How To Use SharePoint 2010 Secure Store As Single Sign-On Service For SAP Applications Using ERPConnect

Jürgen Bäurle

April 2011

Parago Media GmbH & Co. KG

Introduction

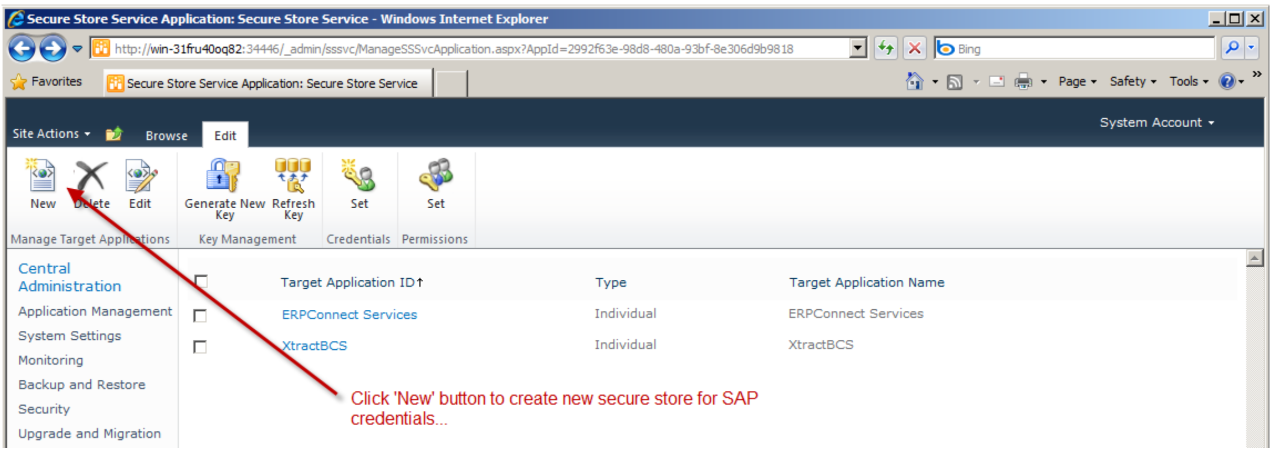
The Secure Store Service in SharePoint 2010 replaces the Single Sign-on Shared Service of MOSS 2007 and provides an easy way to map user credentials of external resources like SAP systems to Windows users. During the process of developing SAP interfaces using the very handy ERPConnect library from Theobald Software you have to open a R3 connection with the SAP system using SAP account credentials (username and password).

In most cases you will use a so called technical user with limited access rights to execute or query objects in SAP, but a SAP system saves a lot of sensitive data which cannot all be shown to all users. So, creating a new secure store in SharePoint 2010 to save the SAP user credentials will be the solution. Accessing the secure store from program code is quite simple.

A trail version of the ERPConnect library can be downloaded at www.theobald-software.com.

Secure Store Configuration

The Secure Store Service will be managed by the Central Administration (CA) of SharePoint 2010 under *Application Management > Manage service applications > Secure Store Service*:

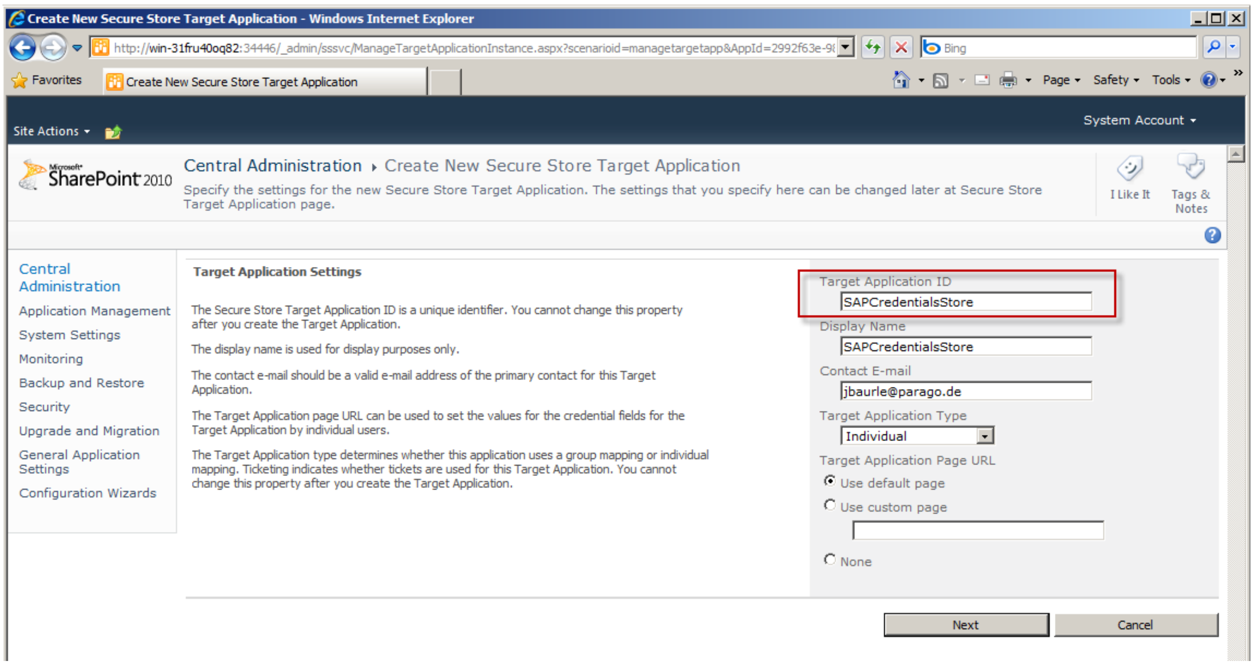

Secure Store Service Application: Secure Store Service - Windows Internet Explorer
http://win-31fru40oq82:34446/_admin/sssvc/ManageSSSvcApplication.aspx?AppId=2992f63e-98d8-480a-93bf-8e306d9b9818
Bing
Favorites
Secure Store Service Application: Secure Store Service
Page
Safety
Tools
Site Actions
Browse
Edit
System Account
New
Delete
Edit
Generate New Key
Refresh Key
Set
Set
Manage Target Applications
Key Management
Credentials
Permissions
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration

Target Application ID	Type	Target Application Name
ERPConnect Services	Individual	ERPConnect Services
XtractBCS	Individual	XtractBCS

Click 'New' button to create new secure store for SAP credentials...

As the screenshot above shows is it possible to create multiple target applications within one Secure Store Service.

Clicking the New button will open the Create New Secure Store Target Application page. In this dialog you have to enter the new Target Application ID, a display name, a contact email address and other application related details (see screenshot below).


Create New Secure Store Target Application - Windows Internet Explorer

http://win-31fru40oq82:34446/_admin/sssvc/ManageTargetApplicationInstance.aspx?scenarioid=managetargetapp&AppId=2992f63e-9(

Central Administration ▸ Create New Secure Store Target Application

Specify the settings for the new Secure Store Target Application. The settings that you specify here can be changed later at Secure Store Target Application page.

Target Application Settings

The Secure Store Target Application ID is a unique identifier. You cannot change this property after you create the Target Application.

The display name is used for display purposes only.

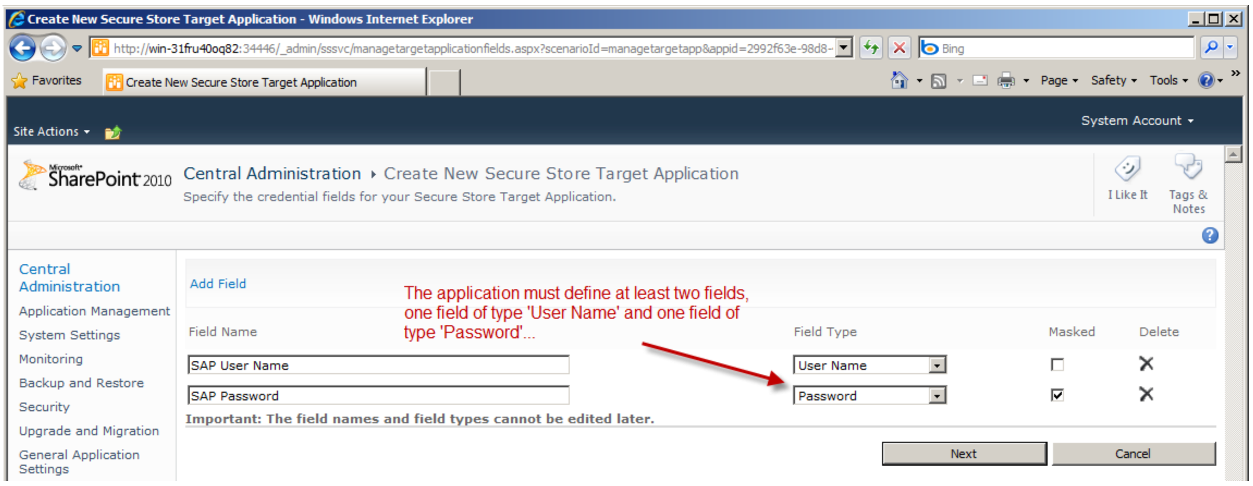
The contact e-mail should be a valid e-mail address of the primary contact for this Target Application.

The Target Application page URL can be used to set the values for the credential fields for the Target Application by individual users.

The Target Application type determines whether this application uses a group mapping or individual mapping. Ticketing indicates whether tickets are used for this Target Application. You cannot change this property after you create the Target Application.

Target Application ID	SAPCredentialsStore
Display Name	SAPCredentialsStore
Contact E-mail	jbaurle@parago.de
Target Application Type	Individual
Target Application Page URL	[✓ Use default page] Use custom page None

Next, the application fields must be defined:


Create New Secure Store Target Application - Windows Internet Explorer

http://win-31fru40oq82:34446/_admin/sssvc/managetargetapplicationfields.aspx?scenarioId=managetargetapp&appid=2992f63e-98d8-

Central Administration ▸ Create New Secure Store Target Application

Specify the credential fields for your Secure Store Target Application.

Add Field

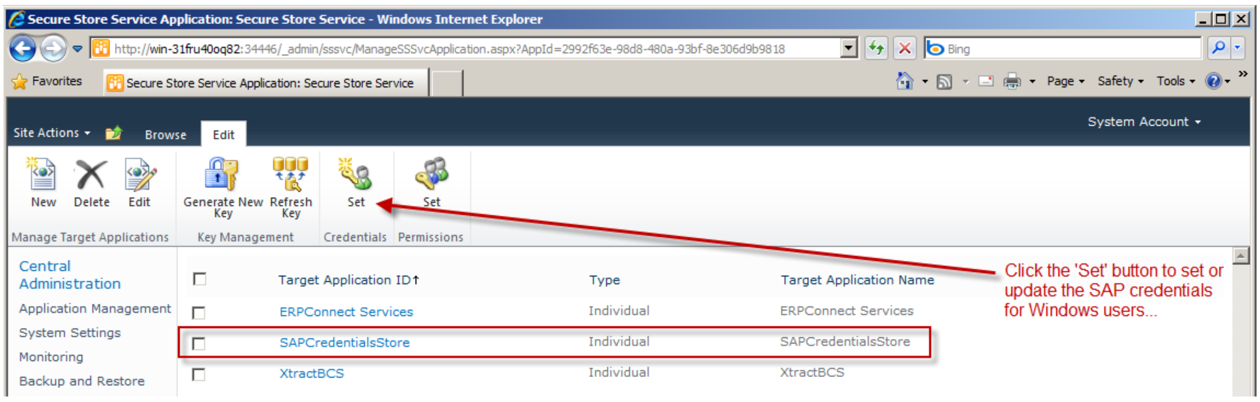
The application must define at least two fields, one field of type 'User Name' and one field of type 'Password'...

Field Name	Field Type	Masked	Delete
SAP User Name	User Name	☐	✕
SAP Password	Password	☑	✕

Important: The field names and field types cannot be edited later.

It's important to select the field type User Name and Password, because our implementation later on will check the target application for those two field types.

In the last dialog step the application administrator must be defined. After defining the administrator and clicking the Ok button SharePoint is creating a new secure store:

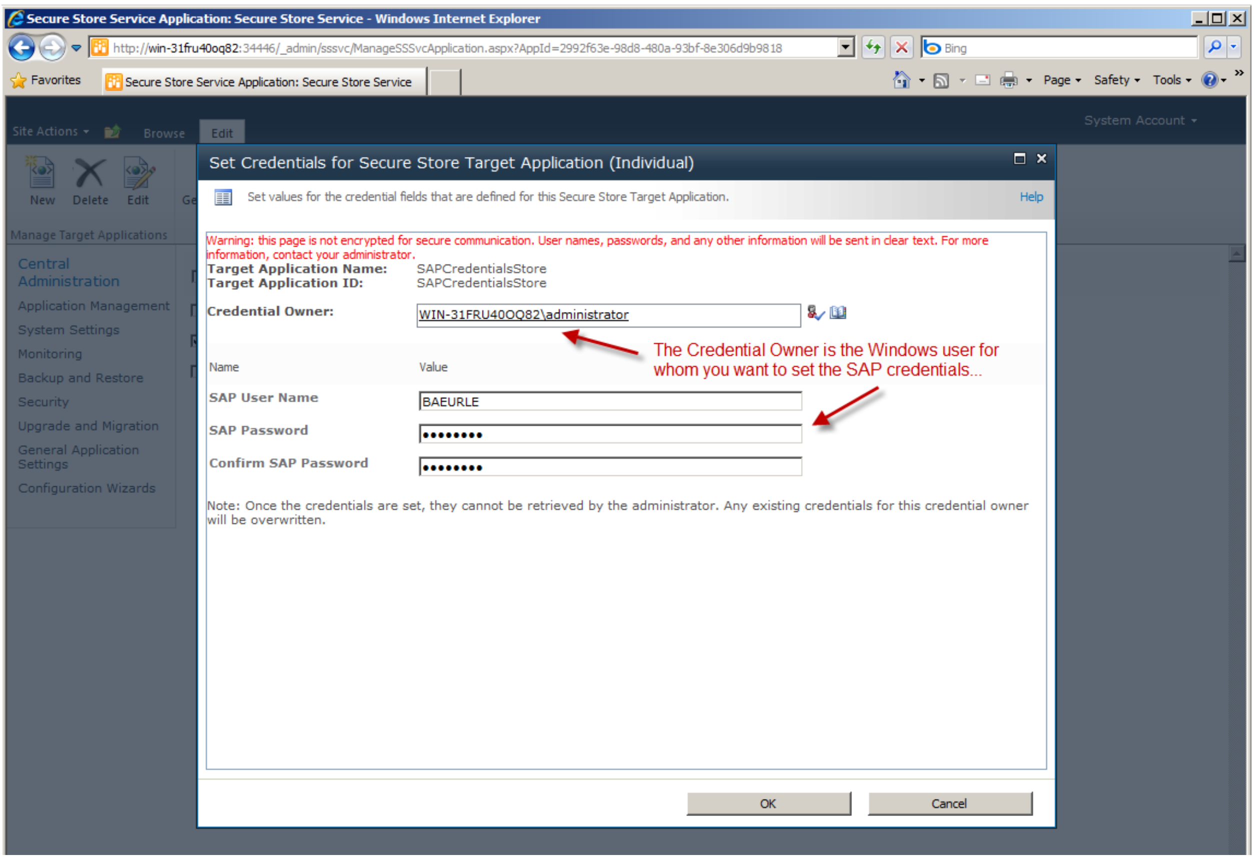

Secure Store Service Application: Secure Store Service - Windows Internet Explorer

http://win-31fru40oq82:34446/_admin/sssvc/ManageSSSvcApplication.aspx?AppId=2992f63e-98d8-480a-93bf-8e306d9b9818

Click the 'Set' button to set or update the SAP credentials for Windows users...

	Target Application ID↑	Type	Target Application Name
☐	ERPConnect Services	Individual	ERPConnect Services
☐	SAPCredentialsStore	Individual	SAPCredentialsStore
☐	XtractBCS	Individual	XtractBCS

Next, the Windows users must be mapped to the SAP user credentails. Therefore mark the checkbox for the newly created secure store SAPCredentialsStore and click the Set button in the toolbar. This opens the following dialog:


Secure Store Service Application: Secure Store Service - Windows Internet Explorer
http://win-31fru40oq82:34446/_admin/sssvc/ManageSSSvcApplication.aspx?AppId=2992f63e-98d8-480a-93bf-8e306d9b9818
Bing
Favorites
Secure Store Service Application: Secure Store Service
Page
Safety
Tools
Site Actions
Browse
Edit
System Account
New
Delete
Edit
Manage Target Applications
Central Administration
Application Management
System Settings
Monitoring
Backup and Restore
Security
Upgrade and Migration
General Application Settings
Configuration Wizards
Help
OK
Cancel

Set Credentials for Secure Store Target Application (Individual)

Set values for the credential fields that are defined for this Secure Store Target Application.

Warning: this page is not encrypted for secure communication. User names, passwords, and any other information will be sent in clear text. For more information, contact your administrator.

Target Application Name: SAPCredentialsStore
Target Application ID: SAPCredentialsStore

Credential Owner: WIN-31FRU40OQ82\administrator

The Credential Owner is the Windows user for whom you want to set the SAP credentials...

Name	Value
SAP User Name	BAEURLE
SAP Password	••••••••
Confirm SAP Password	••••••••

Note: Once the credentials are set, they cannot be retrieved by the administrator. Any existing credentials for this credential owner will be overwritten.

The Credential Owner is the Windows user for whom the SAP user credentials will be set. Enter the SAP username and password and click the Ok button to save them.

That's it !

Secure Store Programming

Accessing the secure store by code is simple. We implement a SecureStore class which will encapsulate all the access logic. A second class called SecureStoreCredentials contains the retrieved user credentials returned by the method GetCurrentCredentials of the SecureStore class.

But first, we need to create a Visual Studio 2010 SharePoint project and reference a couple of assemblies. You can directly enter the file paths in the Reference dialog (Browse tab) to add the assemblies:

Microsoft.BusinessData.dll
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\ISAPI\Microsoft.BusinessData.dll

Microsoft.Office.SecureStoreService.dll
C:\Windows\assembly\GAC_MSIL\Microsoft.Office.SecureStoreService\14.0.0.0__71e9bce111e9429c\Microsoft.Office.SecureStoreService.dll

The following code shows the SecureStore class implementation:

```
internal class SecureStore
{
  public string ApplicationId { get; private set; }

  public SecureStore(string applicationId)
  {
    if(string.IsNullOrEmpty(applicationId))
      throw new ArgumentNullException("applicationId");
    if(!IsApplicationValid(applicationId))
```

```
      throw new ArgumentException(string.Format("Target application with ID '{0}' is not
defined.", applicationId));

    ApplicationId = applicationId;
  }

  public SecureStoreCredentials GetCurrentCredentials()
  {
    SecureStoreProvider provider = new SecureStoreProvider { Context =
SPServiceContext.Current };
    string userName = string.Empty;
    string password = string.Empty;

    using(SecureStoreCredentialCollection data = provider.GetCredentials(ApplicationId))
    {
      foreach(ISecureStoreCredential c in data)
      {
        if(c != null)
        {
          if(c.CredentialType == SecureStoreCredentialType.UserName)
            userName = GetDecryptedCredentialString(c.Credential);
          else if(c.CredentialType == SecureStoreCredentialType.Password)
            password = GetDecryptedCredentialString(c.Credential);
        }
      }
    }

    if(string.IsNullOrEmpty(userName) || string.IsNullOrEmpty(password))
      throw new SecureStoreException("Credentials for the current Windows user are not valid
or not defined.");

    return new SecureStoreCredentials(userName, password);
  }

  public static bool IsApplicationValid(string applicationId)
  {
    if(string.IsNullOrEmpty(applicationId))
      throw new ArgumentNullException("applicationId");

    SecureStoreProvider provider = new SecureStoreProvider { Context =
SPServiceContext.Current };

    foreach(TargetApplication application in provider.GetTargetApplications())
    {
      if(application.ApplicationId == applicationId)
      {
        ReadOnlyCollection<ITargetApplicationField> fields =
provider.GetTargetApplicationFields(applicationId);
        bool existsUserNameDefinition = false;
        bool existsPasswordDefinition = false;

        foreach(TargetApplicationField field in fields)
        {
          if(field.CredentialType == SecureStoreCredentialType.UserName)
            existsUserNameDefinition = true;
          else if(field.CredentialType == SecureStoreCredentialType.Password)
            existsPasswordDefinition = true;
        }

        if(existsUserNameDefinition && existsPasswordDefinition)
          return true;
      }
    }

    return false;
  }

  public static string GetDecryptedCredentialString(SecureString secureString)
  {
    IntPtr p = Marshal.SecureStringToBSTR(secureString);

    try
    {
      return Marshal.PtrToStringUni(p);
    }
    finally
    {
      if(p != IntPtr.Zero)
```

```
            Marshal.FreeBSTR(p);
        }
    }
}
```

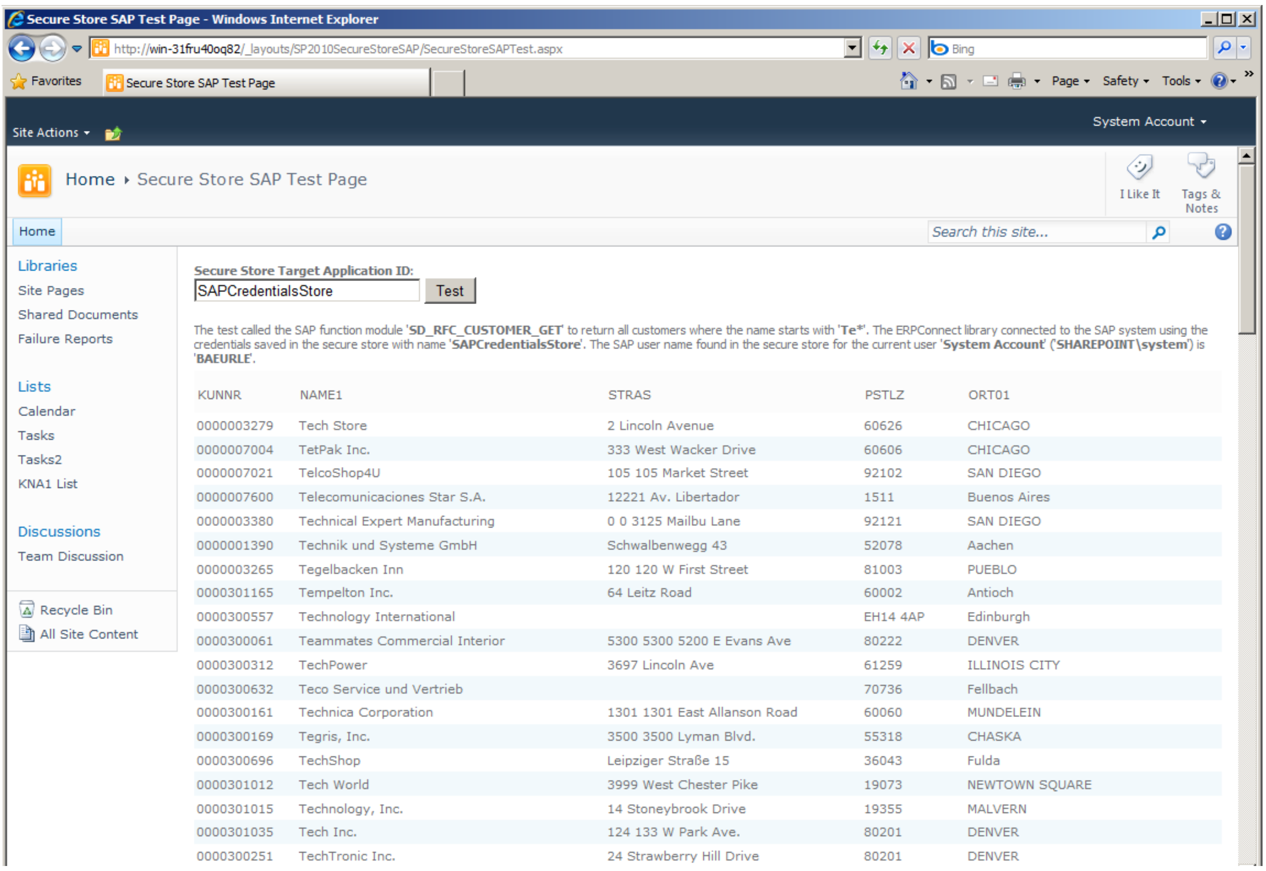
The constructor checks if an application ID is passed and if it's valid by calling the static method IsApplicationValid. In first place, the IsApplicationValid method is creating an instance of the SecureStoreProvider class to get access to Secure Store Service. The SecureStoreProvider class provides all methods to talk to the SharePoint service. Then, the method queries for all target applications and checks for the given application. If the application has been created and can be found, the method will analyse the application field definitions. The IsApplicationValid method then looks for two fields of type User Name and Password (see above).

The GetCurrentCredentials method is actually trying to get the SAP user credentials from the store. The method is creating an instance of the SecureStoreProvider class to get access to service and then calls the GetCredentials method of the provider class. If credentials are available the method decrypt the username and password from type SecureString using the internal method GetDecryptedCredentialString. It then will wrap and return the data into an instance of the SecureStoreCredentails class.

For more details of the implementation see the source code.

Accessing SAP Using The Secure Store Credentials

The sample and test code calls the SAP function module SD_RFC_CUSTOMER_GET to retrieve all customer data that match certain criteria (where NAME1 starts with Te*):



Secure Store Target Application ID: SAPCredentialsStore

The test called the SAP function module '**SD_RFC_CUSTOMER_GET**' to return all customers where the name starts with '**Te***'. The ERPConnect library connected to the SAP system using the credentials saved in the secure store with name '**SAPCredentialsStore**'. The SAP user name found in the secure store for the current user '**System Account**' ('**SHAREPOINT\system**') is '**BAEURLE**'.

KUNNR	NAME1	STRAS	PSTLZ	ORT01
0000003279	Tech Store	2 Lincoln Avenue	60626	CHICAGO
0000007004	TetPak Inc.	333 West Wacker Drive	60606	CHICAGO
0000007021	TelcoShop4U	105 105 Market Street	92102	SAN DIEGO
0000007600	Telecomunicaciones Star S.A.	12221 Av. Libertador	1511	Buenos Aires
0000003380	Technical Expert Manufacturing	0 0 3125 Mailbu Lane	92121	SAN DIEGO
0000001390	Technik und Systeme GmbH	Schwalbenwegg 43	52078	Aachen
0000003265	Tegelbacken Inn	120 120 W First Street	81003	PUEBLO
0000301165	Tempelton Inc.	64 Leitz Road	60002	Antioch
0000300557	Technology International		EH14 4AP	Edinburgh
0000300061	Teammates Commercial Interior	5300 5300 5200 E Evans Ave	80222	DENVER
0000300312	TechPower	3697 Lincoln Ave	61259	ILLINOIS CITY
0000300632	Teco Service und Vertrieb		70736	Fellbach
0000300161	Technica Corporation	1301 1301 East Allanson Road	60060	MUNDELEIN
0000300169	Tegris, Inc.	3500 3500 Lyman Blvd.	55318	CHASKA
0000300696	TechShop	Leipziger Straße 15	36043	Fulda
0000301012	Tech World	3999 West Chester Pike	19073	NEWTOWN SQUARE
0000301015	Technology, Inc.	14 Stoneybrook Drive	19355	MALVERN
0000301035	Tech Inc.	124 133 W Park Ave.	80201	DENVER
0000300251	TechTronic Inc.	24 Strawberry Hill Drive	80201	DENVER

The following code shows the implementation of the Test button click event:

```
protected void OnTestButtonClick(object sender, EventArgs e)
{
  string licenseKey = "<LICENSEKEY>";
  string connectionStringFormat = "CLIENT=800 LANG=EN USER={0} PASSWD={1} ASHOST=HAMLET ...";

  R3Connection connection = null;

  try
  {
    LIC.SetLic(licenseKey);

    ...

    SecureStoreCredentials credentials =
      new SecureStore(ApplicationID.Text).GetCurrentCredentials();
    string connectionstring =
      string.Format(connectionStringFormat, credentials.UserName, credentials.Password);

    connection = new R3Connection(connectionstring);
    connection.Open();

    RFCFunction function = connection.CreateFunction("SD_RFC_CUSTOMER_GET");
    function.Exports["NAME1"].ParamValue = "Te*";
    function.Execute();

    ResultGrid.DataSource = function.Tables["CUSTOMER_T"].ToADOTable();
    ResultGrid.DataBind();

    OutputLabel.Text = string.Format("The test called...",
      ApplicationID.Text, Web.CurrentUser.Name, Web.CurrentUser.LoginName,
      credentials.UserName);
  }
  catch(Exception ex)
  {
    WriteErrorMessage(ex.Message);
  }
  finally
  {
    if(connection != null && connection.Ping())
      connection.Close();
  }
}
```

The interesting part is the retrieval of the SAP user credentials from the secure store defined in the text box named ApplicationID. The application ID will be passed as parameter to the constructor of the SecureStore class. After creating the instance the method GetCurrentCredentials will be called to ask the store for the credentials of the current Windows user.

After the user credential query has been successfully executed the SAP connection string will be constructed. Then the connection string will then be used to create an instance of the R3Connection class to connect with the SAP system. The remaining code is just calling the function module SD_RFC_CUSTOMER_GET and binding the result to the SPGridView.

That's all.

Summary

This article has shown, how easy it can be to integrate Secure Store Services into your own SharePoint application and keeping the security standards high while communicating with SAP.

Contact Information

If you have any feedback or suggestions, please feel free to contact me:

Jürgen Bäurle
jbaurle@parago.de
http://www.parago.de/jbaurle

Parago Media GmbH & Co. KG
Im Wengert 3 | 71336 Waiblingen, Germany | Phone +49.7146.861803 | Internet http://www.parago.de