Copyright

© Copyright 2012 SAP AG. All rights reserved

SAP Library document classification: PUBLIC

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. Microsoft, Windows, Excel, Outlook, PowerPoint, Silverlight, and Visual Studio are registered trademarks of Microsoft Corporation.
IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, z10, z/VM, z/OS, OS/390, zEnterprise, PowerVM, Power Architecture, Power Systems, POWER7+, POWER6+, POWER6, POWER, PowerHA, pureScale, PowerPC, BladeCenter, System Storage, Storwize, XIV, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, AIX, Intelligent Miner, WebSphere, Tivoli, Informix, and Smarter Planet are trademarks or registered trademarks of IBM Corporation.
Linux is the registered trademark of Linus Torvalds in the United States and other countries.
Adobe, the Adobe logo, Acrobat, PostScript, and Reader are trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.
Oracle and Java are registered trademarks of Oracle and its affiliates.
UNIX, X/Open, OS/1, and Motif are registered trademarks of the Open Group.
Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems Inc.
HTML, XML, XHTML, and W3C are trademarks or registered trademarks of W3C®. World Wide Web Consortium, Massachusetts Institute of Technology.
Apple, App Store, iBooks, iPad, iPhone, iPod, iPhoto, iPad, iTouch, Objective-C, Retina, Safari, Siri, and Xcode are trademarks or registered trademarks of Apple Inc.
iOS is a registered trademark of Cisco Systems Inc.
RIM, BlackBerry, BBM, BlackBerry Curve, BlackBerry Bold, BlackBerry Pearl, BlackBerry Torch, BlackBerry Storm, BlackBerry Storm2, BlackBerry PlayBook, and BlackBerry App World are trademarks or registered trademarks of Research in Motion Limited.
Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.
INTERMEC is a registered trademark of Intermec Technologies Corporation.
Wi-Fi is a registered trademark of Wi-Fi Alliance.
Bluetooth is a registered trademark of Bluetooth SIG Inc.
Motorola is a registered trademark of Motorola Trademark Holdings LLC.
Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.
SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.
Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.
Sybase and Adaptive Server, Anywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc.
Sybase is an SAP company.
Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.
All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.
These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
1 Introduction

The SAP NetWeaver® Gateway developer tools simplify service consumption and increase productivity using the development environment of choice.

The tool created with Microsoft Visual Studio® enables you to easily and quickly create ASP.NET starter kit applications which can retrieve data from your existing SAP systems. You can then take the generated code and extend it to support your needs.

It is based on a wizard in Microsoft Visual Studio, which allows you to simply choose an SAP NetWeaver Gateway 2.0 service, define UI screens, and create an ASP.NET application.

The following are the main features of the SAP NetWeaver Gateway developer tool for Visual Studio:

- Allows you to browse through the SAP NetWeaver Gateway services and view the service properties, details, and associations.
- Leverages the Open Data Protocol (OData) SDK to create proxy classes for .Net projects for connection to SAP NetWeaver Gateway services.
- Supports the creation of starter kit applications which include UI screens.

The generated application contains generated UI which only supports read-only scenarios. However, using the generated proxy classes in your code, you will be able to execute read/write calls.

1.1 Target Audience

The SAP NetWeaver Gateway developer tools are targeted for developers. This document describes the use of the tool for Microsoft Visual Studio 2010.

1.2 Glossary

**Entity set/collection** - An entity set is a set of entities of the same type (for example, all persons having an account at a bank). In OData, a Collection is represented as an Atom Feed.

**Association/link** - In object-oriented programming, association defines a relationship between classes of objects which allows one object instance to cause another to perform an action on its behalf. This relationship is structural, because it specifies that objects of one kind are connected to objects of another.

**Navigation Property** - A property of an Entry that represents a Link from the Entry to one or more related Entries. A Navigation Property is not a structural part of the Entry to which it belongs.

**Entry** - Members of an entity set/collection. In OData, entries are represented as Atom Entry Documents. An Entry may represent an instance of an Open Entity Type if, in addition to Declared Properties, the entry includes Dynamic Properties.

**OData Service** - A service which implements the Open Data Protocol (OData).

**Service Metadata Document (Metadata Document)**: A conceptual schema definition language (CSDL) that describes the data model (i.e. structure and organization of all the resources) exposed as HTTP endpoints by an OData service.
2 Installing the Tool

2.1 Prerequisites

The tool must be installed by developers who have administrator rights on their operating system.

The following are the requirements necessary for installing the tool:

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP NetWeaver Gateway</td>
<td>2.0</td>
</tr>
<tr>
<td>.NET</td>
<td>.NET Framework 4.0</td>
</tr>
<tr>
<td>Microsoft Visual Studio</td>
<td>Visual Studio 2010</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows XP, Windows Vista, and Windows 7, and Server 64</td>
</tr>
<tr>
<td>Runtime environment</td>
<td>.NET 4.0</td>
</tr>
</tbody>
</table>

The developer must have understanding of the following subjects:

- OData
- ASP.NET

2.2 Installation Procedure

1. Download the SAP developer tool.exe file from the SCN to your machine.
2. Open the file.
   A User Account Control message is displayed.
3. Click Yes.
   The SAP NetWeaver Gateway developer tool wizard is displayed.
4. Select the I accept the terms in the License Agreement checkbox.
5. Click Install.
6. When the installation is complete, click Finish.
   SAP NetWeaver Gateway developer tool for Visual Studio 2010 is installed and ready to use.
2.3 Uninstalling the Tool

You can uninstall the tool from the Add/Remove Programs application.

**Note:** When you uninstall the tool, the download metadata files are not deleted automatically. These files are created per user. Each user (or administrator) must delete them manually.

**To delete the metadata files manually in Windows 7 and Vista:**
1. Go to C:\Users\<User's Folder>\AppData\Local\SAP
2. Delete the SAP NetWeaver Gateway developer tool folder.

**To delete the metadata files manually in Windows XP:**
1. Go to C:\Documents and Settings\<User's Folder>\Local Settings\Application Data\SAP
2. Delete the SAP NetWeaver Gateway developer tool folder.

2.4 Upgrading the Tool

We recommend that you work on the most updated version of the tool. A check for updates is performed automatically in the background once a month. You will be prompted to update if there is a newer version available. You can also check for updates manually.

**To check for updates:**
1. Open the tool.
   - The Create SAP Application window is displayed.
2. Click Browse.
   - The SAP NetWeaver Gateway Server Details window is displayed.
3. Click Check for Updates.
   - A popup is displayed indicating whether there are updates available. If there are, you are redirected to the download site.

**To upgrade the tool:**
1. Download the **SAP developer tool.exe** file from the [SCN](http://scn.sap.com) to your machine.
2. Open the file.
   - A User Account Control message is displayed.
3. Click Yes.
   - The SAP NetWeaver Gateway developer tool wizard is displayed.
4. Select the **I accept the terms in the License Agreement** checkbox.
5. Click Install.
6. When the installation is complete, click Finish.

   The newest version of the SAP NetWeaver Gateway developer tool for Visual Studio 2010 is installed and ready to use. The SAP NetWeaver Gateway connection details from the older version are saved to the updated version.
3 Using the Tool

The tool creates a starter kit application which enables you to take the code and expand or change it in whichever way you like. The tool also allows you to create proxy classes for an existing project.

3.1 Creating a New Starter Kit Application

The application is created by selecting a service and defining pages.

For each page you can select between two possible types of pages:

- **List** – the UI generated from this type of page is a table displaying information obtained from an entity set. For the chosen entity set, you are required to choose the fields you would like to display which will then become the table column headers. For example, the Customer entity set will have fields such as Customer Name, Country, Street, and Telephone Number.

- **Details** – the UI generated from this type of page is a two-columned table displaying information obtained from an entity set. For the chosen entity set, you are required to choose the fields you would like to display which will then become the table row headers. The first column displays the entity set field and the second column displays the field value.

The relationship between the pages is based on the association between the entity sets. After you have selected an entity set for the first page (for example, Customers), in the second page you can only select an entity set that is associated to it (for example, Banks) or you can choose to display the details of the entity set you selected for the first page (for example, Customers).

**To create a starter kit application:**
2. Click **New Project**.
3. Double-click **SAP Web Application**.
   
   The **Create SAP Application** window is displayed.
4. Click **Browse** to select the service you want to use for generating your application.
   
   The **SAP NetWeaver Gateway Server Details** window is displayed.
5. Enter the details of the SAP NetWeaver Gateway server to which you want to connect.
6. If necessary, select the **Use SSL** checkbox. (Recommended)
   
   **Note:** You must first make sure the SAP administrator has already configured the SAP NetWeaver Gateway to work with SSL.
7. In the **SAP Client** field, enter your client number. If you do not have an SAP client number, select the **Use Default** checkbox.
8. Enter your user name and password.
9. Click **OK**.
The **Select SAP Service Reference** window is displayed containing the following:

- The name of the SAP NetWeaver Gateway server selected.
- List of services available in the SAP NetWeaver Gateway.

**Note**: If there are services that were not loaded (due to problems with the metadata), the ![information icon](image) icon is displayed. In the Event Viewer, a list containing the names of the failed services is displayed.

- The path to the URL of the selected service.
- List of entity sets available in each service.
- A table containing the properties, associations and description for each entity set.

The first time the tool is run, the information on the services available in the SAP NetWeaver Gateway is downloaded to your computer. This enables you to explore proxy classes and starter kit applications even when working offline.

Later, you can click **Refresh** to connect to the SAP NetWeaver Gateway and download any new metadata or updates on metadata.

**Note**: Downloading metadata might be time consuming, but once you see a service in the list, you can begin working with it. There is no need to wait for all services to be downloaded. Services that were not downloaded before **Select** or **Cancel** was clicked will not appear on the list until the list is refreshed.
Note: The CustomerManagementModel is an out-of-the-box example of a model that can be used to explore the various possibilities within SAP data model without the need to connect to a live SAP NetWeaver Gateway system. This model is presented in the form of a file located at the following location:

- For Windows 7 and Vista: C:\ProgramData\SAP\SAP NetWeaver Gateway developer tool\MetadataFiles
- For Windows XP: C:\Documents and Settings\All Users\Application Data\SAP\SAP NetWeaver Gateway developer tool\MetadataFiles

(You can also add your own files to this folder and they will appear in the Browse table above.)

The CustomerManagementModel will not run during runtime since it is a model and not a live service.

10. In the Search field, you can enter the name (full or partial) of a desired service or entity set.
   All services containing the relevant string or containing entity sets that have it, are displayed.

11. From the Service Name section, select the desired service.
    The entity sets contained in the selected service are displayed in the lower-left section of the window.

12. Select an entity set.
    The entity set’s properties are displayed on the table on the lower-right section of the window.

13. Display the Associations tab to view the entity set’s associations. (Associations define the relationship between entity types, for example 1 to 1 or 1 to many.)

14. Display the Entity Set Description tab to view the entity set’s details.

15. Click Select.
    The Create SAP Application window is displayed.
16. Configure the first page of the application as it will be displayed in the UI.

   a. In the Title field, enter a name for the page.

   b. From the Page Type drop-down list, the List option is selected automatically (since this is the only option available for the first page).

   c. For a List page type, select the desired entity set to be displayed from the Entity Set drop-down list.

      Note: If you select an entity set which requires additional URL parameters, after the code is generated, open the Page0.aspx.cs file and search for the following comment:

      Changes to additional query options are made here:
      Add the required filter parameters in the string below the comment.

   d. Click Test in Browser to display in a browser the service response for the entity set you chose.

   e. Click Add to see the fields available for the selected entity set.

      The Add Entity Fields window is displayed.
f. Select the checkboxes of the desired fields for the page.

g. Click OK.

You are returned to the Create SAP Application window.

h. If needed, select a field and click Remove to delete it.

i. Use the Move Up and Move Down arrows to change the positioning of the fields in the page.

17. In the Create SAP Application window, under the Application Pages section, click New to add another page.

   Note: There is a relationship between the pages. The entity sets available for each page are dependent upon the entity set selected in the previous page.

18. Repeat step 16 for configuring the new pages.

   Notes:
   - For pages other than the first, the List option may be disabled depending on the associations defined on the selected service. You can select between List or Details for the page type.
   - For a Details page type, this is done in the Add Entity Fields window from the Entity Set Navigation drop-down list there.
   - Only the entity sets relevant to that selected in the previous page are displayed.

19. Click Delete to delete a page.

   Note: All pages positioned after the one you delete, will be removed too because of the relationship between the pages based on the associations.

20. If necessary, you can click Browse to return to the Select SAP Service Reference window and select a different service for the application.

   Note: If you change the service (or select the same service again), all the pages you created before will be deleted.

   Note: The tool does not support update of the proxy when the service changes.
21. If necessary, you can return to the Select SAP Service Reference window and click Configure… to change the SAP NetWeaver Gateway details. The SAP NetWeaver Gateway Server Details page is displayed.

22. When you have created all the desired pages, click OK. The starter kit application is created.

3.2 Creating Proxy Classes for an Existing Application

2. Click Open Project.
3. Select the desired existing project.
4. In the Solution Explorer, right-click the project and select Add SAP Service Reference.

The Select SAP Service Reference page is displayed.
5. Select the checkbox(es) of the desired service(s).

6. Click OK.

The **SAP Service References** folder is displayed in the Solution Explorer containing a service proxy and metadata proxy classes for each service you selected.

**Note:** The tool currently does not support update of the proxy when the service changes. Optionally you can delete the files and create an updated proxy.
3.3 Offline Metadata File Support

You can create proxies or starter kit applications also when you are not connected to the SAP NetWeaver Gateway. All you need is the md file of the service. The tool provides a mockup service out of the box called CustomerManagementModel which you can use to create a starter kit application.

For the application to work, you must add a predefined runtime URL for the service metadata file before you create the application.

If you wish to use your own metadata file, you can first add a predefined runtime URL and then copy the file to the location described below.

To add a predefined runtime URL for the service:

1. Open the XML file in edit mode.
2. Add the following attributes to the <Edmx> element:
   ```xml
   xmlns:sap="http://www.sap.com/Protocols/SapData" sap:url="[Your URL]"
   
   For example:
   
   Before:
   
   
   After:
   
   xmlns:sap="http://www.sap.com/Protocols/SapData" sap:url="[Your URL]">
   ```
To save the metadata file:
1. In the file system, search for the Metadatafiles folder at the following location:
   - For Windows 7 and Vista: C:\ProgramData\SAP\SAP NetWeaver Gateway developer tool\MetadataFiles
   - For Windows XP: C:\Documents and Settings\All Users\Application Data\SAP\SAP NetWeaver Gateway developer tool\MetadataFiles

   **Note:** A metadata file for a mockup service is supplied with the tool. This file (ManageCustomerMetadata.xml) will already be found at this location.
2. Copy your metadata file to the Metadatafiles folder.
3. Open the tool.
   The service metadata file you saved appears in the list of services.
4. Search for your service.
5. Use this to create a starter kit application or proxy as described in the Creating a New Starter Kit Application procedure above.

### 3.4 Security Requirements
The tool supports basic authentication with the option of SSL. The password is saved in memory and is deleted when the user exits the tool.
4 Extending the Generated Application

4.1 Generated Application Structure

The generated application is composed of the following sections:

User authentication – Contains the code used to authenticate the web application user with the SAP NetWeaver Gateway.

Pages – Contains the code of the pages generated with the tool.

Generated proxies – Contains 3 files: one for the generated OData proxies, one for the SAP-extended proxies (used to support the CRUD options), and one for the metadata dictionary (used to represent the SAP Data Protocol attributes of the selected service metadata (for example sap:label, sap:filterable etc.).)
4.2 Extensibility options for the Generated Application

Once you have used the tool to create a starter kit application, you can use the generated code to extend the application and adapt it to your project.

**To add or edit parameters:**

1. From the Solution Explorer, open the code of the first page of the application.
2. Search for the line with the instruction "Changes to additional query options are made here:"
3. In the line below you can add or edit parameters.

![Code snippet]

**To change the User Interface:**

1. From the Solution Explorer, select any ASPX page you want to edit.
2. Search for the line with the instruction "Below is a list of fields to be displayed in the Data Grid. You may edit fields here"
3. In the lines below you can do the following:
   - Add fields, by adding a new line with the field data (the DataField must already exist in the system).
   - Remove fields, by deleting the relevant line.
   - Change labels, by editing the HeaderText in the relevant line.
   - Change positioning, by changing the positioning of the lines.
To change the URL used to access the SAP NetWeaver Gateway:
1. Open the Web.config file.
2. Search for the <appSettings> section.
3. In the line starting <add key="<service metadata name>_ServiceUrl", change the value parameter to the desired URL.

To define the Log Source:
1. Open the Web.config file.
2. Search for the <appSettings> section.
3. In the line starting <add key="Log Source", change the value parameter to the desired name for the log source.
4. Navigate to the directory in which the SAP NetWeaver Gateway developer tool for Visual Studio is installed.
5. Copy the CreateEventSource.exe file to the machine where the generated SAP Web Application is deployed.
6. In the machine where the application is deployed, run the CreateEventSource.exe file in the command prompt using administrator rights and enter the name you set above as the only parameter.

To set the minimum log level:
1. Open the Web.config file.
2. Search for the <appSettings> section.
3. In the line starting <add key="MinimalLogLevel", change the value parameter to the desired minimum log level.
   - To view log Information messages and higher, enter the following line:
     <add key="LogLevel" value="Info"/>
   - To view log Warning messages and higher, enter the following line:
     <add key="LogLevel" value="Warning"/>
   - To view log Error messages, enter the following line:
     <add key="LogLevel" value="Error"/>
5 Publishing to IIS

To publish to IIS:

1. Open the desired application.
2. In the Solution Explorer, select the application.
3. Right-click and select Publish.
   The Publish Web page is displayed.

   ![Publish Web Page]

4. Fill in the required fields.
5. Select the **Mark as IIS application on destination** checkbox.
6. Click Publish.
   The application is available to anyone inside the specified network.
6 Running the Generated Application

The application generated runs in debug mode.

To run the application:
1. From Visual Studio, open the application.
2. Press F5.

6.1 Accessing Logs during Runtime

Only a user with administrator rights on the computer where the generated application is deployed can access the logs. The minimal log level displayed is defined in the web.config file as described in the To set the minimum log level section above.

To access the logs:
1. Open the Event Viewer.
2. From the Event Viewer section (on the left side of the window), click Application.
   All logs are displayed in the right side of the window.
3. Filter the logs according to the Source type as described in the To define the Log Source section above.

7 Security

7.1 Runtime Flow

The generated web application has an out-of-the-box support for basic authentication.

The authentication flow is as follows.
1. When an anonymous user tries to access one of the application pages, he is redirected to the login page (configured through the web.config file). The login page tries to connect to the URL of the SAP NetWeaver Gateway service with no credentials.
2. If it succeeds, a standard ASP.NET Forms Authentication cookie is created for this guest user and the application is redirected to the desired page. Each page in the application contains a flag indicating whether the user is a guest or an authenticated user. This flag is set when the page is loaded (Forms Authentication). For guests, the application calls the SAP NetWeaver Gateway services without any credentials.
   If it fails, the user is asked to provide credentials on the login page.
3. After submitting the credentials, the application tries to access the service again.
4. If it succeeds:
   - The application gets a logon cookie from the SAP NetWeaver Gateway response (the cookie between SAP NetWeaver Gateway and IIS).
   - The application generates an encryption key and encrypts the SAP NetWeaver Gateway cookie (referred to as SAP NetWeaver Gateway cookie in this text).
   - The application uses the encryption key and creates a standard ASP.NET Forms Authentication cookie with it (the cookie between the browser and the IIS, referred to...
as User cookie in this text). The encrypted SAP NetWeaver Gateway cookie is saved the User cookie along with the information about the SAP NetWeaver Gateway domain. This data will be used by the application on subsequent requests from the user to the SAP NetWeaver Gateway. The encryption key is saved as a session variable between the user and the IIS and is used to decrypt the cookie when needed.

5. For authenticated users, the application tries to get the encryption key from the session and decrypt the cookie and extract from it the SAP NetWeaver Gateway logon cookie and domain information. This data is attached to the service request call and data is retrieved without the need to provide user name or password.

6. If the session between the SAP NetWeaver Gateway and IIS or that between the browser and the IIS expires, the user is redirected to the login page again.

We recommend you create an IIS site with an HTTPS port so that the generated cookies will have a Secure attribute. Only use the SSL connection to the SAP NetWeaver Gateway server for protecting the authentication details. We recommend that you develop a solution based on X.509 client certificates for running the application in a productive environment. The short lived certificates should be generated on the fly by code running on IIS. You can see an example of this solution in the Appendix below.
8 Troubleshooting

8.1 General Guidelines

- When using the proxy to update an entity from the Gateway service, make sure to add the parameter in the `service.SaveChanges` method as shown below:
  ```csharp
  service.SaveChanges(SaveChangesOptions.ReplaceOnUpdate);
  ```
- Gateway supports the sdata and the odata service catalogs. If a service is upgraded and exists both catalogs, only the odata service is displayed in the Select SAP Service Reference page.
  In the logs, you can find a list of the services that have been upgraded to odata.
- SAP Semantics
  The SAP semantics supported by this tool are: email and URL.

8.2 Known Issues

- The generated application’s UI can only display simple type entity properties. Although values of complex type entity properties are not displayed at runtime, they are accessible programmatically through the generated proxy.
- The tool uses the Microsoft OData SDK, which does not allow the use of certain reserved keywords, for example Return. Using these words as service entity names might result in compilation errors in the generated proxy classes.
  You can find the entire list of words at:

8.3 Solving Problems

Problem
When entering the credentials on the generated application, the following message displays: “Your login attempt was not successful. Please try again.”

Solution
Remove the SAP proxy.
1. Go to Tools > Internet Options > Connections > LAN settings.
2. Remove the Automatically detect settings checkbox.
3. Enter the proxy settings manually.

Problem
When a new Gateway connection is installed, the URLs in the metadata are relative URLs and not full URLs. This means that only the CustomerManagementModel service is loaded.

Solution
Implement note 1587789.
Appendix
X.509 Sample Code - Support for Generated Web Applications

Introduction
Typically, HTTPS connection on top of SSL/TLS transport protocol is based on server side authentication only. To ensure the server’s identity, the client side must trust an issuer (CA) of the server certificate. In other words, the CA certificate must be trusted on the client side. On Windows machines, the certificate is located under Trusted Root Certificate Authorities on the client machine. For better security, the server can also validate client identity based on its certificate. This is called mutual authentication.

Mutual authentication requires an X.509 client certificate (along with a private key) to be available on the client side. In turn, the server should trust an issuer (CA) of the certificates. In client-to-server communications, the certificate should be available on the client side and its private key should be stored in a secure manner (that is, the certificate export is protected via password). In server-to-server communications, the X.509 client certificates cannot be permanently available on the consumer side. Instead, they can be generated on the fly by the server code and attached to the outgoing HTTPS request.

.NET Framework does not have embedded support for generating client certificates. Instead, the Bouncy Castle external package can be used. You can find this package at: http://www.bouncycastle.org/csharp/

The sample code assumes that the signing CA certificate:

- Is available under Trusted Root Certificate Authorities (Current User) – could be configurable.
- Is configured as a CA certificate.
- Has appropriate values of Key Usage attributes (Digital Signature, Certificate Signing).

SAP NetWeaver Gateway Implementation

Prerequisites
- Import the CA certificate to the Local Computer store on the machine where the SAP Web Application is deployed.
- Import the CA certificate (public key only) to the SSL Server PSE in STRUST, in the SAP NetWeaver Gateway system that the SAP Web Application is calling.
- The Application Pool Account, which is used by the web application, should have access to the CA certificate private key.

Use WinHttpCertCfg.exe, a Certificate Configuration Tool for that purpose: http://msdn.microsoft.com/en-us/library/Aa384088
Integrating the X509 Solution to the Generated Web Application

To integrate the X509 solution to the Generated Web Application:

1. Include the "X509Helper.cs" class in your generated Web Application.

2. Configure the `const` values in the "X509Helper.cs" class so that they match both the subject of the CA certificate and the subject defined in the SAP NetWeaver Gateway.

3. The sample code uses the current IIS authenticated user as the subject of the client certificate. To test the code, change the main web.config file to use Windows Authentication instead of Forms Authentication.

   ```
   <authentication mode="Windows"/>
   ```

4. Change the web.config file in the Pages folder to allow direct access to the pages. (Does not redirect to the Account/Login.aspx.)

   ```
   <configuration>
   <authorization>
   <deny users="?"><!--authentication mode="Forms">  
   <forms loginUrl="~/Account/Login.aspx" defaultUrl="~/Pages/Page0.aspx" timeout="2880" />  
   </authentication>-->
   </authorization>
   </system.web>
   ```

5. In the class Account/Login.aspx.cs, change the implementation of the method "ProcessPageTicketToSAPService" to:

   ```
   public static void ProcessPageTicketToSAPService(WebRequest request,
   HttpRequest httpRequest, HttpResponse httpResponse,
   byte[] secret, StringBuilder logMessage)
   {
   X509Helper.AttachClientCertToRequest(request,
   HttpContext.Current.User.Identity.Name);
   }
   ```

6. In each generated page, in the method Page_Init(), set the Login\Logout UI control to not visible by calling the following code:

   ```
   Page.Master.FindControl("HeadLoginView").Visible = false;
   ```
Methods in X509Helper.cs

AttachClientCertToRequest
This method receives a web request and a user name as parameters, finds the CA certificate using GetCaCertificate, generates a new client certificate for the user name received using GenerateCertificate, and attaches the new client certificate to the web request.

GetCaCertificate
This method receives a subject name as a parameter and returns an instance of class X509Certificate2 containing both the certificate and its private key. The private key is used for signing the client certificates.

GenerateCertificate
This method receives a reference to the CA certificate and the subject name of the new user certificate. The subject name is of class X509Name. This allows the certificate subject being customized on the customer environment by adding certain fields (e.g. Country, Organization, etc.).

The generated certificate should:
- Be valid a few minutes prior to the current time (as a clock skew time) and up to 24 hours after creation.
- Be signed by the CA private key.
- Not be used as a CA certificate.
- Have a reference to CA certificate via AuthorityKeyIdentifier.

ConvertToPfx
The generated certificate contains only public key. If a certificate should be attached to HTTPS request, a private key should be available as well. The method generates a certificate based on PKCS# 12 format. The generated certificate is comprised from two entries of class X509CertificateEntry: a user certificate and a private key. The private key of the certificate contains a certificate chain that includes also the CA certificate.

Performance Recommendations
1. Generation of asymmetric public/private key pair is extensive performance operation. In order to reduce the overall performance, all generated certificates could use the same key pair. The pair could be cached and renewed according to an expiration value of the generated certificates (i.e. 24 hours).
2. The generated certificates should be cached. The cache size could be defined based on customer needs.