Land Registry

Business Gateway

.NET implementation guide

Version 1.0

Information Systems
Seaton Court
2 William Prance Road
Plymouth
Devon
PL6 5WS

www.landregistry.gov.uk

In order to use the Business Gateway you'll need to have your certificates created and installed. This guide assumes this has already happened, if it has not, please contact your account manager. This guide talks through the creation of a basic C# Windows Form application. These steps work for a Web Application or an application in VB, using versions 3, 3.5 and 4 of the .Net framework. The project will connect to the Business Gateway and submit a request to the Daylist Enquiry service, and collect the response.

To check your certificate is installed, navigate to this URL:
https://bgtest.landregistry.gov.uk/b2b/BGStubService/DaylistEnquiryV2_0WebService?wsdl
If this is unsuccessful, your certificate is either incorrect or not installed. Please contact your account manager. If the wsdl is displayed, please continue.
Create a new C# Windows Form application
Add a button to the Form and add an On Click method. (This is just to give us something to trigger the submission)

Add a configuration file to the application. (On the Project menu, click Add New Item, select Application Configuration File).

Open the newly created app.config, add in this xml snippet:

```
<system.net>
	<defaultProxy useDefaultCredentials="true"/>
</system.net>
```

Now select Add Service Reference.
Enter the Service URL:
https://bgtest.landregistry.gov.uk/b2b/BGStubService/DaylistEnquiryV2_0WebService?wsdl
Give the Service Contract a suitable name and select ok.
This will add a load of values into your app.config file.
Replace the contents of the basicHttpBinding with the snippet below:

```
<binding name="CertPortBinding" closeTimeout="00:01:00" openTimeout="00:01:00"
receiveTimeout="00:10:00" sendTimeout="00:01:00" bypassProxyOnLocal="false"
hostNameComparisonMode="StrongWildcard" maxBufferSize="65536000" maxBufferPoolSize="524288"
maxReceivedMessageSize="65536000" messageEncoding="Text" textEncoding="utf-8"
useDefaultWebProxy="true">
	<readerQuotas maxDepth="32" maxStringContentLength="65536000" maxArrayLength="16384"
	maxBytesPerRead="4096" maxNameTableCharCount="16384" />
	<security mode="Transport">
		<transport clientCredentialType="Certificate" proxyCredentialType="None"
		realm="" />
		<message clientCredentialType="UserName" />
	</security>
</binding>
```

This has a number of changes to the default values, security mode has been changed to transport, this tells the application a certificate is going to be sent along with the request for authentication, various message size values have been increased to handle larger responses from Land Registry. Now we need to add a behaviour to the Service Model to tell the application which certificate to use.

Add the following snippet to your app.config.
Add in the correct serial number (this will be something like "47 cc b3 01")

```
<behaviors>
```

```
        <endpointBehaviors>
                <behavior name="prodCert">
                        <clientCredentials>
                                <clientCertificate findValue="serialNumberHere"
                                x509FindType="FindBySerialNumber" storeLocation="CurrentUser"
                                storeName="My"/>
                                <serviceCertificate>
                                        <authentication certificateValidationMode="PeerTrust"/>
                                </serviceCertificate>
                        </clientCredentials>
                </behavior>
        </endpointBehaviors>
</behaviors>
```

Now we need to tell the endpoint to use the new behaviour.

Within the app.config file, locate the endpoint that has already been added and add in this attribute:
behaviorConfiguration="prodCert"

Change the bindingConfiguration to be the newly created CertPortBinding
Update the URL to correct the target server:
Replace: https://z0.b2b.http.server.landregistry.gov:13007/
With: https://bgtest.landregistry.gov.uk/b2b/

That's us done in the app.config. You should have something like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
        <system.net>
                <defaultProxy useDefaultCredentials="true"/>
        </system.net>
    <system.serviceModel>
        <bindings>
            <basicHttpBinding>
                                <binding name="CertPortBinding" closeTimeout="00:01:00"
openTimeout="00:01:00"
                    receiveTimeout="00:10:00" sendTimeout="00:01:00"
bypassProxyOnLocal="false"
                    hostNameComparisonMode="StrongWildcard" maxBufferSize="65536000"
                    maxBufferPoolSize="524288" maxReceivedMessageSize="65536000"
                    messageEncoding="Text" textEncoding="utf-8" useDefaultWebProxy="true">
                                        <readerQuotas maxDepth="32"
maxStringContentLength="65536000"
                                    maxArrayLength="16384" maxBytesPerRead="4096"
maxNameTableCharCount="16384" />
                                        <security mode="Transport">
                                                <transport clientCredentialType="Certificate"
proxyCredentialType="None"
                                 realm="" />
                                                <message clientCredentialType="UserName" />
                                        </security>
                                </binding>
            </basicHttpBinding>
        </bindings>
        <client>
            <endpoint
address="https://bgtest.landregistry.gov.uk/b2b/BGStubService/DaylistEnquiryV2_0WebService"
                behaviorConfiguration="prodCert" binding="basicHttpBinding"
bindingConfiguration="CertPortBinding"
                contract="DleServRef.DaylistEnquiryV2_0Service"
name="DaylistEnquiryV2_0WSImplPort" />
```

```
        </client>
              <behaviors>
                    <endpointBehaviors>
                          <behavior name="prodCert">
                                <clientCredentials>
                                      <clientCertificate findValue="47 cc b3 01"
x509FindType="FindBySerialNumber" storeLocation="CurrentUser" storeName="My"/>
                                      <serviceCertificate>
                                            <authentication
certificateValidationMode="PeerTrust"/>
                                      </serviceCertificate>
                                </clientCredentials>
                          </behavior>
                    </endpointBehaviors>
              </behaviors>
    </system.serviceModel>
</configuration>
```

Open the source code for your form and include the Service Reference created earlier in the import statements at the top.
using ApplicationName.ServRefName;

Go to the OnClick method we created earlier and add the following code. This creates an instance of the client and request object, it then populates the request with some values and submits it against the service.

```
// create an instance of the client
DaylistEnquiryV2_0ServiceClient client = new DaylistEnquiryV2_0ServiceClient();
// create a request object
RequestDaylistEnquiryV2_0Type request = new RequestDaylistEnquiryV2_0Type();
// populate it
request.ID = new Q1IdentifierType();
request.ID.MessageID = new Q1TextType();
request.ID.MessageID.Value = "testsuccessmanyresults";
request.Product = new Q1ProductType();
request.Product.ContinueIfTitleIsClosedAndContinuedIndicator = new IndicatorType();
request.Product.ContinueIfTitleIsClosedAndContinuedIndicator.Value = true;
request.Product.ExternalReference = new Q1ExternalReferenceType();
request.Product.ExternalReference.Reference = "DotNetQuickStartReference";
request.Product.SubjectProperty = new Q1SubjectPropertyType();
request.Product.SubjectProperty.TitleNumber = new Q2TextType();
request.Product.SubjectProperty.TitleNumber.Value = "DN100";
// submit the request
ResponseDaylistEnquiryV2_0Type response = client.daylistEnquiry(request);
throw new Exception("Application Processed");
```

If you try to run this now and submit the request you should get an error stating *SOAP Header does not have wsse:Security element.* Requests submitted via Business Gateway must contain the login credentials of the user making the submission within the Soap Header.

See Appendix 1 for the C# code to create a header in the necessary format (Appendix 2 contains the code converted to VB).
To use this code, add it to a class in your project, then add an instance of it to your client by adding the following code after you've created the client (for production, you'll need to insert the correct username+password combination):

```
client.ChannelFactory.Endpoint.Behaviors.Add(new
HMLRBGMessageEndpointBehavior("BGUser001","landreg001"));
```

If you now run this, you should get the Exception 'Application Processed'. This means you have successfully submitted a request to the Business Gateway test environment, using the certificate and user credentials to authenticate. When you add all of services you're going to be using, add all the service references you are going to use up front, then each endpoint can use the same binding and behaviour. An example app.config with multiple endpoints is below:

```
<?xml version="1.0"?>
<configuration>
	<system.net>
		<defaultProxy useDefaultCredentials="true"/>
	</system.net>
	<system.serviceModel>
		<bindings>
			<basicHttpBinding>
				<binding name="CertPortBinding" closeTimeout="00:01:00"
openTimeout="00:01:00"
receiveTimeout="00:10:00" sendTimeout="00:01:00"
bypassProxyOnLocal="false"
hostNameComparisonMode="StrongWildcard" maxBufferSize="65536000"
maxBufferPoolSize="524288" maxReceivedMessageSize="65536000"
messageEncoding="Text" textEncoding="utf-8" useDefaultWebProxy="true">
<readerQuotas maxDepth="32"
maxStringContentLength="65536000"
maxArrayLength="16384" maxBytesPerRead="4096"
maxNameTableCharCount="16384" />
<security mode="Transport">
<transport clientCredentialType="Certificate"
proxyCredentialType="None"
realm="" />
<message clientCredentialType="UserName" />
</security>
</binding>
</basicHttpBinding>
		</bindings>
		<client>
			<endpoint
address="https://bgtest.landregistry.gov.uk/b2b/ECDRS_StubService/CorrespondenceV1_0PollRequ
estWebService"
behaviorConfiguration="prodCert" binding="basicHttpBinding"
bindingConfiguration="CertPortBinding"
contract="CorresStub.CorrespondenceV1_0PollRequestService"
name="LiveStubCorres" />
			<endpoint
address="https://bgtest.landregistry.gov.uk/b2b/BGStubService/OutstandingRequestsV2_0WebServ
ice"
behaviorConfiguration="prodCert" binding="basicHttpBinding"
bindingConfiguration="CertPortBinding"
contract="OutstandingRequestsStub.OutstandingRequestsV2_0Service"
name="LiveStubOR" />
			<endpoint
address="https://bgtest.landregistry.gov.uk/b2b/ECDRS_StubService/EDocumentRegistrationV1_0P
ollRequestWebService"
behaviorConfiguration="prodCert" binding="basicHttpBinding"
bindingConfiguration="CertPortBinding"
contract="DrsPollStub.EDocumentRegistrationV1_0PollRequestService"
name="LiveStubDrsPoll" />
			<endpoint
address="https://bgtest.landregistry.gov.uk/b2b/ECDRS_StubService/EDocumentRegistrationV1_0W
ebService"
```

```
                behaviorConfiguration="prodCert" binding="basicHttpBinding"
bindingConfiguration="CertPortBinding"
                contract="DrsStub.EDocumentRegistrationV1_0Service" name="LiveStubDrs" />
            </client>
            <behaviors>
                  <endpointBehaviors>
                        <behavior name="prodCert">
                              <clientCredentials>
                                    <clientCertificate findValue="47 cc b3 01"
x509FindType="FindBySerialNumber" storeLocation="CurrentUser" storeName="My"/>
                                    <serviceCertificate>
                                          <authentication
certificateValidationMode="PeerTrust"/>
                                    </serviceCertificate>
                              </clientCredentials>
                        </behavior>
                  </endpointBehaviors>
            </behaviors>
      </system.serviceModel>
</configuration>
```

Appendix 1 – C# Header code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Xml;
using System.IO;
using System.Xml.Serialization;
using System.Net;
using System.Security.Cryptography.X509Certificates;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;
using System.ServiceModel.Channels;
using System.ServiceModel;
using System.Text;



public class HMLRBGMessageInspector : IClientMessageInspector
{
    #region Member Variables

    private string m_Username = "";
    private string m_Password = "";

    #endregion

    #region Constructor

    public HMLRBGMessageInspector(string username, string password)
    {
        m_Username = username;
        m_Password = password;
    }

    #endregion

    #region IClientMessageInspector Methods

    public void AfterReceiveReply(ref Message reply, object correlationState)
    {

    }

    public object BeforeSendRequest(ref Message request, IClientChannel channel)
    {
        // Create the WsseSecurityHeader with the current user credetials.
        MessageHeader wsseHeader = CreateWsseSecurityHeader();
        request.Headers.Add(wsseHeader);

        // Create the i18nHeader with the locale settings.
        MessageHeader i18nHeader = CreateI18nHeader();
        request.Headers.Add(i18nHeader);

        request.Headers.Action = null;


        return null;
    }
```

```
    #endregion

    #region Private Methods

    private MessageHeader CreateWsseSecurityHeader()
    {
        XmlDocument doc = new XmlDocument();

        StringBuilder xml = new StringBuilder();

        xml.Append("<UsernameToken xmlns=\"http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd\">");
        xml.Append("<Username>");
        xml.Append(m_Username);
        xml.Append("</Username>");
        xml.Append("<Password>");
        xml.Append(m_Password);
        xml.Append("</Password>");
        xml.Append("</UsernameToken>");

        doc.LoadXml(xml.ToString());

        return MessageHeader.CreateHeader("Security", "http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd",
doc.DocumentElement);
    }

    private MessageHeader CreateI18nHeader()
    {
        XmlDocument doc = new XmlDocument();
        doc.LoadXml("<locale xmlns=\"http://www.w3.org/2005/09/ws-
i18n\">en</locale>");

        return MessageHeader.CreateHeader("international",
"http://www.w3.org/2005/09/ws-i18n", doc.DocumentElement);
    }

    #endregion
}

public class HMLRBGMessageEndpointBehavior : Attribute, IEndpointBehavior,
IOperationBehavior, IContractBehavior
{
    #region Member Variables

    private string m_Username = "";
    private string m_Password = "";

    #endregion

    #region Constructor

    public HMLRBGMessageEndpointBehavior(string username, string password)
    {
        m_Username = username;
        m_Password = password;
    }

    #endregion

    #region IEndpointBehavior Methods
```

```
    void IEndpointBehavior.AddBindingParameters(ServiceEndpoint endpoint,
BindingParameterCollection bindingParameters)
    {

    }

    void IEndpointBehavior.ApplyClientBehavior(ServiceEndpoint endpoint,
System.ServiceModel.Dispatcher.ClientRuntime clientRuntime)
    {
        clientRuntime.MessageInspectors.Add(new HMLRBGMessageInspector(m_Username,
m_Password));
    }

    void IEndpointBehavior.ApplyDispatchBehavior(ServiceEndpoint endpoint,
System.ServiceModel.Dispatcher.EndpointDispatcher endpointDispatcher)
    {

    }

    void IEndpointBehavior.Validate(ServiceEndpoint endpoint)
    {

    }

    #endregion

    #region IOperationBehavior Members

    void IOperationBehavior.AddBindingParameters(OperationDescription
operationDescription, BindingParameterCollection bindingParameters)
    {

    }

    void IOperationBehavior.ApplyClientBehavior(OperationDescription
operationDescription, ClientOperation clientOperation)
    {

        clientOperation.Parent.MessageInspectors.Add(new
HMLRBGMessageInspector(m_Username, m_Password));

    }

    void IOperationBehavior.ApplyDispatchBehavior(OperationDescription
operationDescription, DispatchOperation dispatchOperation)
    {

        //throw new NotImplementedException();

    }

    void IOperationBehavior.Validate(OperationDescription operationDescription)
    {

        // throw new NotImplementedException();

    }

    #endregion

    #region IContractBehavior Members
```

```
    void IContractBehavior.AddBindingParameters(ContractDescription
contractDescription, ServiceEndpoint endpoint, BindingParameterCollection
bindingParameters)
    {

    }

    void IContractBehavior.ApplyClientBehavior(ContractDescription
contractDescription, ServiceEndpoint endpoint, ClientRuntime clientRuntime)
    {
        clientRuntime.MessageInspectors.Add(new HMLRBGMessageInspector(m_Username,
m_Password));
    }

    void IContractBehavior.ApplyDispatchBehavior(ContractDescription
contractDescription, ServiceEndpoint endpoint, DispatchRuntime dispatchRuntime)
    {

        //throw new NotImplementedException();

    }

    void IContractBehavior.Validate(ContractDescription contractDescription,
ServiceEndpoint endpoint)
    {

        //throw new NotImplementedException();

    }

    #endregion

}

public class AcceptAllCertificatePolicy : ICertificatePolicy
{
    public AcceptAllCertificatePolicy()
    { }

    public bool CheckValidationResult(ServicePoint sPoint, X509Certificate cert,
WebRequest wRequest, int certProb)
    {            // *** Always accept
        return true;
    }
}
```

Appendix 2 – VB Header code

```
Imports System.ServiceModel
Imports System.ServiceModel.Web
Imports System.Text
Imports System.Collections
Imports System.ServiceModel.Channels
Imports System.ServiceModel.Description
Imports System.ServiceModel.Dispatcher
Imports System.Xml
Imports System.Net
Imports System.Security.Cryptography.X509Certificates
Imports System.IO

Public Class HMLRBGMessageInspector
    Implements IClientMessageInspector


    Public Sub AfterReceiveReply(ByRef reply As System.ServiceModel.Channels.Message, ByVal
correlationState As Object) Implements
System.ServiceModel.Dispatcher.IClientMessageInspector.AfterReceiveReply

    End Sub

    Public Function BeforeSendRequest1(ByRef request As
System.ServiceModel.Channels.Message, ByVal channel As System.ServiceModel.IClientChannel)
As Object Implements
System.ServiceModel.Dispatcher.IClientMessageInspector.BeforeSendRequest
        ' Create the WsseSecurityHeader with the current user credetials.
        Dim wsseHeader As MessageHeader = CreateWsseSecurityHeader()
        request.Headers.Add(wsseHeader)

        ' Create the i18nHeader with the locale settings.
        Dim i18nHeader As MessageHeader = CreateI18nHeader()
        request.Headers.Add(i18nHeader)

        request.Headers.Action = Nothing

        Return Nothing
    End Function
    Private m_Username As String = ""
    Private m_Password As String = ""


    Public Sub New(ByVal username As String, ByVal password As String)
        m_Username = username
        m_Password = password
    End Sub


    Private Function CreateWsseSecurityHeader() As MessageHeader
        Dim doc As New XmlDocument()

        Dim xml As New StringBuilder()

        xml.Append("<UsernameToken xmlns=""http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd"">")
        xml.Append("<Username>")
        xml.Append(m_Username)
        xml.Append("</Username>")
        xml.Append("<Password>")
```

```
        xml.Append(m_Password)
        xml.Append("</Password>")
        xml.Append("</UsernameToken>")

        doc.LoadXml(xml.ToString())

        Return MessageHeader.CreateHeader("Security", "http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd", doc.DocumentElement)
    End Function

    Private Function CreateI18nHeader() As MessageHeader
        Dim doc As New XmlDocument()
        doc.LoadXml("<locale xmlns=""http://www.w3.org/2005/09/ws-i18n"">en</locale>")

        Return MessageHeader.CreateHeader("international", "http://www.w3.org/2005/09/ws-
i18n", doc.DocumentElement)
    End Function
End Class

Public Class HMLRBGMessageEndpointBehavior
    Inherits Attribute
    Implements IEndpointBehavior
    Implements IOperationBehavior
    Implements IContractBehavior
#Region "Member Variables"

    Private m_Username As String = ""
    Private m_Password As String = ""

#End Region

#Region "Constructor"

    Public Sub New(ByVal username As String, ByVal password As String)
        m_Username = username
        m_Password = password
    End Sub

#End Region

#Region "IEndpointBehavior Methods"

    Private Sub IEndpointBehavior_AddBindingParameters(ByVal endpoint As ServiceEndpoint,
ByVal bindingParameters As BindingParameterCollection) Implements
IEndpointBehavior.AddBindingParameters

    End Sub

    Private Sub IEndpointBehavior_ApplyClientBehavior(ByVal endpoint As ServiceEndpoint,
ByVal clientRuntime As System.ServiceModel.Dispatcher.ClientRuntime) Implements
IEndpointBehavior.ApplyClientBehavior
        clientRuntime.MessageInspectors.Add(New HMLRBGMessageInspector(m_Username,
m_Password))
    End Sub

    Private Sub IEndpointBehavior_ApplyDispatchBehavior(ByVal endpoint As ServiceEndpoint,
ByVal endpointDispatcher As System.ServiceModel.Dispatcher.EndpointDispatcher) Implements
IEndpointBehavior.ApplyDispatchBehavior

    End Sub

    Private Sub IEndpointBehavior_Validate(ByVal endpoint As ServiceEndpoint) Implements
IEndpointBehavior.Validate
```

```
    End Sub

#End Region

#Region "IOperationBehavior Members"

    Private Sub IOperationBehavior_AddBindingParameters(ByVal operationDescription As
OperationDescription, ByVal bindingParameters As BindingParameterCollection) Implements
IOperationBehavior.AddBindingParameters

    End Sub

    Private Sub IOperationBehavior_ApplyClientBehavior(ByVal operationDescription As
OperationDescription, ByVal clientOperation As ClientOperation) Implements
IOperationBehavior.ApplyClientBehavior

        clientOperation.Parent.MessageInspectors.Add(New HMLRBGMessageInspector(m_Username,
m_Password))

    End Sub

    Private Sub IOperationBehavior_ApplyDispatchBehavior(ByVal operationDescription As
OperationDescription, ByVal dispatchOperation As DispatchOperation) Implements
IOperationBehavior.ApplyDispatchBehavior

        'throw new NotImplementedException();

    End Sub

    Private Sub IOperationBehavior_Validate(ByVal operationDescription As
OperationDescription) Implements IOperationBehavior.Validate

        ' throw new NotImplementedException();

    End Sub

#End Region

#Region "IContractBehavior Members"

    Private Sub IContractBehavior_AddBindingParameters(ByVal contractDescription As
ContractDescription, ByVal endpoint As ServiceEndpoint, ByVal bindingParameters As
BindingParameterCollection) Implements IContractBehavior.AddBindingParameters

    End Sub

    Private Sub IContractBehavior_ApplyClientBehavior(ByVal contractDescription As
ContractDescription, ByVal endpoint As ServiceEndpoint, ByVal clientRuntime As
ClientRuntime) Implements IContractBehavior.ApplyClientBehavior
        clientRuntime.MessageInspectors.Add(New HMLRBGMessageInspector(m_Username,
m_Password))
    End Sub

    Private Sub IContractBehavior_ApplyDispatchBehavior(ByVal contractDescription As
ContractDescription, ByVal endpoint As ServiceEndpoint, ByVal dispatchRuntime As
DispatchRuntime) Implements IContractBehavior.ApplyDispatchBehavior

        'throw new NotImplementedException();

    End Sub
```

```
    Private Sub IContractBehavior_Validate(ByVal contractDescription As ContractDescription,
ByVal endpoint As ServiceEndpoint) Implements IContractBehavior.Validate

        'throw new NotImplementedException();

    End Sub

#End Region

End Class

Public Class AcceptAllCertificatePolicy
    Implements ICertificatePolicy
    Public Sub New()
    End Sub

    Public Function CheckValidationResult(ByVal sPoint As ServicePoint, ByVal cert As
X509Certificate, ByVal wRequest As WebRequest, ByVal certProb As Integer) As Boolean
Implements ICertificatePolicy.CheckValidationResult
        ' *** Always accept
        Return True
    End Function
End Class
```

Revision History

Date	Version	Description	Author
16/04/2013	1.0	Version to be published	Matthew Bell