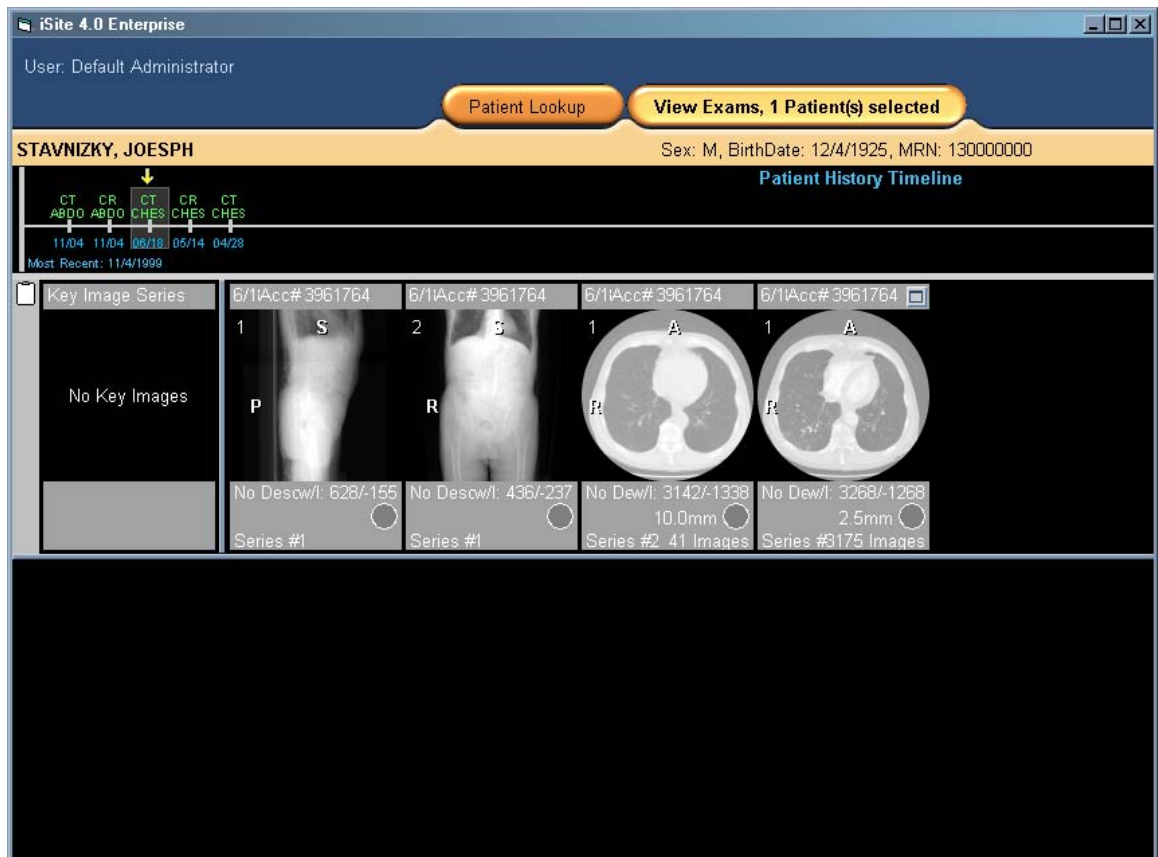
As an example add the following code after logging in:

```
' Use non-visual control to get exam id using patient & accession IDs
Accession = "ABC1234";
PatientID = "003-123-45-6789";
Organization = "iSite";

' Find the exam id based on patient id & accession
ExamID = iSiteNonVisual1.FindExam(Accession, PatientID, Organization)

' Did we find it?
If ExamID <> "" Then
    CanvasPageID = iSiteEnterprise1.OpenCanvasPage(ExamID, "", True,
False, True)
    If CanvasPageID = "" Then
        MsgBox "Error opening the canvas page. The query was
invalid!"
    End If
End If
```

Run the application. You should see the exam that you have specified. Be sure that the IDs and organization are correct. An example is shown below:



3 iSite Enterprise ActiveX API Tutorial VBScript Sample

This tutorial will show how to add the iSite Enterprise ActiveX controls to a Web page using VBScript and to perform the basic functions provided by the controls. Knowledge of HTML and VBScript is assumed. This tutorial is intended to help you get started using the iSite Enterprise ActiveX controls. Please use the reference guides and the sample applications provided with the SDK for more details.

3.1 Adding the Controls to a Web Page

Remember that there can only be one **iSiteNonVisual** control per process.

Note: It is best to make the **iSiteNonVisual** control not visible by setting the size to zero. You may alternately set the display attribute of the object's style to "none". See documentation on style sheets for more information.

If you use a visual environment you may graphically add the controls to your Web page. Otherwise you can use the code provided here. Place this object in the body of your Web page. Note the width and height parameters are set to 0%. This will prevent any visual display of the **iSiteNonVisual** control. See the note above. It may be necessary to update the control's **ClassID** (in red below) to match the appropriate iSite Client version. Refer to the *iSite PACS API Version Information* document for the appropriate **ClassID**.

```
<body>

<p>
<OBJECT id=iSiteNonVisual
    CLASSID = "clsid:DD0E6EAF-2822-4d42-A2CF-7FED75DA06CA"
    width=0%
    height=0%>
</OBJECT>
</p>

</body>
```

Use the below HTML to place the iSiteEnterprise control in your Web page. It may be necessary to update the control's ClassID to match the appropriate iSite Client version.

```
<body>
```

```
<p>
<OBJECT id=iSite
  CLASSID="clsid:F21CDD6E-3FE2-4d78-8709-83D5D939B7B2"
  width=100%
  height=100%
  <PARAM NAME="Options" VALUE="">
</OBJECT>
</p>

</body>
```

Note: Refer to MSDN article <http://msdn2.microsoft.com/en-us/library/ms537508.aspx> on techniques to eliminate the “Click to activate and use this control” tooltip message when embedding the iSite Enterprise visual control in a Web application. The page ISEDIgVBScript.htm has been implemented following the guidelines in this article.

3.1.1 Initializing and Using the Controls

The **iSiteNonVisual** control must be initialized before the **iSiteEnterprise** control may be used. In order to initialize the control you must have the following information set beforehand:

Property	Attribute
Server IP address	iSyntaxServerIP
Server Port Number	iSyntaxServerPort
ISuite URL	ImageSuiteURL

The following code can be used to initialize the **iSiteNonVisual** control:

```
'Init the non-visual control
iSiteNonVisual.iSyntaxServerIP = svrip.value
iSiteNonVisual.iSyntaxServerPort = svrport.value
iSiteNonVisual.ImageSuiteURL = svrurl.value

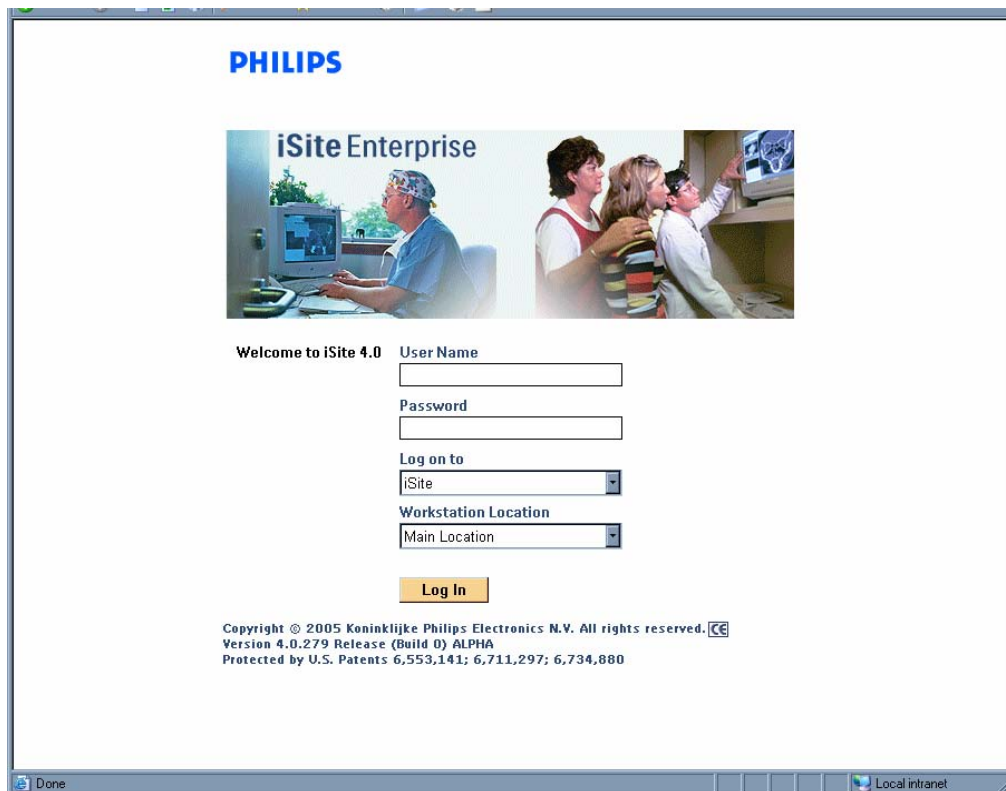
' Set these default options
iSiteNonVisual.Options =
"StentorBackEnd,DisableAutoLogout,HideLogoutButton,HideCloseTabButton,Disa
blePreferences"

' Always initialize the non-visual control first
If iSiteNonVisual.Initialize() Then
  bInitialized = 1
Else
  MsgBox strInitError_Message
```

```
End If
```

```
'Init the iSite Enterprise control only after initializing the  
iSiteNonVisual  
If iSite.Initialize(iSiteNonVisual) Then  
    'Post successful init code here  
End If
```

Run the Web page. You should see the following:



3.2 Programmatically Logging in and Displaying an Exam

Once the controls have been successfully integrated you may now start using the iSite PACS API.

The **iSiteNonVisual** control provides access to the iSite PACS Server application and must be initialized before using the **iSiteEnterprise** control. Only one instance of the **iSiteNonVisual** control is allowed per process. Attempting to initialize more than one instance will fail. It is recommended to create a single instance of the **iSiteNonVisual** control when your application starts up, and destroy that control when your application terminates.

The **iSiteEnterprise** control contains the user interface for the application. It depends upon the **iSiteNonVisual** control for operation. Before interacting with the **iSiteEnterprise** control, you must initialize the **iSiteNonVisual** control. If a user has not been logged in programmatically (via the **iSiteNonVisual::Login** method), a login screen will be presented to the user as shown above.

3.2.1 Programmatically Logging in to the iSite PACS Server

Use the Login functions to log into the iSite PACS Server. The security code will be an empty string if the user is not logged in.

```
Dim x

' security code will be empty if not logged in.
If iSiteNonVisual.SecurityCodes <> "" Or iSiteNonVisual.Login(user,
password, "iSite" "", "") Then
    ' show dialog with iSiteEnterprise control
    x = showModalDialog("iSEDlgVBScript.htm", window,

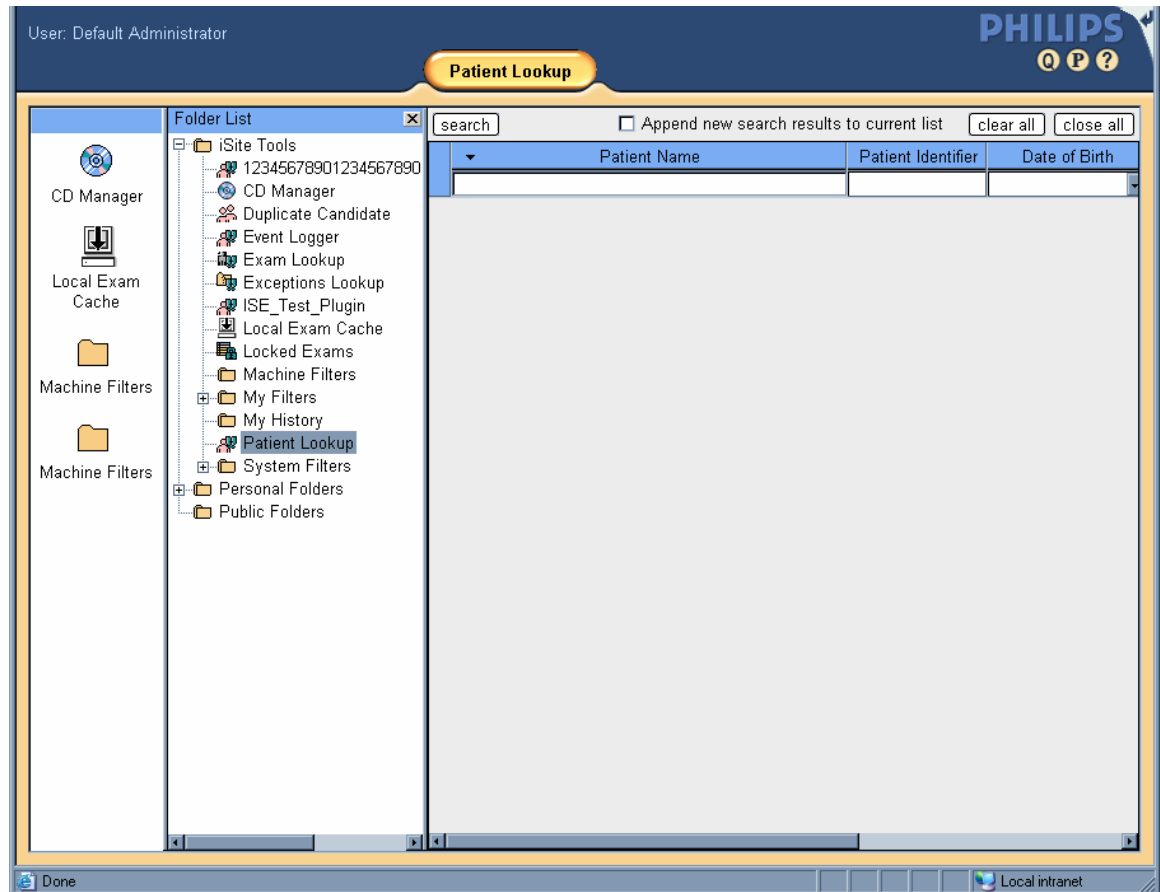
"status:no;resizable:yes;dialogWidth:1024px;dialogHeight:720px")
Else
    Dim errCode
    errCode = iSiteNonVisual.GetLastErrorCode()
    MsgBox errCode
End If
```

Note: See “CustomLogin” sample below if you wish to present your own log in screen. This sample is only available in JavaScript.

To log out is very simple. Just call the Logout() function:

```
iSiteNonVisual.Logout()
```

Run the Web page. Click **Login** and you should see the following:



Note: You may control the appearance of the iSite Enterprise control by setting the Options property.

See the “iSite Enterprise Control” section in the *API Reference Guide* for more details on how to set these options.

3.2.1.1 Displaying an Exam

Displaying an exam is easy if you know the exam ID. Most of the time, you will only know the patient and accession IDs. The **iSiteNonVisual** control has functions to help you find the exam ids.

Use these functions to find and open an exam given the patient and accession id:

```
iSiteNonVisual Control:
FindExam(Accession, PatientID, Organization)
iSiteEnterprise Control:
OpenCanvasPage(IntExamID, IntExceptionId, Reveal, Lock, OpenNew)
Dim examID;

examID = iSiteNonVisual.FindExam(acc.value, pid.value, org.value)
If examID <> "" Then
    Call iSite.OpenCanvasPage(examID, "", 1, 0, 1)
Else
    Dim errCode
    errCode = iSiteNonVisual.GetLastErrorCode()
    MsgBox errCode
End If
```

Open the page in an Internet Explorer browser window. You should see the exam that you have specified. Be sure that the IDs and the organization are correct.

4 iSite Enterprise ActiveX API Tutorial JavaScript Sample

This tutorial will show how to add the iSite Enterprise ActiveX controls to a web page using JavaScript and to perform the basic functions provided by the controls. Knowledge of HTML and JavaScript is assumed. This tutorial is intended to help you get started using the iSite Enterprise ActiveX controls. Please use the reference guides and the sample applications provided with the SDK for more details.

4.1 Adding the Controls to a Web Page

Remember that there can only be one **iSiteNonVisual** control per process.

Note: It is best to make the **iSiteNonVisual** control not visible by setting the size to zero. You may alternately set the display attribute of the object's style to "none". See documentation on style sheets for more information.

If you use a visual environment you may graphically add the controls to your web page. Otherwise you can use the code provided here. Place this object in the body of your web page. Note the width and height parameters are set to 0%. This will prevent any visual display of the **iSiteNonVisual** control. See the note above. Be sure to change the green IP Addresses to the IP Address or Server Name of your local iSite PACS server. It may also be necessary to update the control's **ClassID** to match the appropriate iSite PACS Client version. Refer to the *iSite PACS API Version Information* document for the appropriate **ClassID**.

```
<body>

<p>
<OBJECT id=iSiteNonVisual
        CLASSID = "clsid:DD0E6EAF-2822-4d42-A2CF-7FED75DA06CA"
        width=0%
        height=0%>
    </OBJECT>
</p>

</body>
```

Use the below HTML to place the **iSiteEnterprise** control in your web page. It may be necessary to update the control's **ClassID** to match the appropriate iSite PACS Client version.

```
<body>

<p>
<OBJECT id=iSite
    CLASSID="clsid:F21CDD6E-3FE2-4d78-8709-83D5D939B7B2"
    width=100%
    height=100%
    <PARAM NAME="Options" VALUE="">
</OBJECT>
</p>

</body>
```

Note: Refer to MSDN article <http://msdn2.microsoft.com/en-us/library/ms537508.aspx> on techniques to eliminate the "Click to activate and use this control" tooltip message when embedding the iSite Enterprise visual control in a web application.

4.1.1 Initializing and Using the Controls

The **iSiteNonVisual** control must be initialized before the **iSiteEnterprise** control may be used. In order to initialize the control you must have the following information set beforehand:

Property	Attribute
Server IP address	iSyntaxServerIP
Server Port Number	iSyntaxServerPort
ISuite URL	ImageSuiteURL

The following code can be used to initialize the **iSiteNonVisual** control:

```
// Init the non-visual control
iSiteNonVisual.iSyntaxServerIP = svrip.value;
iSiteNonVisual.iSyntaxServerPort = svrport.value;
iSiteNonVisual.ImageSuiteURL = svrurl.value;

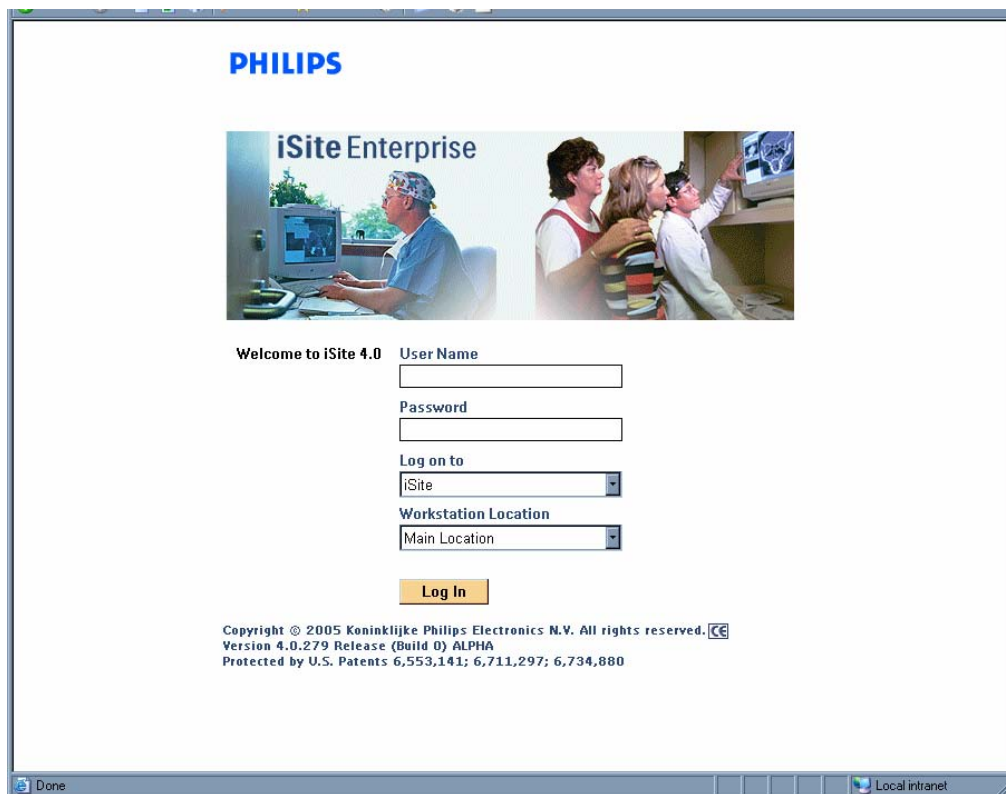
// Set these default options
iSiteNonVisual.Options="StentorBackEnd,DisableAutoLogout,HideLogoutButton,
HideCloseTabButton,DisablePreferences";

// Always initialize the non-visual control first
if (iSiteNonVisual.Initialize())
{
```

```
bInitialized = 1;
}
else
{
    alert(L_strInitError_Message)
}

// Init the iSite Enterprise control only after
// initializing the iSiteNonVisual
if (iSite.Initialize(iSiteNonVisual))
{
}
}
```

Run the web page. You should see the following:



4.1.2 Programmatically Logging in and Displaying an Exam

Once the controls have been successfully integrated you may now start using the iSite PACS API.

The **iSiteNonVisual** control provides access to the iSite PACS Server application and must be initialized before using the **iSiteEnterprise** control. Only one instance of the **iSiteNonVisual** control is allowed per process. Attempting to initialize more than one instance will fail. It is recommended to create a single instance of the **iSiteNonVisual** control when your application starts up, and destroy that control when your application terminates.

The **iSiteEnterprise** control contains the user interface for the application. It depends upon the **iSiteNonVisual** control for operation. Before interacting with the **iSiteEnterprise** control, you must initialize the **iSiteNonVisual** control. If a user has not been logged in programmatically (via the **iSiteNonVisual::Login** method), a login screen will be presented to the user as shown above.

4.1.2.1 Programmatically Logging in to the iSite PACS Server

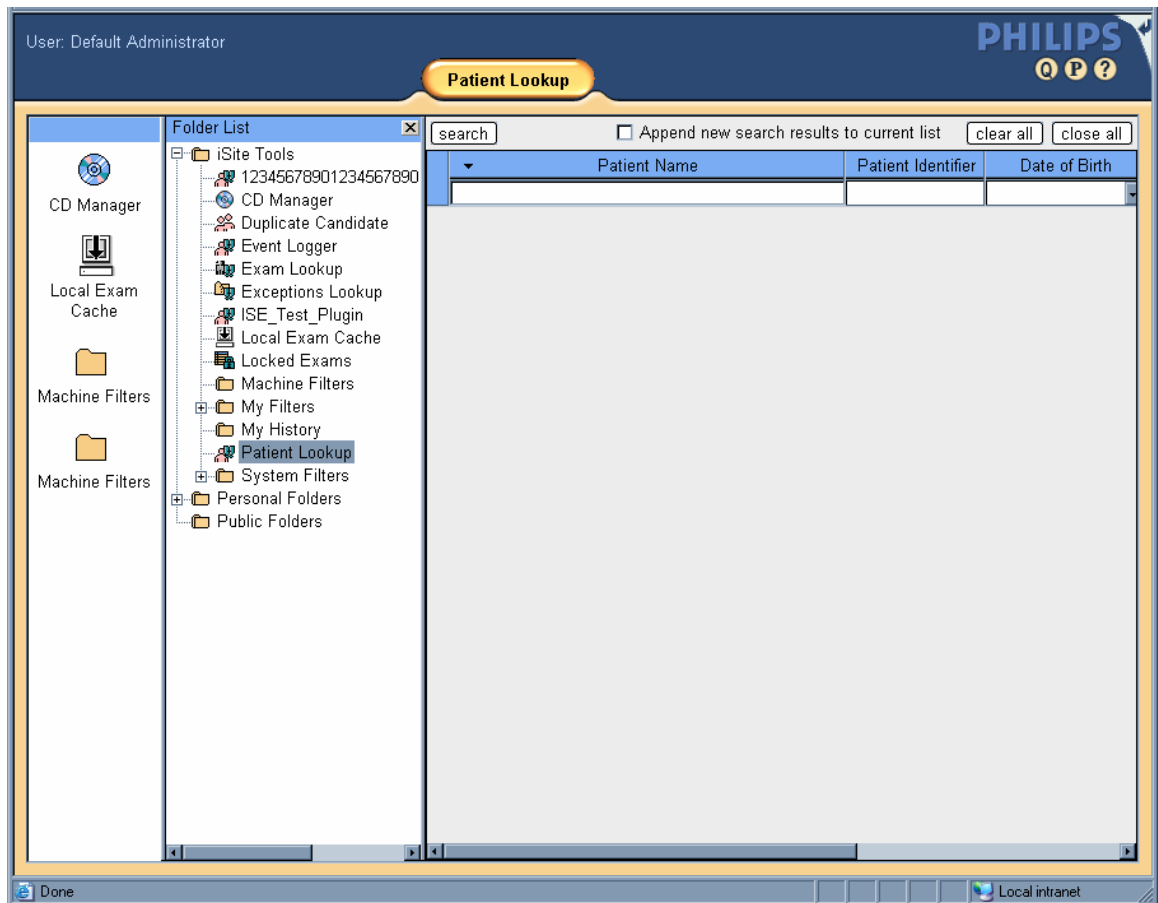
Use the Login functions to log into the iSite PACS Server. The security code will be an empty string if the user is not logged in.

```
// security code will be empty if not logged in.
if (iSiteNonVisual.SecurityCodes != "" || iSiteNonVisual.Login(user,
password, "iSite", "", ""))
{
    // do processing here
}
else
{
    alert(iSiteNonVisual.GetLastErrorCode())
}
```

To log out is very simple. Just call the Logout() function.

```
iSiteNonVisual.Logout();
```

Run the web page. Click **Login** and you should see the following:



Note: You may control the appearance of the iSite Enterprise control by setting the Options property.

See the “iSite Enterprise Control” section in the *API Reference Guide* for more details on how to set these options.

4.1.2.2 Displaying an Exam

Displaying an exam is easy if you know the exam ID. Most of the time, you will only know the patient and accession IDs. The **iSiteNonVisual** control has functions to help you find the exam ids.

Use these functions to find and open an exam given the patient and accession id.

iSiteNonVisual Control:

FindExam(Accession, PatientID, Organization)

iSiteEnterprise Control:

OpenCanvasPage(IntExamID, IntExceptionId, Reveal, Lock, OpenNew)

```
var examID = iSiteNonVisual.FindExam(window.dialogArguments.acc.value,  
window.dialogArguments.pid.value, window.dialogArguments.org.value);
```

```
if (examID != "")  
{  
    iSite.OpenCanvasPage(examID, "", 1, 0, 1);  
}  
else  
{  
    alert(iSiteNonVisual.GetLastErrorCode());  
}
```

Open the page in an Internet Explorer browser window. You should see the exam that you have specified. Be sure that the IDs and organization are correct.