Windows Server® 2008 R2

Using AD FS 2.0 for interoperable SAML 2.0-based federated Web Single Sign-On

Microsoft France

Published: June 2012

Version: 1.0a

Authors*:* Jean-Marie Thia (UPMC*)*, Philippe Beraud (Microsoft France)

Abstract

Through its support for the WS-Federation and Security Assertion Markup Language (SAML) 2.0 protocols, Microsoft Active Directory Federation Services (AD FS) 2.0 provides claims-based, cross-domain Web Single Sign-On (SSO) interoperability with non-Microsoft federation solutions.

Building on existing documentation, this document is intended to provide a better understanding of the different configuration elements to take into account when using AD FS 2.0 for interoperable SAML 2.0-based federated Web SSO.

This document is intended for developers and system architects who are interested in understanding the basic modes of SAML 2.0 interoperability with AD FS 2.0.

UPMC SORBONNE UNIVERSITÉS Microsoft

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

1 Introduction

1.1 Objectives of this paper

By leveraging several OASIS standards like the Security Assertion Markup Language (SAML) 2.0[1] protocol, Microsoft Active Directory Federation Services (AD FS) 2.0 provides claims-based, cross-domain, Web Single Sign-On (SSO) interoperability with third-party federation solutions.

> **Important note:**
>
> *The AD FS role available in Windows Server 2008 (R2) doesn't correspond to AD FS 2.0; this is the previous version 1.1 instead. The AD FS 2.0 software package for your specific operating system version (either Windows Server 2008 or Windows Server 2008 R2) is the AdfsSetup.exe setup file. To download this file, go to Active Directory Federation Services 2.0 RTW[2].*

Wikipedia[3] defines federation as follows:

> "*Federated identity, or the 'federation' of identity, describes the technologies, standards and use-cases which serve to enable the portability of identity information across otherwise autonomous security domains. The ultimate goal of identity federation is to enable users of one domain to securely access data or systems of another domain seamlessly, and without the need for completely redundant user administration.*"

SAML is an XML-based standard for exchanging authentication and authorization data between security domains, that is, between an Identity Provider (IdP), a producer of (SAML) assertions, and a Service Provider (SP), a consumer of assertions.

Microsoft has recently published, thanks to author Dave Martinez, a series of step-by-step guides[4] on configuring AD FS 2.0 to interoperate with SAML 2.0 products with the SAML 2.0 HTTP POST binding:

- AD FS 2.0 STEP-BY-STEP GUIDE: FEDERATION WITH CA FEDERATION MANAGER[5];
- AD FS 2.0 STEP-BY-STEP GUIDE: FEDERATION WITH ORACLE IDENTITY FEDERATION[6];
- AD FS 2.0 STEP-BY-STEP GUIDE: FEDERATION WITH SHIBBOLETH 2 AND THE INCOMMON FEDERATION[7].

These guides complete formerly published white papers at the AD FS 2.0 Beta timeframe:

- BOOSTING INTEROPERABILITY AND COLLABORATION ACROSS MIXED TECHNOLOGY ENVIRONMENTS – STANDARDS-BASED IDENTITY FEDERATION SOLUTIONS FROM MICROSOFT AND NOVELL[8];

[1] Security Assertion Markup Language (SAML) 2.0: http://go.microsoft.com/fwlink/?LinkId=193996

[2] Active Directory Federation Services 2.0 RTW: http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=10909

[3] : http://en.wikipedia.org/wiki/Federated_identity

[4] AD FS 2.0 Step-by-Step and How To guides: http://technet.microsoft.com/en-us/library/dd727938(WS.10).aspx

[5] AD FS 2.0 STEP-BY-STEP GUIDE: FEDERATION WITH CA FEDERATION MANAGER: http://go.microsoft.com/fwlink/?LinkId=196684

[6] AD FS 2.0 STEP-BY-STEP GUIDE: FEDERATION WITH ORACLE IDENTITY FEDERATION: http://go.microsoft.com/fwlink/?LinkId=198368

[7] AD FS 2.0 STEP-BY-STEP GUIDE: FEDERATION WITH SHIBBOLETH 2 AND THE INCOMMON FEDERATION: http://go.microsoft.com/fwlink/?LinkId=204784

- Microsoft “Geneva” Server and Sun Open SSO: Enabling Unprecedented Collaboration across Heterogeneous IT Environments[9].

as well as webcasts on MSDN Channel 9 like Caleb Baker on Geneva Server and SAML 2.0 Interoperability[10].

This paper, developed in collaboration with the French University Pierre and Marie Curie (UPMC) in Paris and, more specifically Jean-Marie Thia, leverages some of the information contain in these documents and webcasts and, interestingly enough provides, on that basis, “real world” tips for enabling federation with third-party solutions when configuring AD FS 2.0, as an IdP or a SP in a SAML 2.0-based federation.

For that purposes, beyond a short depiction of AD FS 2.0 to introduce key concepts for the rest of the paper, it gives an understanding of:

- What the SAML 2.0 standard is all about,
- What its support makes possible,
- The common “gotchas” that may be encountered along with AD FS 2. 0.

So that federation projects involving AD FS 2.0 in this context can be more easily completed, and consequently enabling customers to realize the full interoperability potential of AD FS 2.0.

Furthermore, as of writing, the Update Rollup 2 for AD FS 2.0 is available. This Update Rollup includes hotfixes and updates for AD FS 2.0 RTW that are of special interest in the context of this paper for the support of the SAML 2.0 protocol.

> **Important note:**
>
> *For more information about this Update Rollup and its download, please see article 2681584 Description of Update Rollup 2 for Active Directory Federation Services (AD FS) 2.0[11].*

1.2 Organization of this paper

To cover the whole set of considerations relating to the support of the OASIS SAML 2.0 standard in the context of AD FS 2.0, this document adopts an organization according to the following themes, each of them being addressed as part of an eponymous section:

- An understanding of the SAML 2.0 standard;
- A brief overview of Active Directory Federation Services (AD FS) 2.0;
- The eleven interoperability “Gotchas” you should be aware of.

[8] Boosting interoperability and collaboration across mixed technology environments – Standards-based identity federation solutions from Microsoft and Novell: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9eb1f3c7-84da-40eb-b9aa-44724c98e026

[9] Microsoft “Geneva” Server and Sun Open SSO: Enabling Unprecedented Collaboration across Heterogeneous IT Environments: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=9eb1f3c7-84da-40eb-b9aa-44724c98e026

[10] Caleb Baker on Geneva Server and SAML 2.0 Interoperability: http://channel9.msdn.com/shows/Identity/Caleb-Baker-on-Geneva-Server-and-SAML-20-Interoperability/

[11] Article 2681584 Description of Update Rollup 2 for Active Directory Federation Services (AD FS) 2.0: http://support.microsoft.com/kb/2681584

Finally, references provided in the appendixes enable to easily search the Web for additional information.

1.3 About the audience

Federated identity in general is a broad topic, with many facets, depths of understanding, protocols, standards, tokens, etc. This paper addresses the SAML 2.0 topic only from the AD FS 2.0 perspective and from both conceptual and technical levels.

> **Note:**
>
> *For additional information on AD FS 2.0 in addition to the content of this paper, please refer to the product documentation[12], the dedicated AD FS 2.0 Q&A forum[13] and the product team weblog[14].*

This document is intended for system architects and IT professionals who are interested in understanding the basics of interoperability between AD FS 2.0 and other SAML 2.0-based implementations.

1.4 Terminology used in this paper

Throughout the rest of this document, there are numerous references to federation concepts that are called by different names in the Microsoft products and/or technologies like AD FS 2.0 or Windows Identity Foundation (WIF) 1.0[15] and the OASIS SAML 2.0 standard. The following table assists in drawing parallels between the two.

Concept	Microsoft name	SAML 2.0 name
XML document sent from the federation party that is managing users to the federation party that is managing an application during an access request describing a user	Security Token	Assertion
Partner in a federation that creates security tokens for users	Claims Provider	Identity Provider
Partner in a federation that consumes security tokens for providing access to applications	Relying Party	Service Provider
Data about users that is sent inside security tokens	Claims	Assertion statements

[12] AD FS 2.0 TechNet documentation: http://technet.microsoft.com/en-us/library/adfs2(WS.10).aspx

[13] AD FS 2.0 Q&A forum: http://social.msdn.microsoft.com/Forums/en-US/Geneva/threads

[14] AD FS 2.0 product team weblog: http://blogs.msdn.com/b.card

[15] Microsoft Windows Identity Foundation (WIF) download: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=eb9c345f-e830-40b8-a5fe-ae7a864c4d76

1.5 About the live demo at the MTC Paris

Microsoft Technology Centers[16] (MTC) are collaborative environments that provide access to innovative technologies and world-class expertise, enabling our customers and partners to envision, design, and deploy solutions that meet their needs.

Since 2004, MTC Paris, is part of these global centers designed to provide our customers with an actionable set of steps on how a Microsoft solution can assist them in achieving their key business objectives. Inside this facility, MTC architects and Microsoft technologies Experts, through a discovery process and scenario-based demonstrations running in MTC datacenter, play a critical role in addressing our customers' challenges.

Interestingly enough, MTC Paris is hosting and running Microsoft France Interop Lab in order to allow customers to see and understand how Microsoft solutions and action can interoperate with other technologies or products around several topics such as : advanced Web services, PHP, Java, SAP, application lifecycle management and last but not least security & identity.

In this lab, customers and partners test multi-vendor technical configurations in order to adapt solutions to their needs in terms of operational interoperability. MTC Paris hosts more than 20 competing players' solutions. These solutions are deployed on MTC Paris datacenter infrastructure which is built upon more than 300 servers and 200 terabytes storage. Working with many competing publishers, we facilitate the integration of heterogeneous systems. Thus interoperability becomes a guarantee of integration for our customers and enables them to create value by maximizing the investment in innovation.

In order to ensure both identity portability and security in a loosely coupled environment, it is fundamental to master the identity management part in each involved security realm for the considered scenario. As aforementioned, the Microsoft platform natively offers a series of products and technologies to sustain the notion of claim-based identity: ready to use enterprise-class Claims Provider Security Token Service (STS), Framework for building claims-aware applications and services (including authentication, access control, auditing, etc.), etc. In "real world" heterogeneous environments, these components haven't no choice rather than being truly interoperable.

To illustrate this interoperability, the MTC Paris Security and Identity Management Interop Lab proposes a permanent dedicated platform offering multiple identity management scenarios, and more especially the one describes in this paper, i.e. the federated collaboration scenario by using the SAML 2.0 protocol with the SAML 2.0 HTTP POST binding notably based on Oracle Open SSO, Internet2 Shibboleth 2, Microsoft AD FS 2.0 for identity solutions and Microsoft SharePoint 2010, and Microsoft Outlook Web Access 2010/Exchange 2010 for the exposed collaboration resources.

[16] Microsoft Technology Centers: http://microsoft.com/mtc

2 An understanding of the SAML 2.0 standard

The Security Assertion Markup Language (SAML) 2.0[17] standard is an XML-based standard for exchanging authentication and authorization data between security domains/realms, that is, between an Identity Provider (IdP), a producer of (SAML) assertions, and a Service Provider (SP), a consumer of assertions.

The SAML standard is governed by the OASIS Security Services (SAML) Technical Committee (TC)[18] from whom Microsoft Corporation is a TC participant.

This standard released in 2005 is being broadly adopted across all relevant segments and is consequently already supported by all IDA vendors, including Microsoft.

SAML 2.0 results from the convergence of the previous version of the standard itself, i.e. SAML 1.1, and from the following two extensions/specifications based on it forming the foundation for the standard:

- Liberty Identity Federation Framework (ID-FF) 1.2[19];
- Internet2 Shibboleth 1.3.

The Liberty Alliance project[20] was formed in September 2001 by approximately 30 organizations to establish open standards, guidelines and best practices for identity management. It has released, among other thing, the ID-FF specification to address identity federation.

Like SAML 1.1, the ID-FF specification is a cross-domain, browser-based, Single Sign-On (SSO) framework. In addition, the specification defined the notion of circle of trust (CoT), where each participating domain/realm is trusted to accurately document the processes used to identify a user, the type of authentication used, and any policies associated with the resulting authentication credentials. Other members of the circle of trust may examine these policies to determine whether to trust such information. The CoT represents a static trust schema. Liberty Alliance contributed its federation specification to OASIS.

In an effort to grow the identity marketplace, Liberty Alliance also introduced the Liberty Interoperable certification program[21], operated by the Drummond group[22], and designed to test commercial and open source products against its own specifications like the aforementioned ID-FF specification and published standards like the SAML standard to assure base levels of interoperability between products.

[17] Security Assertion Markup Language (SAML) 2.0: http://go.microsoft.com/fwlink/?LinkId=193996

[18] OASIS Security Services (SAML) Technical Committee (TC): http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[19] Liberty Identity Federation Framework (ID-FF) 1.2 : http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications/?f=resource_center/specifications/liberty_alliance_id_ff_1_2_specifications

[20] Liberty Alliance project: http://www.projectliberty.org/

[21] Liberty Interoperable certification program: http://www.projectliberty.org/liberty/liberty_interoperable

[22] Drummond group Web site: http://www.drummondgroup.com/

As of writing, over 80 solutions from numerous vendors and organizations worldwide have passed testing. This is the case of AD FS 2.0 (see next chapter).

In June 2009, all Liberty Alliance work and related materials have been contributed to the Kantara Initiative[23] (kan-TAR-a: swahili for "bridge"; arabic roots in "harmony"). The project Web site remains as an archive of the work of the Liberty Alliance.

Kantara Initiative is working to "*bridge and harmonize the identity community with actions that will help ensure secure, identity-based, online interactions while preventing misuse of personal information so that networks will become privacy protecting and more natively trustworthy environments*".

As a consequence of this transition, the SAML 2.0 interoperability certification program formerly run from Liberty Alliance is now handled by the Kantara Initiative.

Shibboleth (, as a reference to the Hebrew word "shibbóleth" and the related Biblical use, i.e. to discover hiding members of the opposing group,) is an Internet2/MACE (Middleware Architecture Committee for Education)[24] project. The project (http://shibboleth.internet2.edu) refers to both a specification and an open-source implementation for federated identity-based authentication and authorization infrastructure that implements them as a distributed system. Shibboleth was designed to fill higher education needs in terms of identity federation and attributes propagation for a number of partners.

As a specification, Shibboleth 1.3 is an extension of the SAML 1.1 to define a protocol to exchange security information in order to implement Web Single Sign-On.

As an implementation, the current version released in 2008, i.e. Shibboleth 2 now builds on the SAML 2.0 standard.

2.1 A suite of specifications

The SAML 2.0 standard is a suite of specifications and, as such, comprises a set of normative and non-normative documents:

- SAML V2.0 Executive Overview[25] (SAMLExecOvr);
- Security Assertion Markup Language (SAML) V2.0 Technical Overview[26] (SAMLTechOvw);
- Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0[27] (SAMLCore), the core specification*;*

[23] Kantara initiative: http://kantarainitiative.org/

[24] Internet2/MACE (Middleware Architecture Committee for Education): http://middleware.internet2.edu/MACE

[25] SAML V2.0 Executive Overview: http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf

[26] SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0 TECHNICAL OVERVIEW: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[27] Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

- BINDINGS FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[28] (SAMLBind), which maps the SAML messages onto the standard messaging or communication protocols;
- PROFILES FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[29] (SAMLProf), the use cases or the "How-to" in regards to the use of SAML to solve specific problems of the extended enterprise;
- METADATA FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[30] (SAMLMeta), the configuration data (endpoint URLs, key material for verifying signatures, etc.) to establish trusts between SAML entities;
- AUTHENTICATION CONTEXT FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[31] (SAMLAuthnCxt), a detailed description of the user authentication mechanisms;
- CONFORMANCE REQUIREMENTS FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[32] (SAMLConform), the operational modes for the SAML 2.0 implementations;
- SECURITY AND PRIVACY CONSIDERATIONS FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[33] (SAMLSec), an analysis of both the security and the privacy in SAML 2.0;
- GLOSSARY FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0[34] (SAMLGloss), the terminology used in SAML 2.0.

> **Note:**
>
> *Following the release of the SAML 2.0 standard in 2005, the OASIS SAML TC has continued work on several enhancements. These documents are available from the OASIS SAML TC Web site.*

In order to have a good understanding of the standard and be able to further dig into the nitty-gritty details if needed to, for instance solve interoperability issue, we strongly advise to start reading the non-normative SAMLTechOvw document, which, as its name indicates, gives the key to understand the standard and its organization and ramification.

The critical aspects of SAML 2.0 are covered in detail in the normative documents SAMLCore, SAMLBind, SAMLProf, and SAMLConform.

SAML 2.0 defines XML-based **assertions** and **protocols**, **bindings**, and **profiles**. The term SAML Core, in relationship with the SAMLCore core specification, refers to the general syntax and semantics of SAML assertions as well as the protocol used to request and transmit those assertions from one system entity to another. SAML assertions are usually transferred from an IdP to a SP.

[28] BINDINGS FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[29] PROFILES FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[30] METADATA FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[31] AUTHENTICATION CONTEXT FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf

[32] CONFORMANCE REQUIREMENTS FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf

[33] SECURITY AND PRIVACY CONSIDERATIONS FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf

[34] GLOSSARY FOR THE OASIS SECURITY ASSERTION MARKUP LANGUAGE (SAML) V2.0: http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf

2.2 SAML 2.0 Assertions

A SAML 2.0 assertion is a (signed) (security) token and can be seen as the vehicle/container for (security) information. Such assertions contain beyond a subject and conditions, which apply to the assertions, statements or claims that SPs typically use to make or derive access control decisions. Three types of statements are provided by SAML:

- Authentication statement, which asserts that the security principal has been authenticated by the IdP at a particular time using a particular method of authentication. An authentication context may also be disclosed as such a statement;
- Attribute statement, which asserts that a subject is associated with certain attributes. An attribute is typically a string name-value pair. Relying parties use these attributes or claims to make or derive access control decisions;
- Authorization decision statement, which asserts that a subject is allowed to perform a specific action on specific resource given specific evidence;

> **Note:**
>
> *The vocabulary is intentionally limited to promote another OASIS standard instead: the eXtensible Access Control Markup Language (XACML) governed by the eponym OASIS TC[35]. XACML is a XML-based declarative access control policy language and a processing model describing how to interpret the policies.*

In the context of this paper, the SAML assertion we have to consider is the so-called "bearer" assertion, a short-lived bearer token (, i.e. without a proof of possession,) issued by an IdP to a SP. Such an assertion includes both an authentication statement and an attribute statement.

2.3 SAML 2.0 protocols

A SAML 2.0 protocol describes how certain SAML elements (including assertions) are packaged within SAML request and response elements, and gives the processing rules that SAML entities like IdP and SP must follow when producing or consuming these elements. For the most part, a SAML protocol is a simple request-response protocol.

It is important to keep in mind that a SAML protocol always refers to what is transmitted, and not how (the latter is determined by the choice of binding).

In the context of this paper, the most interesting SAML protocols are the Authentication Request Protocol, the Artifact Resolution Protocol, and the Single Logout Protocol.

2.4 SAML 2.0 bindings

A SAML 2.0 binding determines how SAML requests and responses map onto standard messaging or communications protocols. In other words, it’s a mapping of a SAML protocol message onto standard messaging formats and/or communications protocols. SAML 2.0 completely separates the binding concept from the underlying profile (see next section).

The SAML 2.0 standard defines several bindings:

[35] OASIS eXtensible Access Control Markup Language (XACML) TC: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

- HTTP Redirect (GET) Binding;
- HTTP POST Binding;
- HTTP Artifact Binding;
- Etc.

2.5 SAML 2.0 Profiles

A SAML 2.0 profile is a concrete manifestation of a defined use case using a particular combination of assertions, protocols, and bindings, assertions. Indeed, it describes in detail how SAML 2.0 assertions, protocols, and bindings combine to support the considered use case.

The SAML 2.0 standard defines several profiles:

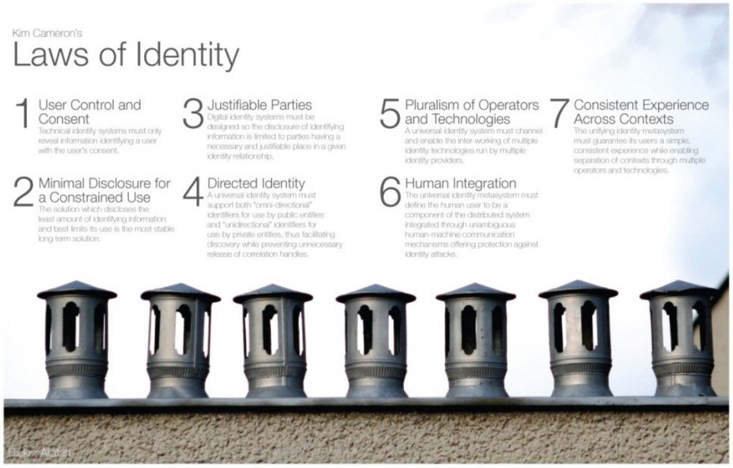
- Web Browser SSO Profile;
- Artifact Resolution Profile;
- Single Logout Profile;
- Identity Provider Discovery Profile;
- Etc.

The most important one is certainly the Web Browser SSO Profile since this is the primary SAML use case for Web SSO and federation. The exchange begins with a request to the SP side. This modern approach referred as SP-initiated provides greater flexibility but rises the so-called Identity Provider Discovery problem in the SAML 2.0 jargon, as a reference to the eponym profile. This is also referred to as the Home Realm Discovery (HRD) and the Where Are You From (WAYF) issues.

> **Note:**
>
> *Such an issue can be solved by the notion of information card and the use of an identity selector as described in another OASIS standard: Identity Metasystem Interoperability governed by the OASIS Identity Metasystem Interoperability (IMI) TC[36]. The notion of Identity Metasystem results from the large initiative launched 4 years ago by Kim Cameron, Digital Identity Chief Architect at Microsoft, through his weblog http://www.identityblog.com in order to improve both the security and the interoperability of online identities. The broad discussions that led to build a consensus around 7 laws of identity as discussed and presented in the white paper THE LAWS OF IDENTITY[37].*

[36] OASIS Identity Metasystem Interoperability (IMI) TC: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=imi

[37] THE LAWS OF IDENTITY: http://www.identityblog.com/?p=354



Kim Cameron's

Laws of Identity

1 User Control and Consent
Technical identity systems must only reveal information identifying a user with the user's consent.

2 Minimal Disclosure for a Constrained Use
The solution which discloses the least amount of identifying information and best limits its use is the most stable long term solution.

3 Justifiable Parties
Digital identity systems must be designed so the disclosure of identifying information is limited to parties having a necessary and justifiable place in a given identity relationship.

4 Directed Identity
A universal identity system must support both "omni-directional" identifiers for use by public entities and "unidirectional" identifiers for use by private entities, thus facilitating discovery while preventing unnecessary release of correlation handles.

5 Pluralism of Operators and Technologies
A universal identity system must channel and enable the inter-working of multiple identity technologies run by multiple identity providers.

6 Human Integration
The universal identity metasystem must define the human user to be a component of the distributed system integrated through unambiguous human-machine communication mechanisms offering protection against identity attacks.

7 Consistent Experience Across Contexts
The unifying identity metasystem must guarantee its users a simple, consistent experience while enabling separation of contexts through multiple operators and technologies.

Flickr Aliatan

> *Taken together, these laws define a "unified identity meta-system" offering to the Internet foundations of an identity layer it is obviously lacking.*

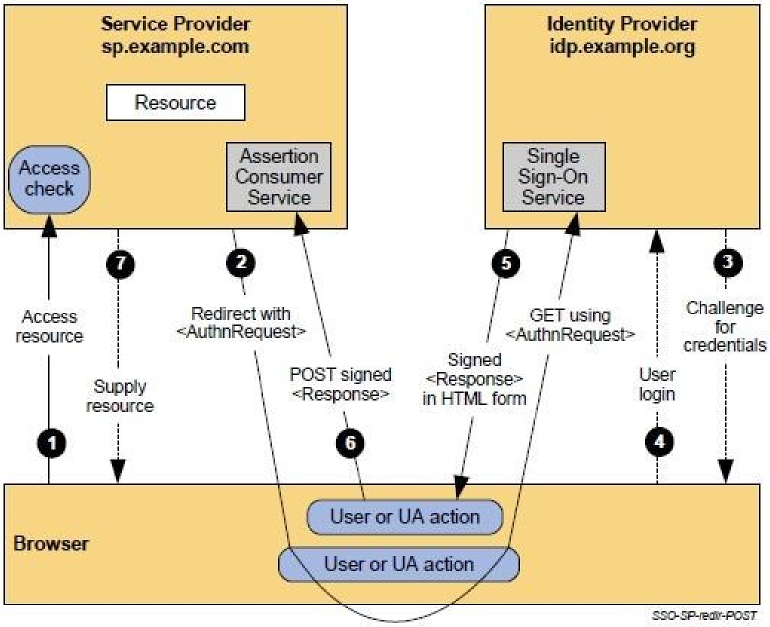
> **Note:**
>
> *Whilst the SP-initiated is a new addition and the intended approach, it should be mentioned that the former IdP-initiated approach as introduced by the previous version of SAML is still supported, but is no longer the preferred one.*

To illustrate the flexibility or simply to number of possible deployments resulting from the possible combinations within a profile, a SP can choose from four bindings (HTTP Redirect, HTTP POST and two flavors of HTTP Artifact), while the IdP has three binding options (HTTP POST plus two forms of HTTP Artifact).

This conducts to a total of 12 possible deployments of the SAML 2.0 Web Browser SSO Profile.

To take on example, the “SP-initiated SSO: Redirect/POST Bindings” describes an SP-initiated SSO exchange where he HTTP Redirect Binding is used to deliver the <*AuthnRequest*> message to the IdP and the HTTP POST Binding is used to return the <*Response*> message containing the assertion to the SP.

The following figure from the SAMLTechOvw non-normative document illustrates the SSO exchange.



2.6 SAML 2.0 Operational Modes

The SAMLConform document provides the technical requirements for SAML 2.0 conformance that software vendors typically care about because it is one measure of cross-product compatibility. It indeed describes features that are mandatory and optional for implementations claiming conformance to SAML 2.0.

For that purposes, several operational modes are defined in this normative document:

- IdP*;
- IdP Lite*;
- SP*;
- SP Lite*;
- Enhanced Client/Proxy (ECP);
- Etc.

*Extended IdP, SP modes also possible.

One should note that, for the same reasons, customers also take care about it as well when they want to assess a specific vendor product's capabilities regarding the SAML 2.0 standard. In other words, if you need to make a formal reference to SAML 2.0 from another document, you simply need to point to this one.

Meanwhile, the previously mentioned certification program now ran by the Kantara Initiative is also a key point to assess the real capabilities of a specific vendor product to truly operate as expected.

3 A brief overview of Active Directory Federation Services (AD FS) 2.0

Microsoft Active Directory Federation Services (AD FS) 2.0 is a component of the Windows (Server) platform and, as such, the right to use it is included in the associated license costs.

AD FS provides final users with a rich SSO experience (on the Web among other scenarios) between applications, services, and platforms:

- Within the enterprise;
- Between organizations;
- On the Internet and in the Cloud as the Microsoft Windows Azure platform[38], the Microsoft's Cloud services platform or the Microsoft Office 365[39], the cloud versions of the Microsoft communications and collaboration products with the latest version of the desktop suite for businesses of all sizes.

3.1 A passive/active Security Token Service (STS)

AD FS 2.0 is fundamentally a Security Token Service (STS). Such a service is able to issue, validate and exchanging security tokens like SAML assertion (see section § 2.2 SAML 2.0 ASSERTIONS). For that purpose, AD FS uses Active Directory Domain services (AD DS)/ Active Directory Lightweight Directory Services (AD LDS) as a credential store. AD FS 2.0 can also use attributes coming from Microsoft SQL Server databases, and other data sources.

The concept of exchange induces the processing and transforming capacity of tokens in terms of type of trust, token format, semantics and (values of) claims for "impedance adaptation".

AD FS 2.0 can consequently play the following roles (and participate accordingly in several types of trust schema's topologies):

- A pure Identity Provider Security Token Service (IP-STS) - This is when AD FS 2.0 has no configured Claim Providers, except a credential store and optional attribute store(s).

  The authentication is performed by the IP-STS against the credential store and a security token is issued to the target relying party so that access control decisions can be made or derived on that basis;
- A pure Relying Party STS (RP-STS) - This is when AD FS 2.0 has configured Claims Providers, but all local authentication methods are disabled in the configuration. AD FS 2.0 can only direct the user to authenticate with a trusted STS/IdP.

  The RP-STS checks the security token presented by the requestors and generates in turn a security token to the target resource or the next relying party in the chain to the target resource. In the former case, it can issue a delegation token (Act As tokens) in order to support delegation scenarios;

[38] Microsoft Windows Azure platform: http://www.windowsazure.com/

[39] Microsoft Office 365: http://office365.microsoft.com/

- Hybrid - This is when AD FS 2.0 has configured Claims Providers, and uses a local authentication method enabled in the configuration.

3.2 Federation in heterogeneous environments

To adapt to an open set of federation scenarios, it supports multiple OASIS standards: WS-Federation, SAML 2.0, WS-Trust, and IMI.

Indeed, similar to the previous versions 1.x, AD FS 2.0 supports the WS-Fed Passive protocol[40] (OASIS WS-Federation standard) for browser-based passive clients. It uses the SAML assertion format for security tokens, but as its name suggest, not the protocol.

This protocol is adopted by most 3rd party IDA vendors. Consequently, having AD FS 2.0 supporting WS-Fed Passive protocol potentially allows interoperability with major market solutions like:

- BMC Universal Identity Federator ;
- CA eTrust SiteMinder Federation Security Services (6 SP5) ;
- IBM Tivoli Federated Identity Manager ;
- Internet2 Shibboleth System (1.3) (avec extension) / Internet2 Shibboleth System[41];
- Novell Access Manager ;
- Oracle Identity Federation ;
- Ping Identity PingFederate Server ;
- RSA Federated Identity Manager ;
- Sun OpenSSO ;
- symLABS Federated Identity Suite ;
- Version3 Enhanced Authentication Edition.

AD FS 2.0 adds to this the support the SAML 2.0 protocol and furthermore, natively offers the ability of a protocol gateway by acting as a gateway between SAML 2.0 and WS-Fed Passive protocols for front-channel federation.

This helps expose very simply in this context an application like Microsoft SharePoint 2007/2010 with SAML 2.0-based federation. In practice, part of the conversation between SharePoint 2010 and AD FS 2.0 (in both ways) happens under WS-Fed Passive protocol, while the other part between AD FS 2.0 and a SAML 2.0-based IdP (in both ways) happens under SAML 2.0 protocol.

Both SAML 2.0 protocol support and the ability to bridge protocols were greeted by Scott Cantor:

> "*As a Shibboleth and OpenSAML project developer, and a deployer of the Shibboleth software at The Ohio State University, I'm excited and gratified that Microsoft is implementing the SAML 2.0 Web SSO profile in its upcoming products. Throughout the life of the Shibboleth project, and my work on the SAML 2.0 standard, our goal has been to leverage open standards to foster broad interoperability in federated identity within the higher education community and between it and its many commercial and non-commercial partners. Microsoft is clearly one of those critical partners, and as a key technology supplier, its support for the SAML standard reflects an understanding of our community's needs and goals, and will expand the scope and impact of our efforts.*

[40] WEB SERVICES FEDERATION LANGUAGE (WS-FEDERATION) VERSION 1.2 : http://docs.oasis-open.org/wsfed/federation/v1.2/ws-federation.pdf

[41] While fully implemented/supported, the support of the WS-Fed Passive protocol isn't widely used with Shibboleth-based communities.

> *Our users will benefit by obtaining access to the broadest potential set of federated applications and services, and our worldwide community will benefit from the opportunity to deploy Microsoft's identity solutions with the knowledge that they will interoperate with Shibboleth. Microsoft's willingness to listen to our requirements and suggestions demonstrates a commitment to real-world compatibility. I look forward to continuing the dialog with Microsoft as we drive further interoperability in the use of federation metadata to scale and simplify both SAML 2.0 and WS-Federation deployments.*"

Vis-à-vis SAML 2.0 protocol, to be more specific, AD FS 2.0 supports:

- The SAML 2.0 IdP Lite and SP Lite operation modes as described in the Liberty Alliance/Kantara Initiative interoperable program. There is a slight difference between this description and the OASIS eponym operational modes (see section § 2.6 SAML 2.0 OPERATIONAL MODES)
- As well as the eGov SAML 2.0 Profile v1.5, first of many vertical-specific constraining profiles (General Service Administration) version 1.5 (see LIBERTY ALLIANCE EGOVERNMENT PROFILE FOR SAML 2.0[42]).

Which our customers told us were important to them.

> **Note:**
>
> *IdP and SP Lite modes cover indeed the essential federation capabilities.*

The table hereafter presents the SAML 2.0 functions matrix for IdP Lite and SP lite operational modes supported by AD FS 2.0.

Functions	IdP Lite	SP Lite	ADFS 2.0
Web SSO, <*AuthnRequest*>, HTTP Redirect	MUST	MUST	X
Web SSO, <*Response*>, HTTP POST	MUST	MUST	X
Web SSO, <*Response*>, HTTP Artifact	MUST	MUST	X
Artifact Resolution, SOAP	MUST	MUST	X
Name Identifier Management, HTTP Redirect (IdP-initiated)	MUST NOT	MUST NOT	
Name Identifier Management, SOAP (IdP-initiated)	MUST NOT	MUST NOT	
Name Identifier Management, HTTP Redirect	MUST NOT	MUST NOT	
Name Identifier Management, SOAP (SP-initiated)	MUST NOT	MUST NOT	
Single Logout (IdP-initiated) - HTTP Redirect	MUST	MUST	X
Single Logout (IdP-initiated) - SOAP	OPTIONAL	OPTIONAL	
Single Logout (SP-initiated) - HTTP Redirect	MUST	MUST	X
Single Logout (SP-initiated) - SOAP	OPTIONAL	OPTIONAL	
Identity Provider Discovery (cookie)	OPTIONAL	OPTIONAL	X (IdP & SP)

[42] LIBERTY ALLIANCE EGOVERNMENT PROFILE FOR SAML 2.0: http://www.projectliberty.org/liberty/content/download/4711/32210/file/Liberty_Alliance_eGov_Profile_1.5_Final.pdf

Note:

In order to have the HTTP Artifact Binding available in AD FS 2.0, AD FS 2.0 should be configured to use a Microsoft SQL Server configuration database. The Windows Internal Database (WID), a variant a variant of SQL Server Express included with Windows Server 2008 R2 cannot be used in this context.

AD FS 2.0 exposes the following endpoint URLs for the SAML 2.0 protocol support:

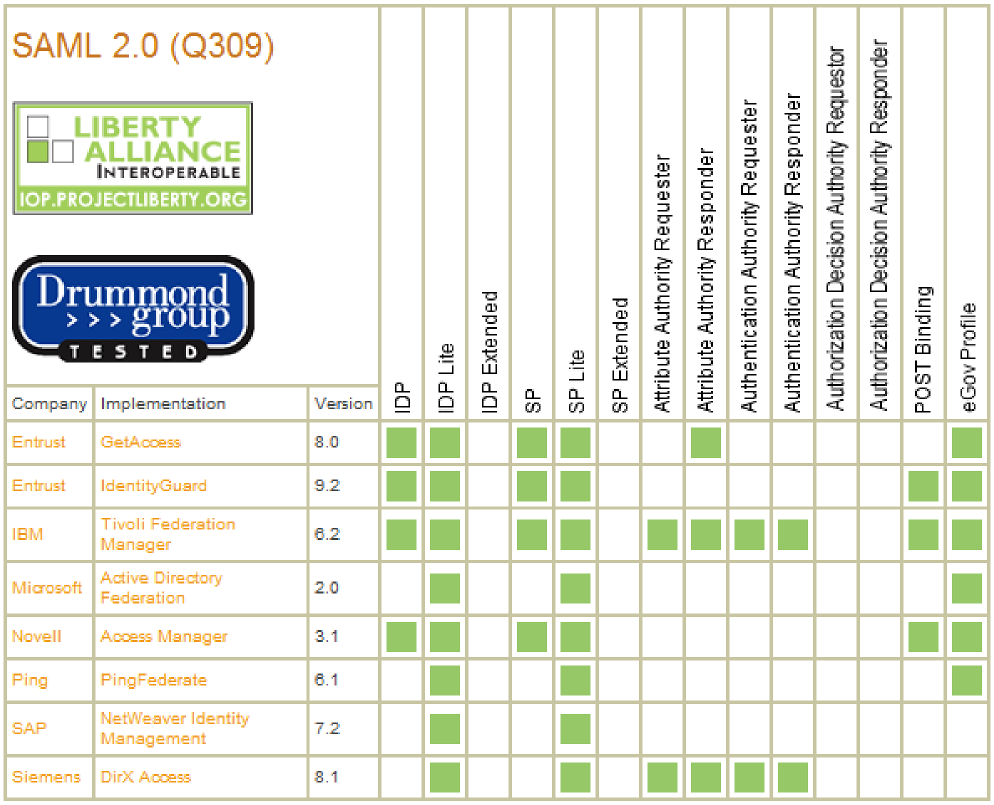
Description	URL
SAML 2.0 Web SSO (IDP-initiated)	/adfs/ls/**IdpInitiatedSignOn.aspx**
SAML 2.0 Web SSO (SP-initiated)	/adfs/ls**/**
SAML 2.0 Artifact Resolution	/adfs/services/trust/**artifactresolution**
Federation Medata	/FederationMetadata/2007-06/**Federationmetadata**.xml

AD FS 2.0 successfully passed the SAML 2.0 interoperability tests for these modes as described in the document LIBERTY INTEROPERABILITY TESTING PROCEDURES FOR SAML 2.0 VERSION 3.2.2[43] .

Note:

One should note that there are some slight differences between the Liberty Alliance SAML 2.0 IdP Lite and SP Lite operation modes and the OASIS eponym operational modes (see section § 2.6 SAML 2.0 OPERATIONAL MODES). Indeed, Enhanced client/proxy (ECP) support is required in OASIS IdP & SP Lite criteria, but it is not required in Liberty IdP & SP Lite criteria. AD FS 2.0 has Liberty certification for IDP & SP Lite

[43] LIBERTY INTEROPERABILITY TESTING PROCEDURES FOR SAML 2.0 VERSION 3.2.2:
http://www.projectliberty.org/liberty/content/download/4709/32204/file/Liberty_Interoperability_SAML_Test_Plan_v3.2.2%20.pdf



SAML 2.0 (Q309)

LIBERTY ALLIANCE INTEROPERABLE
IOP.PROJECTLIBERTY.ORG

Drummond group
TESTED

Company	Implementation	Version	IDP	IDP Lite	IDP Extended	SP	SP Lite	SP Extended	Attribute Authority Requester	Attribute Authority Responder	Authentication Authority Requester	Authentication Authority Responder	Authorization Decision Authority Requestor	Authorization Decision Authority Responder	POST Binding	eGov Profile
Entrust	GetAccess	8.0	🟩	🟩		🟩	🟩			🟩						🟩
Entrust	IdentityGuard	9.2	🟩	🟩		🟩	🟩								🟩	🟩
IBM	Tivoli Federation Manager	6.2	🟩	🟩		🟩	🟩		🟩	🟩	🟩	🟩			🟩	🟩
Microsoft	Active Directory Federation	2.0		🟩			🟩									🟩
Novell	Access Manager	3.1	🟩	🟩		🟩	🟩								🟩	🟩
Ping	PingFederate	6.1		🟩			🟩									🟩
SAP	NetWeaver Identity Management	7.2		🟩			🟩									
Siemens	DirX Access	8.1		🟩			🟩		🟩	🟩	🟩	🟩				

see Liberty Alliance press release ENTRUST, IBM, MICROSOFT, NOVELL, PING IDENTITY, SAP AND SIEMENS PASS LIBERTY ALLIANCE SAML 2.0 INTEROPERABILITY TESTING[44]

This capability of AD FS 2.0 is a consequence of the major announcement[45] that was made by Microsoft on February 2008 about the enhancements of its products openness, interoperability, and the creation of new opportunities for developers, partners, customers and competitors.

Exchanging information between people and organizations, interoperability between applications and services have become first-class needs. Microsoft committed to interoperability a while ago, after having been exchanging with their customers about their interoperability needs and listening to them on how Microsoft products should become even more open and interoperable.

In order to fulfill those stakes and needs, Microsoft applies four interoperability principles to their own broadly used products like Windows Server, SharePoint, etc. from now on:

1. Guarantee an open connection to these products;
2. Promote data portability;
3. Enhance industry standards support;
4. Favor exchange and collaboration in the IT industry including with the Open Source communities about interoperability and standards topics.

Of course, these principles apply to AD FS 2.0 which clearly has such goals.

[44] Liberty Alliance press release ENTRUST, IBM, MICROSOFT, NOVELL, PING IDENTITY, SAP AND SIEMENS PASS LIBERTY ALLIANCE SAML 2.0 INTEROPERABILITY TESTING: http://www.projectliberty.org/liberty/news_events/press_releases/entrust_ibm_microsoft_novell_ping_identity_sap_and_siemens_pass_liberty_alliance_saml_2_0_interoperability_testing

[45] News Press Release. Microsoft Makes Strategic Changes in Technology and Business Practices to Expand Interoperability: http://www.microsoft.com/presspass/press/2008/feb08/02-21ExpandInteroperabilityPR.mspx

Beyond mostly browser-based protocols like the SAML 2.0 and WS-Fed Passive protocols, AD FS also supports for smart clients the OASIS WS-Trust standard. This standard is governed by the OASIS Web Services Secure Exchange (WS-SX) TC[46].

All these capacities are recognized by the market. Indeed, on the occasion of the European Identity Conference (EIC) 2009, the leading European event for Identity and Access Management (IAM) and GRC (Governance, Risk Management, and Compliance), the analyst firm Kuppinger Cole conferred the European Identity Award 2009[47], in the category "Best innovation", to Microsoft for the Geneva project (AD FS 2.0 & WIF 1.0), in which federation becomes part of user containers, "*one of the most significant enhancements for future use and dissemination of the Identity Federation*".

On the occasion of the European Identity Conference 2010 (EIC), Kuppinger Cole's conferred the European Identity Award 2010[48], in the category "Best Project B2C", to the University of Washington (UW). UW was honored for its identity federation solution based on both AD FS 2.0 & WIF 1.0 in research and education which was developed together with Microsoft and is intended to form part of the "Live@Edu" initiative.

"*The University of Washington is delighted to have its work with Microsoft on federation services honored by Kuppinger Cole*", said RL "Bob" Morgan, Identity Architect for UW Information Technology and Shibboleth Project core team member. "*At UW, we are committed to standards-based federation to extend the value of UW identity to the services our users need. It is great to partner with Microsoft since they too are making a commitment to federation for Windows Live and Live@edu. Live@edu's support of higher-education federations including InCommon is a key differentiator. Making it all work has many challenges, but it's essential so the higher-ed community can collaborate seamlessly and securely in cloud environments."*

Nathan Dors, manager of Identity and Access Management for UW Information Technology, added that: "*we agree with Microsoft on the importance of being both standards-oriented and pragmatic. Choice of federating technology is key and we appreciate Microsoft's striving to reach parity between AD FS 2.0 and Shibboleth solutions*".

[46] OASIS Web Services Secure Exchange (WS-SX) TC: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-sx

[47] European Identity Award 2009: http://www.id-conf.com/blog/2009/05/07/awards-for-outstanding-identity-management-projects/

[48] European Identity Award 2010: http://www.id-conf.com/blog/2010/05/05/outstanding-projects-and-initiatives-in-im-honored/

4 The eleven interoperability “Gotchas” you should be aware of

Federation events typically have a short Time to Live (TTL). Therefore, it is vitally important to ensure that both computers have their clocks synchronized in order to avoid errors based on time-outs.

> **Note:**
>
> *For information about how to synchronize a Windows Server 2008 R2 domain controller to an Internet time server, see article 816042 How to configure an authoritative time server in Windows Server[49].*

Beyond a basic but typical issue like this one, this chapter covers the 11 interoperability “gotchas” you should be aware of when deploying AD FS 2.0 in a SAML 2.0 world.

4.1 Encryption

The SAML 2.0 protocol enables advanced use of PKI for federation security that is outside the scope of this document. Capabilities notably include:

- Encryption of SAML 2.0 authnrequests sent by the SP to the IdP;
- Encryption of SAML 2.0 security tokens and assertion data.

For additional information, please refer to the normative SAMLCore and SAMLSec documents.

AD FS 2.0 automatically configures itself to encrypt token data whenever it receives an encryption certificate from a partner/partner metadata includes an encryption certificate.

When performing encryption, which covers claims encryption and Name ID encryption in logout request, AD FS 2.0 defaults to using 256-bit Advanced Encryption Standard (AES) keys, or AES-256

In contrast, Java-based solutions support only AES-128 by default. There is no knob to turn down AD FS 2.0 encryption strength from the UI.

You can opt to one of the 2 following solutions:

1. Upgrade Java encryption strength with the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6[50];
2. Turn off encryption in AD FS 2.0 using the Windows PowerShell command-line and scripting environment.

```
Add-PSSnapin Microsoft.adfs.powershell

set-ADFSRelyingPartyTrust –TargetName foo –EncryptClaims $False
```

[49] Article 816042 How to configure an authoritative time server in Windows Server: http://go.microsoft.com/fwlink/?LinkID=60402

[50] Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6: http://www.oracle.com/technetwork/java/javase/downloads/index.html

Note:

Windows PowerShell is a command-line shell and scripting language that is designed for system administration and Automation. It uses administrative tasks called cmdlets. Each cmdlet has required and optional arguments, called parameters, that identify which objects to act on or control how the cmdlet performs its task. You can combine cmdlets in scripts to perform complex functions that give you more control and help you automate the administration of Windows and applications. It has become a common way to manage the latest generation of Microsoft Server products, including Windows Server 2008 (R2), etc.

For more information about Windows PowerShell 2.0, please see the Windows PowerShell Web site[51], the Windows PowerShell online help[52], and the Windows PowerShell Weblog[53] Windows PowerShell Software Development Kit (SDK)[54] that includes a programmer's guide along with a full reference.

For more information on the AD FS 2.0 cmdlets, see the AD FS 2.0 ADMINISTRATION WITH WINDOWS POWERSHELL[55] section of the AD FS 2.0 OPERATIONS GUIDE and the AD FS 2.0 CMDLETS REFERENCE [56].

4.2 Signing

The SAML 2.0 protocol enables advanced use of PKI for federation security that is outside the scope of this document. Capabilities notably include:

- Digital signing of SAML 2.0 HTTP Artifact Profile artifact requests;
- Digital signing of SAML 2.0 logout requests and responses.

For additional information, please refer to the normative SAMLCore and SAMLSec documents.

AD FS 2.0 signs using the SHA-256 algorithm by default. This is relevant when signing as IdP (assertions, responses, logout requests) and as SP (authnrequests, logout responses, artifact GETs)

Furthermore, AD FS 2.0 expects all partners, by default, to also sign using SHA-256 while most partner solutions currently sign using SHA-1, consequently creating errors.

To fix this situation, the solution consists in changing AD FS 2.0 Claims Provider/Relying Party entries to expect SHA-1 instead.

[51] Windows PowerShell Web site: http://www.microsoft.com/powershell

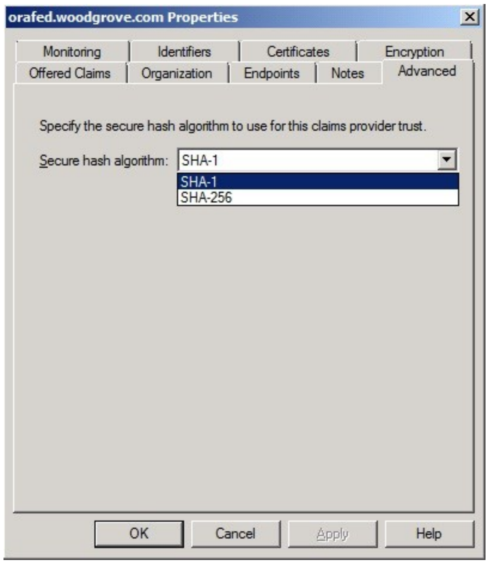
[52] Windows PowerShell online help: http://technet.microsoft.com/en-us/library/bb978526.aspx

[53] Windows PowerShell Weblog: http://blogs.msdn.com/powershell

[54] Windows PowerShell SDK: http://msdn2.microsoft.com/en-us/library/aa830112.aspx

[55] AD FS 2.0 ADMINISTRATION WITH WINDOWS POWERSHELL: http://go.microsoft.com/fwlink/?LinkId=194005

[56] AD FS 2.0 CMDLETS REFERENCE: http://go.microsoft.com/fwlink/?LinkId=177389).


orafed.woodgrove.com Properties

Specify the secure hash algorithm to use for this claims provider trust.

Secure hash algorithm: SHA-1

SHA-1
SHA-256

4.3 CRL checking

The primary benefit of using certificates issued from a certification authority (CA) is the ability to check for possible certificate revocation against the certificate revocation list (CRL) from the issuing CA when acting as a SP.

AD FS 2.0 default is to check CRL in partner signing & encryption certificates. In other words, CRL checking is enabled by default for all Claims Provider trusts.

This has obvious implications in federation deployments between a SAML 2.0 environment (acting as an IdP) and AD FS 2.0 (acting as an RP):

- CA-issued certificates must have a valid CDP extension configured - If the signing private key used by SAML 2.0 environment includes a CRL Distribution Point (CDP) extension that location must be accessible by the AD FS 2.0 Federation Server, or CRL checking fails, resulting in a failed access attempt.

  CDP extensions are added by default to certificates that are issued by Active Directory Certificate Services (AD CS) in Windows Server 2008 R2.

- No CDP extension means no CRL checking - If the signing private key does not include a CDP extension, no CRL checking is performed by AD FS 2.0.

As Laura E. Hunter, a Principal Technology Architect at MS IT Identity & Management, says that “*if federation is broken, it's PKI. If it is not PKI, there's a typo. If you typed it correctly (case counts!). It's PKI*”. (Federation metadata support is a mean to get rid of typos.)

You can turn off CRL checking for a specific Claims Provider trust by using the Windows PowerShell command-line and scripting environment as follow:

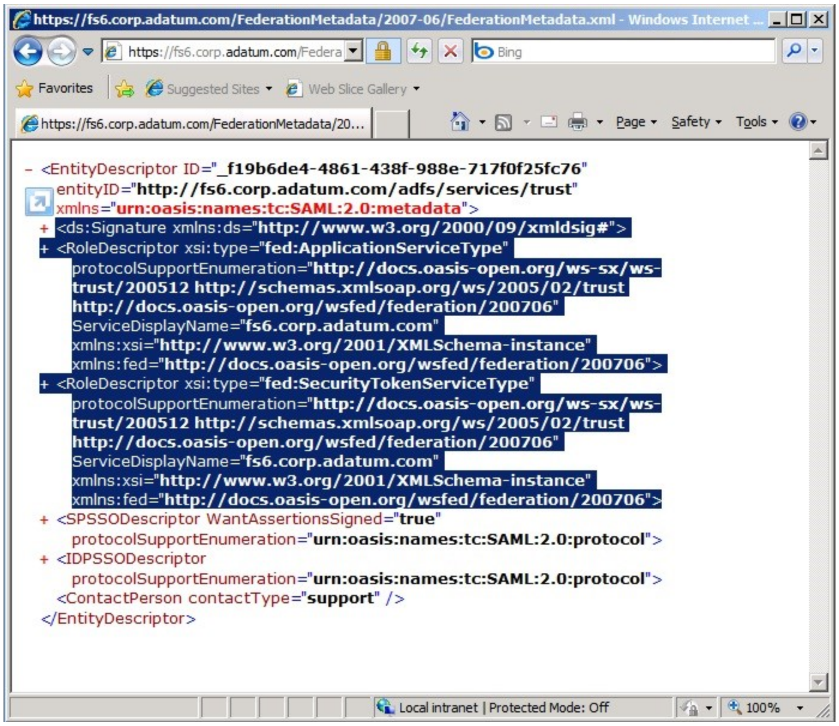
```
Add-PSSnapin Microsoft.adfs.powershell

set-ADFSClaimsProviderTrust –TargetName foo –SigningCertificateRevocationCheck None
set-ADFSClaimsProviderTrust –TargetName foo –EncryptionCertificateRevocationCheck None
```

4.4 Metadata handling

Adding a partner using AD FS 2.0 into a 3rd party IDA solution can be done either manually or through metadata import.

The auto-generated AD FS 2.0 metadata file Federationetadata.xml includes information about performing both the IdP and SP roles, including the public key which will be used to validate security tokens signed by AD FS 2.0 in the identity provider role.

In terms of standards, it conforms to the OASIS WS-Federation metadata, which uses extension points in the OASIS SAML 2.0 metadata standard (see SAMLMeta document) in order to include WS-Trust content.


https://fs6.corp.adatum.com/FederationMetadata/2007-06/FederationMetadata.xml - Windows Internet ...
https://fs6.corp.adatum.com/Federa
Bing
Favorites
Suggested Sites
Web Slice Gallery
https://fs6.corp.adatum.com/FederationMetadata/20...
Page
Safety
Tools
Local intranet | Protected Mode: Off
100%

```
- <EntityDescriptor ID="_f19b6de4-4861-438f-988e-717f0f25fc76"
    entityID="http://fs6.corp.adatum.com/adfs/services/trust"
    xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  + <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  + <RoleDescriptor xsi:type="fed:ApplicationServiceType"
      protocolSupportEnumeration="http://docs.oasis-open.org/ws-sx/ws-
      trust/200512 http://schemas.xmlsoap.org/ws/2005/02/trust
      http://docs.oasis-open.org/wsfed/federation/200706"
      ServiceDisplayName="fs6.corp.adatum.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:fed="http://docs.oasis-open.org/wsfed/federation/200706">
  + <RoleDescriptor xsi:type="fed:SecurityTokenServiceType"
      protocolSupportEnumeration="http://docs.oasis-open.org/ws-sx/ws-
      trust/200512 http://schemas.xmlsoap.org/ws/2005/02/trust
      http://docs.oasis-open.org/wsfed/federation/200706"
      ServiceDisplayName="fs6.corp.adatum.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:fed="http://docs.oasis-open.org/wsfed/federation/200706">
  + <SPSSODescriptor WantAssertionsSigned="true"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  + <IDPSSODescriptor
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <ContactPerson contactType="support" />
  </EntityDescriptor>
```

When the metadata file import is supported to add a partner and consequently create a trust relationship, many IDA solutions choke on that content, and fail to import the metadata file.

This solution consist in taking out the WS-Trust metadata content, and the signature, and then most import processes will succeed.

▷ To remove the WS-Trust metadata content and the metadata signature:

1. Go to the AD FS 2.0 metadata XML file at https://<your-ADFS-sts>/FederationMetadata/2007-06/FederationMetadata.xml using Internet Explorer, Internet Explorer.
2. Click Page, and then click Save As to save *FederationMetadata.xml* to the desktop.
3. Open FederationMetadata.xml with an XML editor.
4. Delete the sections of the file shown in the following table.

Description	Section starts with…	Section ends with…
Metadata document signature	*<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">*	*</ds:Signature>*
WS-Trust & WS-Federation application service metadata	*<RoleDescriptor xsi:type="fed:ApplicationServiceType"*	*</RoleDescriptor>*
WS-Trust & WS-Federation security token service metadata	*<RoleDescriptor xsi:type="fed:SecurityTokenServiceType"*	*</RoleDescriptor>*

5. Save the edited file as *FederationMetadata_4SAML2.xml*, and then close it.

4.4.1 AD FS 2.0 as the IdP

As previously mentioned, the auto-generated AD FS 2.0 metadata file Federationetadata.xml includes information about performing both the IdP and SP roles. This is potentially the case for *FederationMetadata_4SAML2.xml* (based upon the AD FS 2.0 configuration).

Not every 3rd party solution supports having both SAML 2.0 IdP and SP descriptors in the metadata file when trying to add an IdP on that basis.

▷ To add the AD FS 2.0 partner and consequently create the remote IdP entity using the amended metadata:

1. Open *FederationMetadata_4SAML2.xml* from the desktop using an XML editor.
2. Delete the following section of the file.

Description	Section starts with…	Section ends with…
SAML 2.0 SP metadata	*<SPSSODescriptor WantAssertionsSigned="true"*	*</SPSSODescriptor>*

The first two elements of the resulting file should look like:

```
<EntityDescriptor ID=…>
   <IDPSSODescriptor WantAssertionsSigned="true"…
```

3. Save the file to the desktop as *FederationMetadata_idp.xml* and close it.
4. Use *FederationMetadata_idp.xml* accordingly in the SP 3rd party solution to declare the AD FS 2.0 IdP.

4.4.2 AD FS 2.0 as the SP

As before, you can now use the amended version of the AD FS 2.0 Federationetadata.xml file to create the remote AD FS 2.0 SP entity in the IdP 3rd party solution.

As previously mentioned, the auto-generated AD FS 2.0 metadata file Federationetadata.xml includes information about performing both the IdP and SP roles. This is potentially the case for *FederationMetadata_4SAML2.xml* (based upon the AD FS 2.0 configuration).

Not every 3rd party solution supports having both SAML 2.0 IdP and SP descriptors in the metadata file when trying to add a SP on that basis.

▷ To add the AD FS 2.0 partner and consequently create the remote SP entity using the amended metadata:

1. Open *FederationMetadata_4SAML2.xml* from the desktop using an XML editor.
2. Delete the following section of the file.

Description	Section starts with…	Section ends with…
SAML 2.0 IdP metadata	*<IDPSSODescriptor WantAssertionsSigned="true"*	*</IDPSSODescriptor>*

The first two elements of the resulting file should look like:

```
<EntityDescriptor ID=…>
   <SPSSODescriptor WantAssertionsSigned="true"…
```

3. Save the file to the desktop as *FederationMetadata_sp.xml* and close it.
4. Use *FederationMetadata_sp.xml* accordingly in the SP 3rd party solution to declare the AD FS 2.0 SP.

4.5 Name ID formats

For AD FS 2.0 the name identifier (Name ID) is yet another claim but you need to generate name identifiers to use the SAML 2.0 protocol:

- Name identifier is the default attribute for user lookup in most 3rd party solutions;
- Name identifier is necessary if you plan to take advantage of SAML 2.0 Single Logout Protocol.

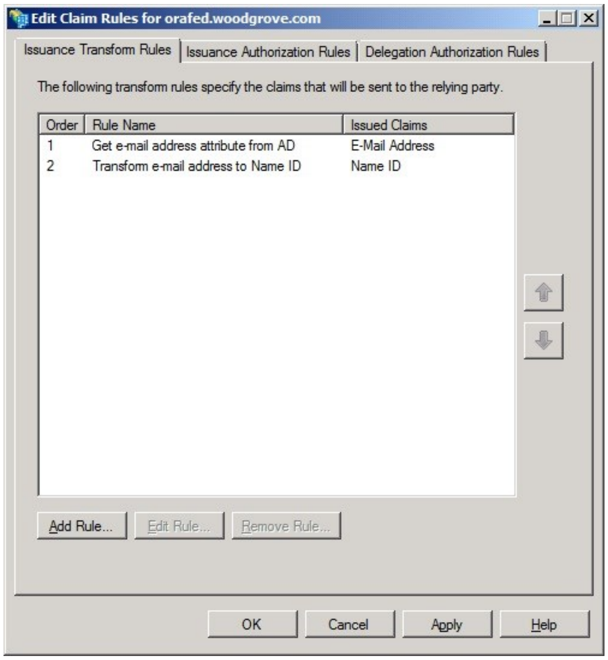
> **Note:**
>
> *You may also want to generate name identifiers if you plan to federate with non-AD FS 2.0 deployment regardless of the protocol being used for the federation.*

In the IdP role, AD FS 2.0 sends Name ID claim without a name identifier format while some products expect a format.

Formats include Persistent Identifier, Transient Identifier, E-mail Address, etc.

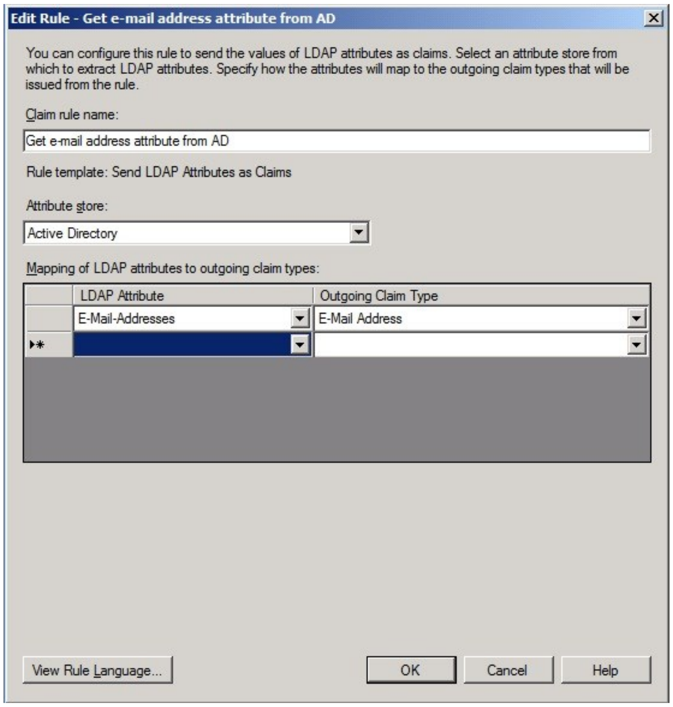
The solution to fix this is a 2-step approach in the AD FS 2.0 UI:

1. Pull attribute into arbitrary claim type. You must use something other than the "Name" claim type. In the screenshot hereafter, we use the E-Mail Address claim type.


Edit Claim Rules for orafed.woodgrove.com
Issuance Transform Rules
Issuance Authorization Rules
Delegation Authorization Rules
Add Rule...
Edit Rule...
Remove Rule...
OK
Cancel
Apply
Help

The following transform rules specify the claims that will be sent to the relying party.

Order	Rule Name	Issued Claims
1	Get e-mail address attribute from AD	E-Mail Address
2	Transform e-mail address to Name ID	Name ID


Edit Rule - Get e-mail address attribute from AD

You can configure this rule to send the values of LDAP attributes as claims. Select an attribute store from which to extract LDAP attributes. Specify how the attributes will map to the outgoing claim types that will be issued from the rule.

Claim rule name: Get e-mail address attribute from AD

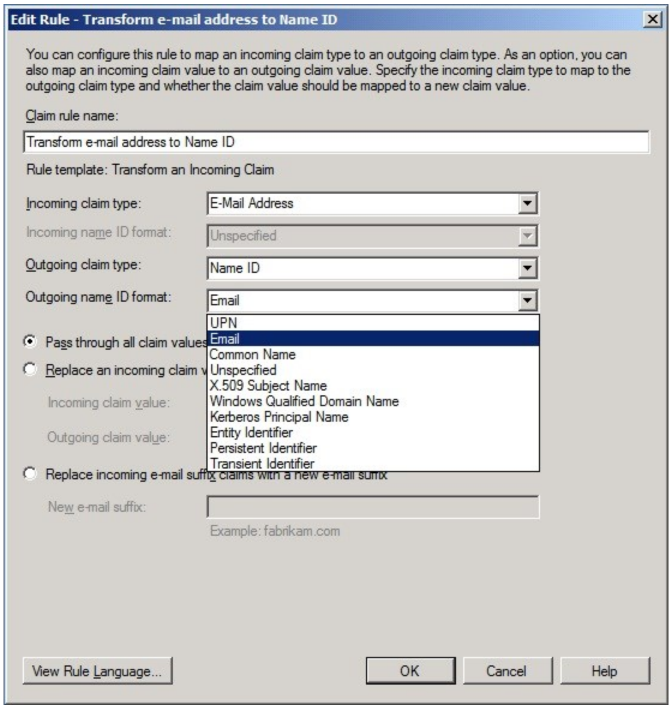
Rule template: Send LDAP Attributes as Claims

Attribute store: Active Directory

Mapping of LDAP attributes to outgoing claim types:

LDAP Attribute	Outgoing Claim Type
E-Mail-Addresses	E-Mail Address

2. Transform into Name ID claim type (outgoing claim type), and apply format there (outgoing name ID format).


Edit Rule - Transform e-mail address to Name ID

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name: Transform e-mail address to Name ID

Rule template: Transform an Incoming Claim

Incoming claim type: E-Mail Address

Incoming name ID format: Unspecified

Outgoing claim type: Name ID

Outgoing name ID format: Email

● Pass through all claim values

○ Replace an incoming claim v

Incoming claim value:

Outgoing claim value:

○ Replace incoming e-mail suffix claims with a new e-mail suffix

New e-mail suffix:

Example: fabrikam.com

4.6 Persistent & transient Name IDs

The concept is to use opaque, alphanumeric string to represent a user instead of readable and understandable value (e.g. e-mail address) for name identifier (Name ID). Opaque identifiers come into 2 flavors in order to address 2 different specific use cases.

- **Persistent name identifier**, which uses the same value for each access request per user.

  Persistent identifier is meant to obfuscate the real user identity, so it’s not possible to link user activities across different relying parties. At the same time, the IdP guarantees that persistent id will remain the same each time same the user logs in again.

  In a SAML 2.0 assertion, it may look similar to:

```
<Assertion ID=”_90bd669e-4a85-412d-9969-90e43e031fac” IssueInstant=”2010-12-01T02:50:48.719Z" Version="2.0"
           xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
  <Issuer>http://mysts/adfs/services/trust</Issuer>
  …
  <Subject>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameidformat:persistent">
       zbonsm0Yn9Gnw14uQEEPr6AO7d+IvxwCQN3t+o24jYs=
    </NameID>
    …
  </Subject>
  …
</Assertion>
```

  The associated use case is the account linking, because persistent name identifiers can be appended to an application-side user account and then used like any other attribute for user disambiguation.

- **Transient name identifier**, which uses a unique value for each access request.

  Transient identifier has similar properties but it’s only valid for single login session (i.e. it will be different each time the user authenticates again, but will stay the same as long as the user is authenticated).

  The associated use case is the pseudo-anonymous access. Transient name identifiers are useful in cases in which a user identity is not needed at the application, only confidence that the user successfully authenticated at a trusted relying party, but an ID that tracks back to a specific user is needed for repudiation and similar purposes.

4.6.1 AD FS 2.0 as an IdP

In the IdP role, AD FS 2.0 supports the formats, but generating opaque values requires some customization by setting up an appropriate Relying Party Trust issuance policy to create a persistent identifier or a transient identifier in the SAML assertion.

> **Note:**
>
> *For more information, please refer to the blog post NAME IDENTIFIERS IN SAML ASSERTIONS[57].*

As before, for AD FS 2.0 the name identifier (Name ID) is yet another claim.

We assume that you already configured sample Relying Party Trust with basic policy. In case you don’t, here is some recommended reading:

- CONFIGURING RELYING PARTIES[58];
- AD FS 2.0 POLICY LANGUAGE[59].

[57] NAME IDENTIFIERS IN SAML ASSERTIONS: http://blogs.msdn.com/b/card/archive/2010/02/17/name-identifiers-in-saml-assertions.aspx

[58] CONFIGURING RELYING PARTIES: http://technet.microsoft.com/en-us/library/dd807074(WS.10).aspx

[59] AD FS 2.0 POLICY LANGUAGE: http://technet.microsoft.com/en-us/library/dd807118(WS.10).aspx

To create a persistent name identifier:

1. Choose your Relying Party Trust and make a custom issuance transform rule to create unique user identifier claim.

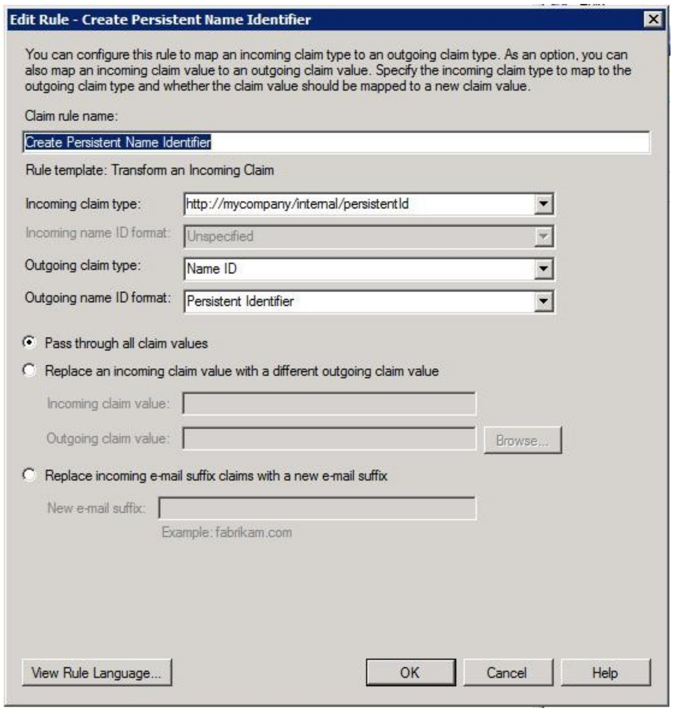
   For the first rule we have to create advanced rule that will use custom built-in store for generating opaque identifiers. Sample rule below will use Windows Account Name Claim as a seed to generate unique identifier that will persist across all sessions.

```
c:[type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname" ]
=> add(store  = "_OpaqueIdStore",
       types = ("http://mycompany/internal/persistentId"),
       query = "{0};{1};{2}",
       param = "ppid",
       param = c.Value,
       param = c.OriginalIssuer);
```

> **Note:**
>
> *The rule that we are using is an attribute extraction rule from one of the built-in attribute stores. The store takes 3 parameters: mode ("ppid") and 2 parameters that are used as a seed for generating pseudo-random identifier. The result is also mixed with AD FS 2.0 installation specific secret entropy.*

2. Transform persistent identifier claim into Name Identifier claim.

   We can use built-in rule to transform the persistent identifier claim created in step 1 into name identifier claim.



Edit Rule - Create Persistent Name Identifier

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name:
Create Persistent Name Identifier

Rule template: Transform an Incoming Claim

Incoming claim type: http://mycompany/internal/persistentId
Incoming name ID format: Unspecified
Outgoing claim type: Name ID
Outgoing name ID format: Persistent Identifier

● Pass through all claim values
○ Replace an incoming claim value with a different outgoing claim value
Incoming claim value:
Outgoing claim value:
○ Replace incoming e-mail suffix claims with a new e-mail suffix
New e-mail suffix:
Example: fabrikam.com

To create a transient name identifier:

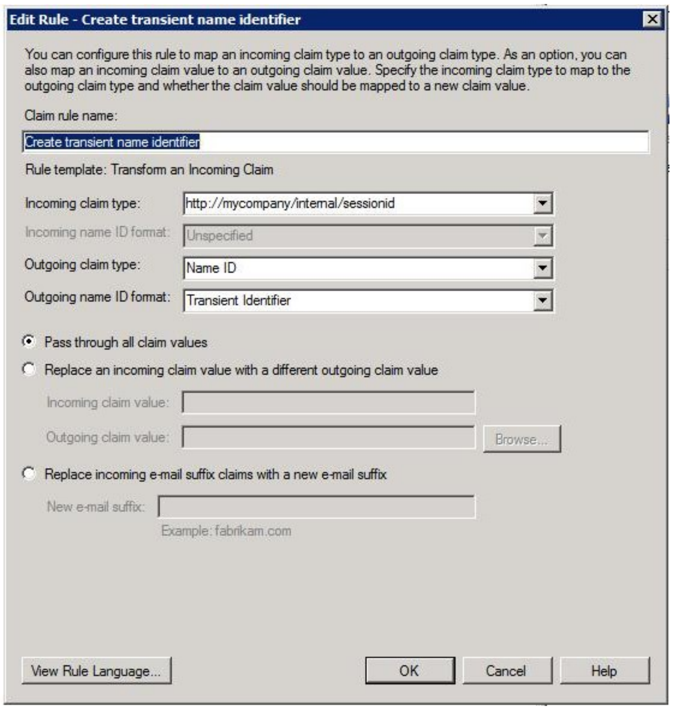
1. Create custom rule to create per session identifier.

   For identifier we will use second mode of opaque store where it takes extra entropy to mix into result. In addition to Windows Account Name, we will also use authentication instant to generate identifier that will persist only for current login session.

```
c1:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"] &&
c2:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant"]
 => add(
       store = "_OpaqueIdStore",
       types = ("http://mycompany/internal/sessionid"),
       query = "{0};{1};{2};{3};{4}",
       param = "useEntropy",
       param = c1.Value,
       param = c1.OriginalIssuer,
       param = "",
       param = c2.Value);
```

> **Note:**
>
> *This time we also extract values from built-in store. This time, the mode is “useEntropy”, which basically means that we will use 2 extra parameters (3rd and 4th) to additionally mix into returned identifier. Parameter 3 is site specific entropy (empty means use relying party identifier). Parameter 4 is any other entropy: we use authentication instant.*

2. Transform temporary claim into Name Identifier claim.



Edit Rule - Create transient name identifier

You can configure this rule to map an incoming claim type to an outgoing claim type. As an option, you can also map an incoming claim value to an outgoing claim value. Specify the incoming claim type to map to the outgoing claim type and whether the claim value should be mapped to a new claim value.

Claim rule name:

Create transient name identifier

Rule template: Transform an Incoming Claim

Incoming claim type: http://mycompany/internal/sessionid

Incoming name ID format: Unspecified

Outgoing claim type: Name ID

Outgoing name ID format: Transient Identifier

● Pass through all claim values

○ Replace an incoming claim value with a different outgoing claim value

Incoming claim value:

Outgoing claim value:

○ Replace incoming e-mail suffix claims with a new e-mail suffix

New e-mail suffix:

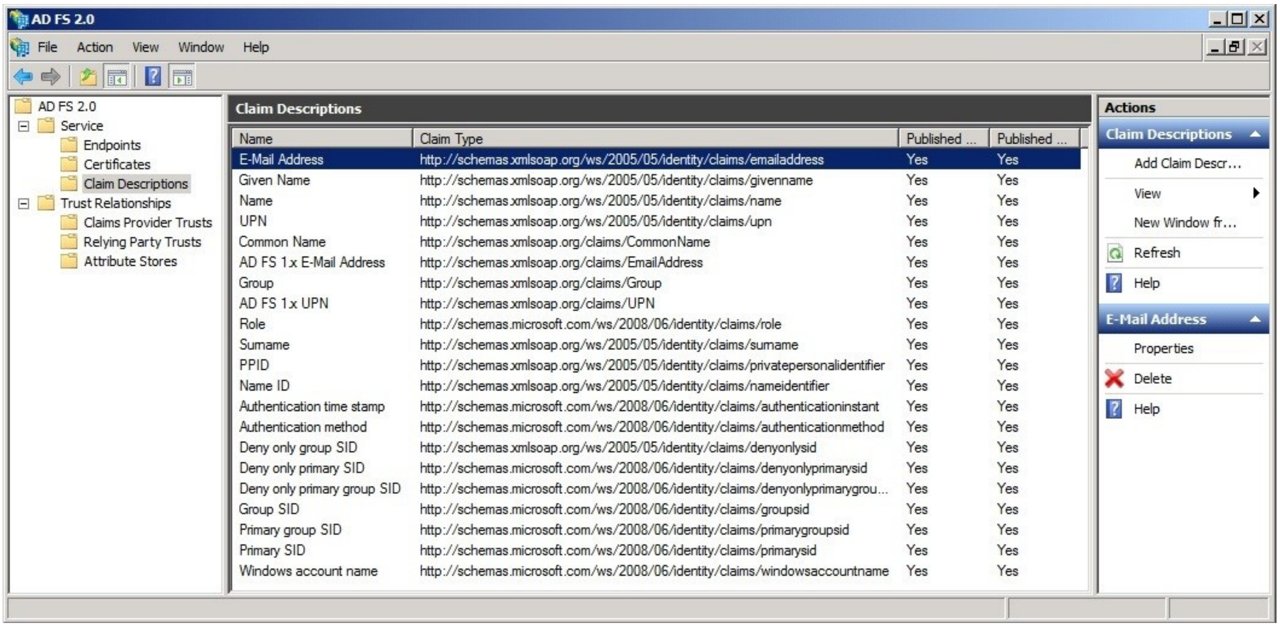
Example: fabrikam.com

We will use similar rule as for persistent name identifier. This time we will change the format of the claim to indicate that Name Identifier will be transient.

4.6.2 AD FS 2.0 as a SP

AD FS 2.0 does not support the account linking scenario. Such a scenario can still be achieved in some ways with an appropriate incoming policy.

4.7 Sharing attributes with SAML 2.0 SPs



Claim Descriptions

Name	Claim Type	Published ...	Published ...
E-Mail Address	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	Yes	Yes
Given Name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	Yes	Yes
Name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name	Yes	Yes
UPN	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn	Yes	Yes
Common Name	http://schemas.xmlsoap.org/claims/CommonName	Yes	Yes
AD FS 1.x E-Mail Address	http://schemas.xmlsoap.org/claims/EmailAddress	Yes	Yes
Group	http://schemas.xmlsoap.org/claims/Group	Yes	Yes
AD FS 1.x UPN	http://schemas.xmlsoap.org/claims/UPN	Yes	Yes
Role	http://schemas.microsoft.com/ws/2008/06/identity/claims/role	Yes	Yes
Surname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	Yes	Yes
PPID	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier	Yes	Yes
Name ID	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier	Yes	Yes
Authentication time stamp	http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant	Yes	Yes
Authentication method	http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod	Yes	Yes
Deny only group SID	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/denyonlysid	Yes	Yes
Deny only primary SID	http://schemas.microsoft.com/ws/2008/06/identity/claims/denyonlyprimarysid	Yes	Yes
Deny only primary group SID	http://schemas.microsoft.com/ws/2008/06/identity/claims/denyonlyprimarygrou...	Yes	Yes
Group SID	http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid	Yes	Yes
Primary group SID	http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid	Yes	Yes
Primary SID	http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid	Yes	Yes
Windows account name	http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname	Yes	Yes

ADFS has default-configured claims for typical shared attributes, each with a friendly name and URI "claim type".

The URIs become the "Name" of the claim in the SAML XML; the "Name" field from the UI is not used.

Consequently, you need to configure 3rd party solutions using URIs for expected assertion attribute "Names" for proper attribute sharing.

4.8 HTTP Artifact binding

SAML 2.0 includes HTTP POST and HTTP Artifact bindings for passing SAML assertions:

- HTTP POST is a push model where the SAML assertion is sent to the SP through browser intermediary;
- Conversely, HTTP Artifact is a pull model where SP retrieves the SAML assertion directly from IdP using a reference (generated by IdP, and passed through browser)

AD FS 2.0 defaults to the HTTP POST binding. One can see some disadvantages in this default configuration in the sense that it relies on JavaScript and mandates signed assertions, since tokens pass through the browser intermediary.

As previously outlined, AD FS 2.0 also supports the HTTP Artifact binding.

> **Note:**
>
> *In order to have the HTTP Artifact Binding available in AD FS 2.0, AD FS 2.0 should be configured to use a Microsoft SQL Server configuration database. The Windows Internal Database (WID), a variant a variant of SQL Server Express included with Windows Server 2008 R2 cannot be used in this context.*

The JavaScript can be avoided (no browser intermediary)

AD FS 2.0 does not support creating/consuming unsigned assertions, regardless of binding. Fortunately, 3rd parties do support signed assertions using the Artifact binding.

> **Note:**
>
> *Artifact also has its disadvantages as it implies more round trips. You can also dive into firewall and proxy issues with the direct SP to IdP connection.*

4.9 Pre-formatted hyperlinks

Some SAML 2.0 products use links to initiate federation vs. hitting target application directly:

- In the SP-initiated use case, this avoids the HRD/WAYF issue;
- For IP-initiated use case, this also results in fewer browser redirects.

4.9.1 IdP-initiated use case

AD FS 2.0 deploys a Web application, called the Sign-In Pages, to handle passive federation requests. AD FS 2.0 supports the IdP-initiated SAML 2.0 SSO via the IdpInitiatedSignOn.aspx page.

This page is one of the several Sign-In Pages as listed in the following table.

ASP.NET page	Function
AutoLogon.aspx	Attempts to sign in automatically using an Identity Selector. The sign-in request succeeds only if the user previously opted to use an Information card with this STS.
HomeRealmDiscovery.aspx	Presents a selection UI for users to select the organization to which they belong.
FormsSignIn.aspx	Handles form-based authentication with user name and password.
SignOut.aspx	Handles sign-out requests.
IdpInitiatedSignOn.aspx	Presents a selection UI for users to select a relying party application to sign in to. This page only works for relying party applications that use the SAML 2.0 protocol.
Error.aspx	Displays authentication errors to the user.
MasterPages/MasterPage.master	A master page template for all the pages.

> **Note:**
>
> *For all the details and the API reference, be sure to check out the MSDN documentation for AD FS 2.0[60].*

This page allows initiating a sign-in to a relying party from this AD FS 2.0 instance by choosing the relying party from the drop-down list, the user will first sign in to this Claims Provider (either directly or by federating with another IdP), and will then be redirected to the chosen relying party with the appropriate SAML assertion.

[60] Active Directory Federation Services 2.0 SDK documentation: http://msdn.microsoft.com/en-us/library/ee914589.aspx

This page supports the LoginToRP parameter for specifying the relying party identifier URI.

Out of the box, AD FS 2.0 only supports the LoginToRP parameter for identifier that map to SAML 2.0-based relying party, as mentioned in the page Sign-In Pages Overview[61]:

> "*Request initiated by AD FS 2.0. In this case, the user requests to sign in to the RP application directly from AD FS 2.0. This is handled by the IdpInitiatedSignOn.aspx page. This is limited to RP applications that understand the SAML protocol.*"

In other words, AD FS 2.0 does not support SAML 2.0-based IDP-initiated SSO to a WIF relying party application. Please refer to section § 4.10 SSO from SAML 2.0 IdPs to WIF relying party applications.

The rationale here is that the WS-Fed Passive protocol doesn't explicitly support an IdP-initiated flow. Additionally, WIF applications out of the box didn't go a reasonable behavior when presented with an unsolicited response. The "other side of the coin" here is that the LoginToRP parameter is most useful in the SAML 2.0 world since SAML 2.0 protocol messages are harder to handle.

This said, the pre-formatted hyperlink is something like:

```
<a href="https://<your_ADFS_sts>/adfs/ls/IdpInitiatedSignOn.aspx?LoginToRp=<SAML_partner_URI>”> Link to Test IDP-
initiated POST Single Sign-on from AD FS 2.0</a>
```

Until the Update Rollup 2 for AD FS 2.0, the support for other parameters is limited.

IsPassive/ForceAuthn, Consent, RequestedAuthenticationContext are all declared in the aspx page, but they require page customization.

Moreover, no support is officially provided for ProtocolBinding, the default binding for partner in configuration will be used.

Likewise, AD FS 2.0 RTW does not support specifying the relay state in the case of IdP-initiated request. (The support for RelayState is limited to echoing back in SP-initiated requests.) The relying party must identify the target resource in its configuration.

To pass relay state in ADFS 2.0 RTW, there was a **non-supported workaround** which requires some custom code (for additional information, please refer to the discussions How can I specify the target URL directly in the SAML request and have AD FS 2.0 automatically redirect?[62] and Specify RelayState URL[63].

The Update Rollup 2 for AD FS 2.0 update adds a new capability that enables AD FS 2.0 to now consume relay state in order to redirect the user to the RP application. For more information, please see the article Supporting Identity Provider Initiated RelayState[64] on Microsoft TechNet.

4.9.2 SP-initiated use case

The act of initiating federated access to an AD FS 2.0-protected application can use a preformatted hyperlink, or a user can visit the application directly and take advantage of a feature in AD FS 2.0 called home realm discovery (HRD) via the *HomeRealmDiscovery.aspx* page. This is essentially SP-

[61] Sign-In Pages Overview : http://msdn.microsoft.com/en-us/library/ee895359.aspx

[62] How can I specify the target URL directly in the SAML request and have AD FS 2.0 automatically redirect?: http://social.msdn.microsoft.com/Forums/en/Geneva/thread/af6dc3dc-d24a-48de-a83e-b173af2a7e6f

[63] Specify RelayState URL: http://social.msdn.microsoft.com/Forums/en-US/Geneva/thread/91812934-e620-44c7-b4ef-8383083dc3c4

[64] Supporting Identity Provider Initiated RelayState: http:/technet.microsoft.com/en-us/library/jj127245(v=ws.10).aspx

initiated SSO, because it results in AD FS 2.0 sending an authnrequest to the IdP, but it provides an interface to allow a user to select their IdP from a list.

AD FS 2.0 does not support SAML 2.0 SP-initiated SSO via hyperlink but there's a workaround for WS-Fed Passive speaking application as detailed in the next section.

4.10 SSO from SAML 2.0 IdPs to WIF relying party applications

IdP-initiated SSO to a WIF relying party application using SAML 2.0 is not possible. As of writing, indeed, WIF speaks WS-Federation and not SAML 2.0.

To avoid HRD/WAYF, you need to use a WS-Federation SP-initiated hyperlink instead.

AD FS 2.0 will convert to using SAML 2.0 automatically once it looks up partner:

```
<a href="https://<your_ADFS_sts>/adfs/ls/?wa=wsignin1.0&wtrealm=<federated_app_URL>&whr=<SAML_partner_URI>”> Link to
Test SP-Initiated POST Single Sign-On to OIF from AD FS 2.0</a>
```

The WS-Federation query string parameters are the following ones:

Parameter	Description
whr	This value is a URI that uniquely identifies the requestor IdP that SHOULD receive the wsignin1.0 request message
wa	The value MUST be the literal string "wsignin1.0"
wtrealm	This parameter MUST be included in a request message to a different security domain/realm from the relying party. If present, this value MUST be a URI that the requestor IdP and the relying party have agreed to use to identify the security domain/realm of the relying party in messages to the requestor IdP. If present, the *wreply* parameter MUST NOT be present
wreply	This parameter MUST be included in request messages to the same security domain/realm as the relying party. If present, this value MUST be a URL at the relying party to which responses MUST be directed. If present, the *wtrealm* parameter MUST NOT be present.

4.11 IdP Discovery

IdP Discovery is a way for SPs to figure out where to redirect an SSO request when no IdP is identified. This is:

- Similar to AD FS 2.0 home realm discovery (HRD);
- Useful for SAML 2.0 implementations without a WAYF/HRD page/mechanism.

The whole process uses a “common domain cookie” as described in section § 4.3.1 of the SAMLProf document.

After a user's first authenticates, the IdP appends its unique ID to a list of visited IdPs in a cookie in user's browser. The SP simply reads the list. Some vendors use the most recently visited IdP, but can also present the list for user selection.

As previously outlines, AD FS 2.0 ships a Web application that reads/writes common domain cookies. You can find it in Program Files, in CDC.Web folder.

In the IdP role, AD FS 2.0 can append its unique ID to CDC IdP list. This requires configuration in the CDC application and in AD FS 2.0, both in web.config files.

In the RP role, AD FS 2.0 can read cookie but using contents requires customizing the *HomeRealmDiscovery.aspx* HRD page.

For additional information, you can refer to the page CUSTOMIZING THE SIGN-IN PAGES USING WEB.CONFIG[65] in the AD FS 2.0 SDK documentation.

[65] CUSTOMIZING THE SIGN-IN PAGES USING WEB.CONFIG: http://msdn.microsoft.com/en-us/library/ee895366.aspx