Windows Identity Foundation (WIF)
An overview of changes between Beta2 & Release Candidate

Windows Identity Foundation Team

Microsoft Corp

Table of contents

1. Overview

This document highlights the changes made between Beta 2, which was released during TechEd 2009 in May'09, and Release Candidate, which is releasing in Oct'09.

The audience for this document is the developers who evaluated the May 2009 Beta (Beta 2) and are now evaluating the RC from Oct 2009 and who are familiar with the basic components available in the Windows Identity Foundation (WIF), previously referred to as the Geneva Framework.

WIF provides a set of APIs for .NET developers to build a claims-aware and federation-enabled application by externalizing the authentication logic to a security token service (STS). It also provides a set of interfaces for building security token services.

Beta 1 was released in October 2008 and Beta 2 was released in May 2009 and Release Candidate is released in November 2009.

Note that this document provides an overview on the major changes happened between Beta-2 and RC. For more information on specific features, you may refer to the product documentation, the Framework White paper, or other collateral materials on the WIF Team Site.

2. New features added in Release Candidate

2.1 Replay detection of Bearer SAML tokens

- A replay detection cache has been introduced to mitigate the replay of bearer SAML 1.1 and SAML 2.0 tokens. This feature ensures bearer tokens cannot be replayed during the configured caching interval for the token or the token's validity period, whichever is shorter. You can enable this feature by setting DetectReplayTokens property in ServiceConfiguration instance to true.

2.2 WSTrustChannelFactory & WSTrustClient

- To mirror the WCF programming model with Channels, WSTrustClient has been replaced with WSTrustChannelFactory and WSTrustChannel. The WSTrustChannelFactory derives from a WCF Channel Factory, and the WSTrustChannel derives from a WCF IChannel. Explicit control of these classes is provided so that middle tier applications that support identity delegation have the flexibility to use these classes in any number of ways to suit their particular functional or performance needs.

2.3 GetWindowsIssuerName() method in IssuerNameRegistry

- To differentiate between a claim that came from Windows Token (or a SAML token that is mapped to Windows) versus one that got instantiated using Claim constructor (i.e. new Claim ()), a new method named GetWindowsIssuerName() has been added to the IssuerNameRegistry. This allows changing the issuer value used by default for all claims that originate from a Windows token. In Beta-2, the issuer is LocalAuthority and with this extensibility point in RC one can choose to differentiate the issuer name based on the origin of the claim.

- The following matrix summarizes the expected issuer names for various authentication modes:
  Regular claims = Claims that are augmented in STS (such as Age claim, Zip code claim etc.,)
  Windows claims = Claims that are result of Windows tokens being used

Authentication Mode	Issuer for regular claims	Issuer for Windows claims	Extensibility point to use to customize the issuer name
SAML	STS Certificate	N/A	GetIssuerName()
SAML map to Windows	STS Certificate	LocalAuthority	GetWindowsIssuerName()
X509	X509 Issuer Certificate	N/A	GetIssuerName()
X509 map to Windows	X509 Issuer Certificate	LocalAuthority	GetWindowsIssuerName()
Username	LocalAuthority	N/A	GetIssuerName()
Username map to Windows	LocalAuthority	LocalAuthority	GetWindowsIssuerName()
Kerberos	N/A	LocalAuthority	GetWindowsIssuerName()
SPNego	N/A	LocalAuthority	GetWindowsIssuerName()

2.4 Cryptography updates

- The default hashing algorithm used by WIF is now SHA-256. In Beta 2, this used to be SHA-1.
    - Note that SHA-256 OID is not registered by default in W2K3 OS, follow the below instructions to register SHA-256 in W2K3 OS:
        - Browse to http://www.codeplex.com/clrsecurity/SourceControl/changeset/view/18423 and click "Download".
        - Unzip and build the downloaded Security.Cryptography\src\Security.Cryptography.csproj VS 2008 project.
        - Write a program to call Security.Cryptography.Oid2.RegisterSha2OidInformationForRsa().
            - Create a console application that references Security.Cryptography.dll
            - Add code to call Oid2.RegisterSha2OidInformationForRsa() and execute the console application.
- For the Issuer certificate of a SAML token, Peer Trust and/or ChainTrust validation is now performed by default. This can be changed to a different validation mode using a configuration property.
- WIF picks up crypto settings from the machine configuration and offers crypto agility.

2.5 Support for both SAML 1.1 and SAML 2.0 tokens at the same end point

- On a WCF relying party binding, if the token type is not specified, both the SAML 1.1 and SAML 2.0 tokens will be accepted. In Beta-2, an unspecified token type meant that only SAML 1.1 would be accepted.

2.6 Tracing

- For easy diagnosibility, tracing statements have been added in various places of the framework's processing pipeline. The following code snippet enables WIF Tracing and logs the traces to C:\logs\WIF.xml. You can use the SvcTraceViewer tool to view these traces.

```
<system.diagnostics>
  <sources>
    <source name="Microsoft.IdentityModel" switchValue="Verbose">
```

```
        <listeners>
          <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
               initializeData="C:\logs\WIF.xml" />
        </listeners>
      </source>
    </sources>
    <trace autoflush="true" />
  </system.diagnostics>
```

2.7 New event AuthorizationFailed triggered from WS-FAM module

- A new AuthorizationFailed event has been added to the Federated Authentication Module's pipeline. This event is triggered whenever the user is not authorized to access a resource. Application developers can choose to register for this event and perform the necessary actions for scenarios where users need to be re-authenticated.

2.8 SecurityTokenCache

- ASP.NET cookies have a new session mode where the entire cookie is not written to the browser but only a cookie artifact is written.
- SessionSecurityToken has a new property called *IsSessionMode*. Setting this property to false in ASP.NET case will enable session cookies to be written out.
- SecurityTokenCache has two new interfaces to TryAddAllEntries and TryRemoveAllEntries.
- New SctCookieSerializer class that implements cookie serialization code has been introduced. This can be overridden to create a custom cookie serializer or to cache the cookies in a database in a web farm scenario.
- For breaking changes in SecurityTokenCache refer to "SecurityTokenCache & Session Security Token Handler" section below.

3. Changes made to features existed in Beta-2

3.1 Controls

- The "Federated Passive Token Service" control has been cut and instead an object model is introduced to programmatically build a WS-Federation Passive STS's. Visual Studio Templates and Samples are updated to use this new object model.
- The "Federated Passive Sign-In" control can now read from WS-FAM's configuration settings specified in web.config. To enable this functionality set `UseFederationPropertiesFromConfiguration` property to 'true'. The behavior of reading its own federation properties still exists and it is the default behavior of this control.

3.2 FedUtil

- The Visual Studio menu item "Modify STS Reference…" is renamed to "Add STS Reference…". Additionally, the restriction to show this menu item only for Web Site projects has been relaxed such that the menu item appears for all project types.
- FedUtil's reentrant behavior has changed such that the application's configuration file is updated only when there is a change to it from the existing version, for example federation metadata of the STS that the application trusts has been updated. In Beta 2, the behavior was to always re-write the application's configuration file regardless of any changes.

- FedUtil is made accessible directly from “Federated Passive Sign-In” control.
    - This feature of “Federated Passive Sign-In” control allows developers to easily federate an application from Visual Studio’s design view i.e. drag-n-drop the federated passive sign-in control and then right-click on the control in Design View and select “Add STS Reference…” smart tag, which would invoke FedUtil. Upon successful execution of FedUtil application is now configured to use the WS-Federation configurations in <FederatedAuthentication> element in web.config.
    - Note that this “Federated Passive Sign-In” control’s smart tag is available only when FedUtil tool is available in the system. FedUtil tool is included as part of WIF SDK package.

3.3 Bootstrap token in IClaimsIdentity

- ClaimsIdentity has now been updated to contain the bootstrap token of the primary identity as a property named “BootstrapToken”. In Beta-2 one can call GetBootstrapTokens() method for accessing the bootstrap tokens on a ClaimsPrincipal and this method has been removed and instead it is recommended to use this
- Bootstrap tokens are no longer saved by default. To save bootstrap tokens you would need to set ServiceConfiguration.SaveBootstrapTokens property to ‘true’.

3.4 Properties in Claims OM

- The ‘delegate’ property in claims OM has been renamed to ‘actor’. The namespace of this property is changed to http://schemas.xmlsoap.org/ws/2009/09/identity/claims/actor.
- The namespace for ‘OriginalIssuer’ property in the claims OM has been changed to http://schemas.xmlsoap.org/ws/2009/09/identity/claims/originalissuer.

3.5 SecurityTokenCache & Session Security Token Handler

- The SecurityContextSecurityTokenHandler and SessionSecurityTokenHandler have been merged into a single SecurityTokenHandler called SessionSecurityTokenHandler.
- The Keys property of the SecurityTokenCache has been removed.
    - SecurityTokenCache is keyed based of a SecurityTokenCacheKey. The Key contains four parts ContextId, KeyGeneration, EndpointId and IsSessionMode.
- The Session cookies have an endpoint Identifier stamped on this. When the cookies are received the endpoint identifier is validated against the current endpoint.

3.6 No propagation of service-level settings to ActAs and OnBehalfOf tokens

- In Beta-2, service-level settings, for example: Audience validation mode is set to ‘Always’, were propagated to all token handler collections including the ones that handle ActAs or OnBehalfOf tokens. In RC, there is no automatic propagation of service-level settings to the token handlers that handle ActAs or OnBehalfOf tokens. Settings specific to these token handlers need to be configured separately.
- For example, if you had an ActAs tokens support and had the following configuration to reflect in both regular tokens and ActAs tokens in Beta-2,

```
<microsoft.identityModel>
   <service>
     …
     <audienceUris mode="Always" >
       <clear/>
     </audienceUris>
     …
      </service>
 </microsoft.identityModel>
```

for RC you would need to modify the configuration like below:

```
<microsoft.identityModel>
  <service>
   …
   <audienceUris mode="Always" >
     <clear/>
   </audienceUris>
   …
   <securityTokenHandlers name="ActAs">
       <securityTokenHandlerConfiguration>
         <audienceUris mode="Never" >
       </securityTokenHandlerConfiguration>
   </securityTokenHandlers>
   …
    </service>
  </microsoft.identityModel>
```

3.7 WSTrustClient

- As mentioned above in “WSTrustChannelFactory” section, “WSTrustClient” object model has been replaced with “WSTrustChannelFactory” and “WSTrustChannel” object model.

3.8 IssuerNameRegistry

- Custom implementations of IssuerNameRegistry can now return “null” or “empty string” from the GetIssuerName() method instead of throwing an exception when an ‘un-trusted issuer’ is encountered. Custom security token handler implementations that invoke the IssuerNameRegistry will need to detect this condition and behave accordingly.
- The Claim type “SamlIssuername” is removed because it is redundant to “Issuer” property in claims object model.

3.9 Name changes

- The product name has been changed to “Windows Identity Foundation” from Geneva Framework.
- The Claims object model property “IClaimsIdentity.Delegate” is renamed to “IClaimsIdentity.Actor”.
- Geneva Token Service (GTS) is renamed to “Claims to Windows Token Service” (C2WTS).

3.10 Setup updates

- Setup packages for Vista/W2K8/W2K8 R2/Win7 have moved to being a CBS package, instead of being an MSI based package, hence the file extension of the setup package has been changed from being a “.msi” to a “.msu”.
- WIF installation no longer appears in the Installed Programs list under Control Panel and “Add/Remove Programs” applet. It appears in the ‘Installed Updates’ list, which you can select by clicking “View Installed Updates” option in the “Programs and Features” applet.