K2 Authentication

4.6.7

March 2014

K2 Authentication

Claims and OAuth in K2

- Configuration
  - Supported Configurations
- Supported Technologies and Terminology
- Introduction to OAuth
- Token Flow in OAuth and Claims

K2 smartforms Claims and OAuth

- K2 Multi Auth Introduction
  - K2 Multi-Auth - Configure SmartForms for Azure Active Directory (AAD)
  - K2 Multi-Auth - Configure SmartForms for Active Directory Federation Services (AD FS)
  - K2 Multi-Auth - Configure SmartForms for SQL Server User Manger (SQLUM)
  - K2 Multi-Auth - Consolidation to Multi-Auth
- K2 Multi Auth Upgrading Secondary smartforms sites to 1.0.6
- K2 smartforms Authentication Management Settings

K2 for SharePoint

- Overview and Architecture of K2 for SharePoint
- The SharePoint 2013 Service Brokers
- SharePoint 2013 Integration Requirements

Claims and OAuth in K2

Configuration

Configuring OAuth and claims should be handled for you by the installation and configuration of K2 blackpearl. If you are integrating with SharePoint 2013, the K2 for SharePoint App should be used to configure your environment. Manual configuration is possible but not recommended.

If you have SmartForms installed you can get to the following forms for managing OAuth and Claims settings:

- Manage Resource Types: [SmartForms URL]/ Runtime/Runtime/Form/Manage+Resource+Types/
- Manage Resources (instances): [SmartForms URL]/ Runtime/Runtime/Form/Manage+OAuth+Resources/
- Manage Tokens: [SmartForms URL]/ Runtime/Runtime/Form/Manage+OAuth+Tokens/
- Manage Claims: [SmartForms URL]/Runtime/Runtime/Form/Manage+Claims/
- Manage Issuers: [SmartForms URL]/Runtime/Runtime/Form/Manage+Issuers/
- Manage Site Realms: [SmartForms URL]/Runtime/Runtime/Form/Manage+Site+Realms/

You can also browse to these forms by going to the K2 Designer and clicking All Items > System > Management > Security > Forms.

For more information about Site Realms see Introduction to Multi-Auth in the SmartForms User Guide.
For specific information on how to configure K2 blackpearl and K2 smartforms for specific systems, see the following topics in the SmartForms User Guide:

- Configure SmartForms for AAD
- Configure SmartForms for SQLUM
- Configure SmartForms for AD FS

If you do not have SmartForms installed and you need to manually modify OAuth and Claims settings, you can use the SQL script examples below. Note that these contain values from a fictitious company called Denallix. You will have to replace these values with actual values from your environment. The values that must be replaced include:

Claims Configuration

You need the following values for configuring claims:

- Identity.AddIssuer
    - Name: The name of the issuer.
    - Issuer: The issuer.
    - Thumbprint: The thumbprint of the issuer. Can be retrieved using the attached script.
    - Description (optional): The description of the issuer.
    - Uri: The URI of the issuer.
    - UseForLogin: Surfaces the name of this issuer on the SmartForms login page if true.
- Identity.AddClaimTypeMapping

    - SecurityLabel: If configuring a third-party WS-Federation-based SAML identity provider, you must register a custom security provider with a unique label.
    - ClaimTypeInfo (True/False): If you want to have groups in your idP (like SharePoint), you should set this to True. Note: There can't be two claim type mappings with the same label, the same value for ClaimTypeInfo (true in this case), and Nii. These three properties must be unique across all labels.
    - Nii (NamedIdentityIssuer): The named identity issuer. Use the attached script to discover this. Note: This is not necessary for SharePoint 2010 and is only relevant to SharePoint 2013.
    - GroupSPSTSRSTR (typically c:0+.w for Windows authentication): The prefix of groups in SharePoint for this claim mapping. Note that the Everyone group does not have the proper prefix. The w in the example and in the example below stands for Windows.
    - UserSPSTSRSTR (typically i:o#.w for Windows authentication): The prefix for user logins in SharePoint for this claim mapping. Use the attached script to get an idea of what these are for your environment.
- Identity.AddRealm (Note: If you have SmartForms installed you will see at least three entries in the ClaimAudience table. These correspond to the design, runtime and view flow applications. View flow is necessary to configure so that users can be authenticated with the view flow site. The other two are only necessary if you're running SmartForms, and are probably setup for you correctly but are included here for completeness.)
    - RealmUri: The identifier for the realm (typically the address of the resource).
    - HomeRealm (typically NULL): If you have multiple identity providers for a given STS, you can specify a default idP here. It will always go to that idP.
    - Freshness (typically 0): Indicates the upper bound of the credential's age in minutes. A value of zero means that the STS should immediately verify the identity or use the minimum age credentials possible when verification is not possible.
    - SignOutReplyUri (always NULL): Not used.
    - PersistentCookiesOnPassiveRedirects (typically True/1): If set to true, the cookie is persisted across browser sessions.
    - ReplyUri (typically the relative part of RealmUri): The default reply URI.

```
 USE[K2]
 GO
 DECLARE@IssId INT
 EXEC@IssId =[Identity].[AddIssuer] @Name="Your STS",
@Issuer=YourSTS,@Thumbprint="THUMBPRINT",
@Description=NULL,@Uri="http://yourURI.com/something",
@UseForLogin=1
 EXEC[Identity].[AddClaimTypeMapping]@IssuerId=@IssId,
@SecurityLabel=K2,@ClaimTypeInfo=True,
@Nii="urn:office.idp.activedirectory",
@GroupSPSTSRSTR="c:0+.w",@UserSPSTSRSTR="i:o#.w"
 EXEC[Identity].[AddRealm] @IssuerId=@IssId,
@RealmUri="https://k2.denallix.com/ViewFlow/",@HomeRealm=NULL,
@Freshness=0,@SignOutReplyUri=NULL,
@PersistentCookiesOnPassiveRedirects=1,@ReplyUri="/ViewFlow/"
 EXEC[Identity].[AddRealm] @IssuerId=@IssId,
@RealmUri="https://k2.denallix.com/Designer/",@HomeRealm=NULL,
@Freshness=0,@SignOutReplyUri=NULL,
```

```
@PersistentCookiesOnPassiveRedirects=1,@ReplyUri="/Designer/"
 EXEC[Identity].[AddRealm] @IssuerId=@IssId,
@RealmUri="https://k2.denallix.com/Runtime/",@HomeRealm=NULL,
@Freshness=0,@SignOutReplyUri=NULL,
@PersistentCookiesOnPassiveRedirects=1,@ReplyUri="/Runtime/"
```

Warning: Do not manually update the K2 database tables unless instructed to do so by K2 Support.

Download the PowerShell script that returns the current claims configuration GetClaimsConfig.

Configuring OAuth

To configure OAuth you must add the following items:

- Resource Type: Use the [Authorization].[AddOAuthResourceType] stored procedure.
- Resource Type Parameters: Use the [Authorization].[AddOAuthResourceTypeParameter] stored procedure. This stored proc requires many parameters that can be found in the procedure. Use an existing OAuth resource as well as knowledge of the resource you're setting up to determine which parameters you need to specify. The out-of-the-box resources, such as SharePoint, SharePoint S2S and Azure Active Directory have specific parameters that you can use as examples. Find these in the Authorization.OAuthResourceTypeParameter table.
- Resource: The resource is the instance of the resource type. You must register an instance of a resource type to then fill in the values of the resource type parameters.
- Resource Values: The values of the resource type parameters.

Supported Configurations

The following configurations are supported by K2. Check the table below to see which architecture best fits your environment. A generic claims-enabled line of business (LOB) system must be manually configured.

- Claims-enabled LOB: Must be manually configured
  - The claims configuration resides in the K2 database. It is no longer stored in web.config or K2HostServer.exe.config files. If you configured claims before the process is similar. If you had an existing claims configuration before upgrading, your claims configuration was moved into the K2 database.
  - It is highly recommended that you use SmartForms to configure OAuth and Claims. For more information see Configuration (Insert link to blackpearl topic).

- SharePoint

Version	Authentication	Configuration
SharePoint 2010	Windows	None
SharePoint 2010	Claims (Windows, Forms or Trusted Identity Provider (IdP))	Manual Configuration or Script
SharePoint 2013	Claims (Windows)	Run the Registration Wizard
SharePoint 2013	Azure Active Directory	Run the Registration Wizard, also see KB 1443

SharePoint 2013	Claims(Forms or Trusted Identity Provider (IdP))	Manual Configuration or Script

Supported Technologies and Terminology (Claims)

OAuth

K2 supports OAuth 2.0 for authorization flow between SharePoint, K2, AzureAD and other OAuth 2.0-compatible systems. OAuth Resource Types can be registered with K2 and then Resources setup as instances of those types. The following Resource Types are automatically registered when installing K2 blackpearl:

- Azure Active Directory
- SharePoint
- SharePoint S2S

An on-premises SharePoint site will register a SharePoint S2S resource for interacting with the SharePoint site, while a SharePoint Online site will register an Azure Active Directory and a SharePoint resource instance.

SAML

K2 supports SAML (Security Assertion Markup Language) versions 1.1 & 2.0. SAML tokens are XML representations of claims, which is what many Microsoft products are adopting for authentication and authorization. The K2 platforms include a STS (Security Token Service) which cracks and packages claims into SAML tokens so that it can communicate with other claims-aware applications and line of business systems.

Terminology

Term	Definition
Claim	A statement that one subject makes about itself or another subject. For example, the statement can be about a name, identity, key, group, privilege, or capability. Claims have a provider that issues them, and they are given one or more values. This data about users is sent inside security tokens (SAML).
Claim rule	A rule that is written in the claim rule language of the provider that defines how to generate, transform, pass through, or filter claims.
Security Assertion Markup Language (SAML)	A protocol that specifies how to use HTTP Web browser redirects to exchange assertions data. SAML tokens are XML representations of claims.
Identity Provider (IdP)	A Web service that handles requests for trusted identity claims and issues SAML tokens. An identity provider uses a database called an identity store to store and manage identities and their associated attributes.
Relying Party (RP)	An application that consumes claims to make authentication and authorization decisions. For example, the K2 server receives claims that determine if the issuer user can access K2 data.
Claims-aware application	A relying party software application that uses claims to manage identity and access for users.
Security Token Store (STS)	A Web service that issues security tokens. SharePoint implements a STS to authorize activities within the application from multiple authentication providers

Secure Sockets Layer (SSL)	A protocol that improves the security of data communication by using a combination of data encryption, digital certificates, and public key cryptography. SSL enables authentication and increases data integrity and privacy over networks. SSL does not provide authorization or non-repudiation.
Active Directory Federation Services (AD FS)	A component of Windows Server 2008 that supports identity federation and Web single sign-on (SSO) for Web browser-based applications.
Federation server	A computer running Windows Server 2008 or Windows Server 2008 R2 that has been configured using the AD FS 2.0 Federation Server Configuration Wizard to act in the federation server role. A federation server issues tokens and serves as part of a Federation Service.
Federation Service	A logical instance of a security token service such as AD FS 2.0.

Introduction to OAuth

What is OAuth

OAuth is a standard, simple, and secure authorization protocol that enables users to approve an application to act on their behalf without sharing their user name and password. It is an authentication standard that allows users to grant specific credentials for the sharing of resources from web, mobile and desktop applications. This enables users to share their specific private resources or data (i.e. lists, documents, etc.) that are stored on one site/application with another site/application.

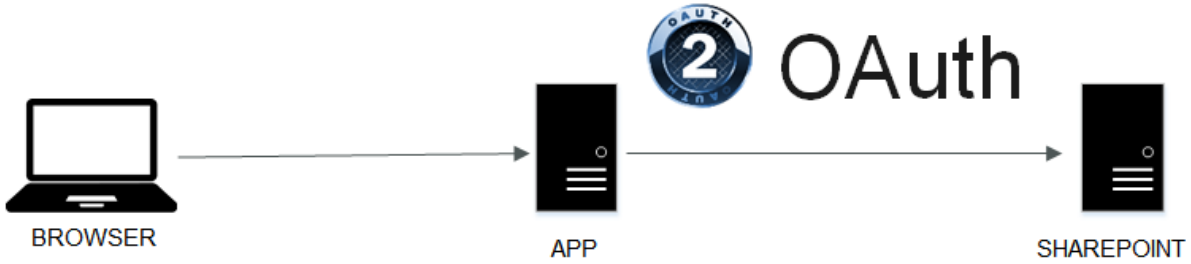
Why K2 uses OAuth

OAuth enables outbound data calls with claims-based users' identity. This allows K2 to access line of business (LOB) resources, such as SharePoint, Google, Twitter, and Facebook, on behalf of the user.
OAuth is not outbound claims, and when using OAuth, K2 does not create or pass Security Assertion Markup Language (SAML) tokens via brokers.

Where K2 uses OAuth

K2 smartforms uses OAuth to create trust credentials that can then be shared across diverse systems. This allows a user to securely access resources without needing to continually login to those systems.

OAuth is used in K2 for SharePoint as part of the 'App' Model. In SharePoint 2007 and 2010 the K2 integration had to be installed on the SharePoint server. In SharePoint 2013, Apps are able to run outside of SharePoint, and OAuth is used to allow the K2 App to perform actions on behalf of the SharePoint user. The App requests the level of permission it requires, and these can only be granted by someone with that permission level.



How K2 uses OAuth

OAuth uses a token system to manage sessions. There are two token types used, access tokens and refresh tokens.

Access Token

The K2 App will use the access token whenever it interacts with SharePoint resources. K2 stores the access token with the expiration value, and if the token has expired, or is about to expire in the next 60 seconds (this value is configurable), then the refresh token is used to request a new access token (see below). In SharePoint 2013, access tokens are good for 12 hours. Access tokens are used to create a client-side object model (CSOM) ClientContext that identifies the user and authorization level for the particular action.

Refresh Token

The K2 App will use the refresh token to request a new access token without prompting the user to "Trust It" again. The K2 App will update the stored refresh token with every request. A refresh token is good for 6 months after creation.

Token Conditions

The following three conditions occur when using OAuth in K2.

No Token: When there is no token present for the application, the OAuth Trust dialogue will be presented in order for the user to authorize the creation of the access and refresh tokens. The app communicates with the K2 Host Server and receives the 'No Token' condition. It then initiates the Trust dialogue between the K2 system, the SharePoint system, and the backend identity and authentication servers (Security Token Services and Active Directory).

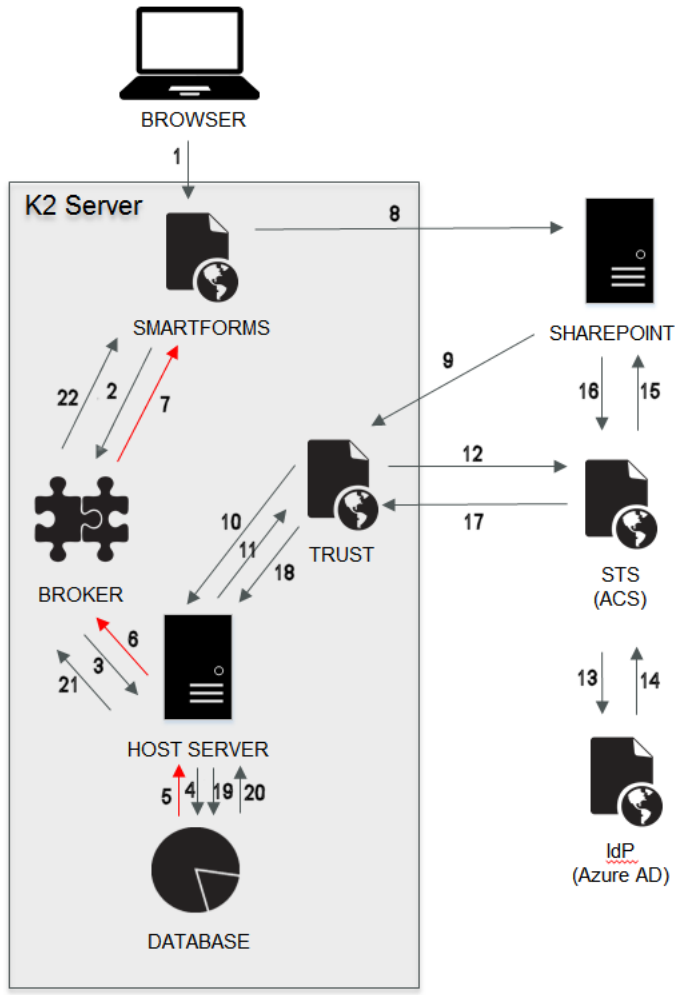


Image Data Flow
The K2 App through K2 smartforms (1) sends an authentication request through the broker layer (2) to the K2 server (3). The K2 server then interrogates the database (4) for an access token. If an access token is not stored for that user (5) then it returns the request (6) with the 'No Token' condition (7). K2 smartforms then initiates (8) the trust dialogue with SharePoint. If the user selects to 'Trust' the K2 App (9), a trust request is then sent to the K2 server (10). The K2 server sends a trust request (11) to the STS (12), which authenticates the user using Active Directory and SharePoint (13-16), and returns an access token through the Trust site (17) to the K2 server (18). That access token is then stored in the database (19) and a valid token condition is then returned to K2 smartforms (20-22).

Valid Token: When the access and refresh tokens have been created, and the app calls for an OAuth token, the K2 Host Server retrieves the token data from the database and uses it to perform the data functions between K2 and SharePoint, as shown below:

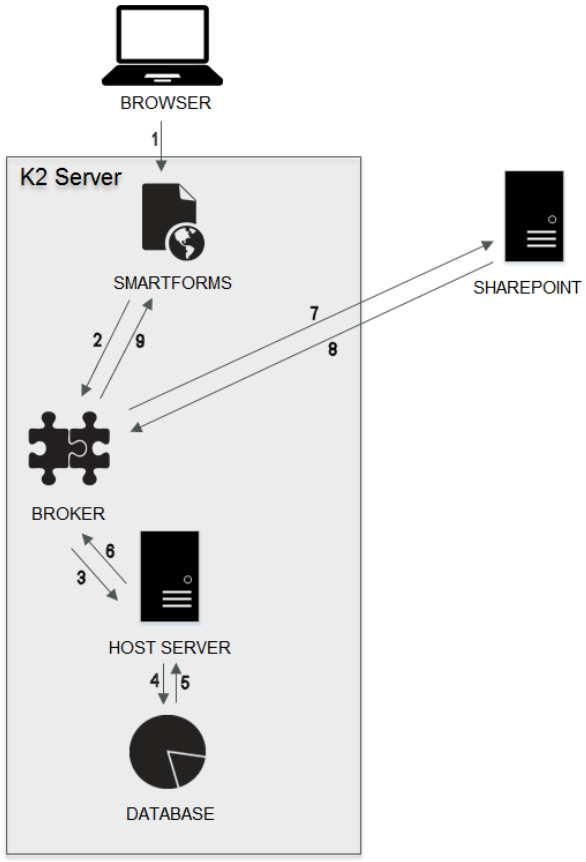


Image Data Flow
The K2 App through K2 smartforms (1) sends an authentication request through the broker layer (2) to the K2 server (3). The K2 server then interrogates the database (4) for an access token. The access token that is stored in the K2 Database for that user (5) is returned through the broker (6) to SharePoint (7). SharePoint then uses the token to authenticate the action, and SharePoint data is then returned to K2 smartforms (8-9) and displayed in the K2 App.

Expired Token: When an access token is about to expire or has already expired, the refresh token is used to create a new access token. This takes place from within the K2 Host Server.

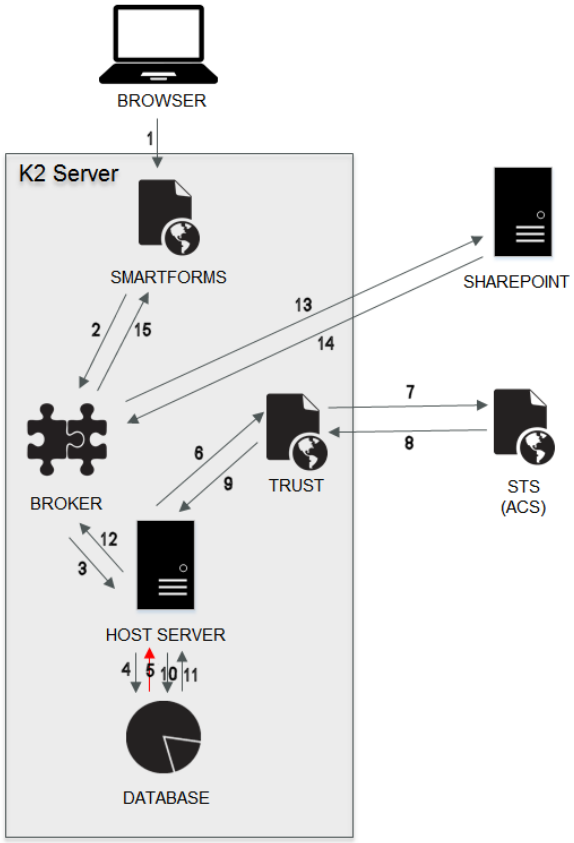


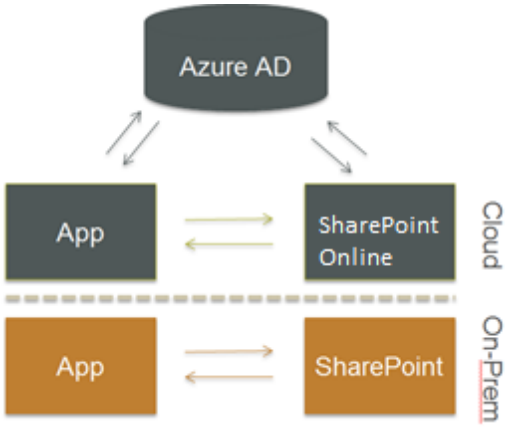
Image Data Flow
The K2 App through K2 smartforms (1) sends an authentication request through the broker layer (2) to the K2 server (3). The K2 server then interrogates the database (4) for an access token. The access token that is stored in the K2 Database for that user has expired so an 'Expired Token' condition is returned to the K2 server(5). The K2 server initiates a 'Refresh Token' request through the Trust site (6) to the STS (7). The STS issues a new Access Token (8) which is returned to the K2 server (9) and stored in the database (10). That is returned through the broker (11-12) to SharePoint (13). SharePoint then uses the token to authenticate the action, and SharePoint data is then returned to K2 smartforms (14-15) and displayed in the K2 App.

OAuth and Claims

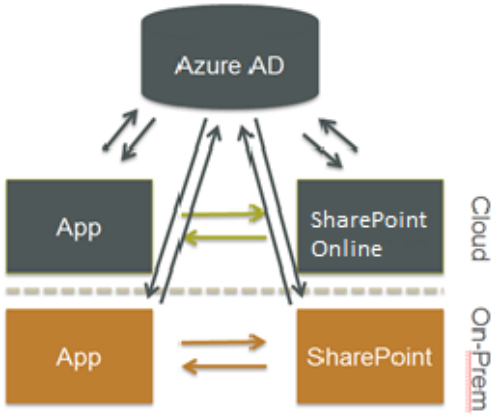
OAuth is NOT a replacement for claims-based authentication; the OAuth token system is simply a way to provide access to a calling application for an already authenticated and authorized user. K2 users still login via an identity provider / Security Token Service (STS) which will issue a Security Assertion Markup Language (SAML) token containing the user's identity claim. That identity claim is used to uniquely identify that user in both SharePoint and K2. While using OAuth, K2 relies on the configuration of a K2 user manager to provide authentication and user and group resolution for identity stores such as Azure Active Directory, SharePoint Site Collection Groups, Active Directory, SQL, LDAP or Custom directory stores. K2 provides the ability for incoming claims-based authentication through configuration of mappings between claims-based identity providers and K2 user managers and more specifically the user's identity claim.

On Premises vs Online

The authentication path of the OAuth token system for K2 in the cloud is different to using K2 on-premises. The following diagram demonstrates the three-legged path between the K2 App, Azure AD and Office 365/SharePoint Online vs the K2 App to SharePoint path used on a local system.



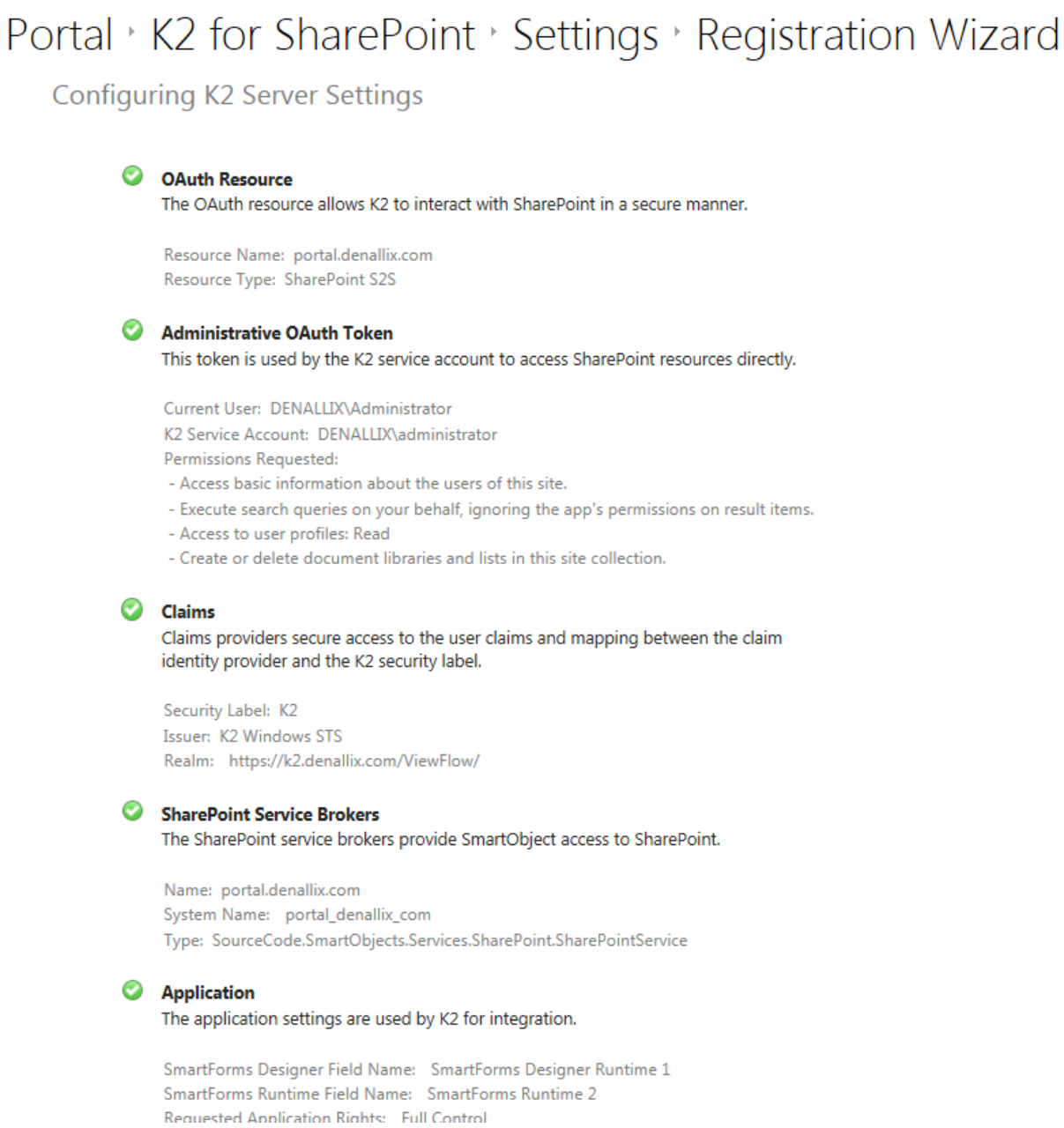
The authentication path of the OAuth token system in a hybrid environment (online and on-premises) adds the three legged path between the K2 App, Azure AD, and the on-premises SharePoint server.



K2 for SharePoint Settings

The Trust request for OAuth is presented when first adding the app to the SharePoint site. Note: The K2 for SharePoint App must be added to each site in SharePoint. Once trusted, the K2 for SharePoint App Registration Wizard presents a configuration screen to create the OAuth Resource.

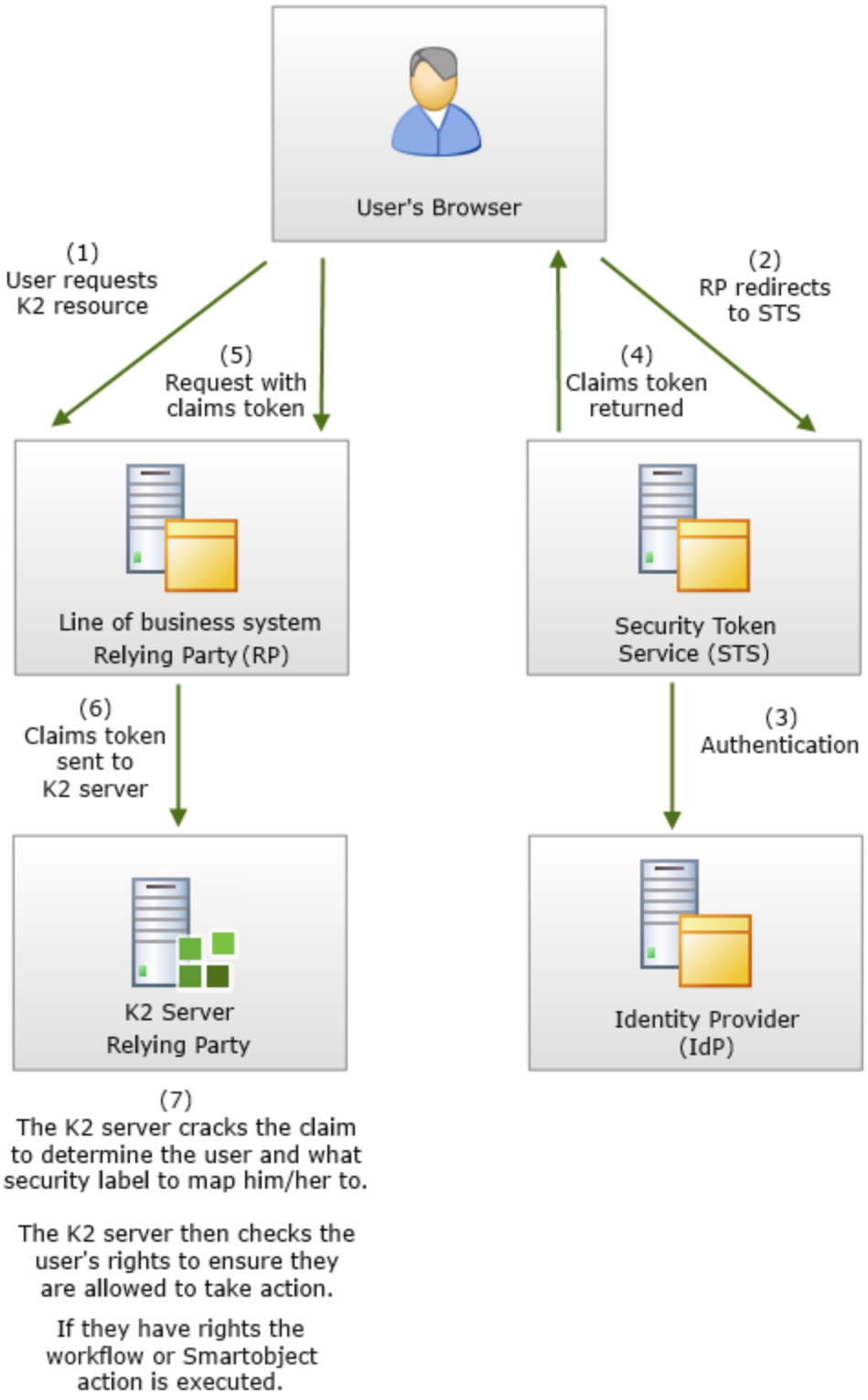
The Registration Wizard creates the tokens that the K2 App will use to communicate securely with SharePoint. The Registration Wizard will then report back on the creation of these settings.



Use the K2 smartforms Authentication Management Settings page to modify the OAuth configuration or to delete OAuth tokens. See K2 smartforms Authentication Management Settings for more information.

Token Flow (OAuth and Claims)

The token flow for claims authorization is similar to what it was in previous versions of K2 blackpearl. However, because K2 now has its own Security Token Service (STS) and service brokers can be configured for OAuth, the requirement for the K2 server to have direct access to a line of business (LOB) system, such as SharePoint, via Windows authentication is removed.
In the following diagram a generic LOB system is the relying party.

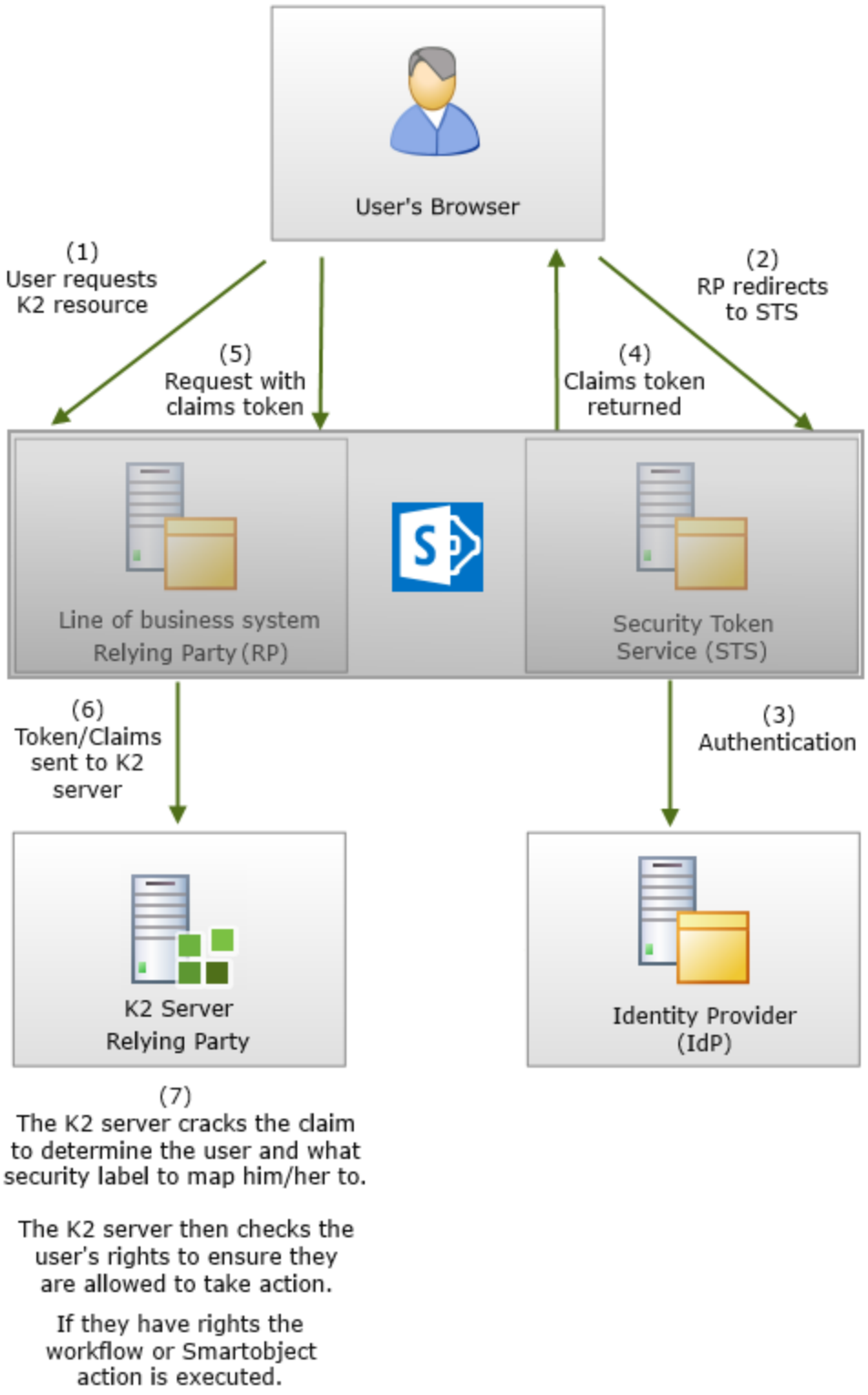


Functional Flow

1. The user initiates the request via the browser, and the request contains a K2 resource that must be handled by the K2 server.
2. The LOB Relying Party (RP) server then redirects the user to the Security Token Service (STS).

3. The STS sends the user to the Identity Provider (IdP) which he or she authenticates with.
4. The STS then sends the claims token back to the user's browser.
5. The browser then redirects back to the LOB system with the claims token.
6. The LOB system sends the claims token on to K2.
7. K2 cracks the claim, maps the user to a K2 label and takes the workflow or SmartObject action if the user has rights.

In the following diagram, SharePoint 2013 is the relying party and the STS.
Note: This diagram only applies to SharePoint 2013 on-premises. SharePoint Online's STS is Azure AD.



Functional Flow

The functional flow is the same as with any LOB system except for the fact that SharePoint 2013 on-premises includes both the LOB relying party and the STS. Also, once K2 cracks the claim and needs anything from the SharePoint server, such as through the SharePoint 2013 service broker, it uses OAuth to authenticate with SharePoint. For on-prem, a SharePoint Server to Server (S2S) high-trust token is used and for SharePoint online the token associated with the K2 service account is used.

TODO: Note which scenarios preserve the original user's identity and which ones do not (list item/document vs. List/Library/Site).

For more about OAuth see Introduction to OAuth.

K2 smartforms Claims and OAuth

Introduction to Multi-Auth

With the introduction of K2 blackpearl 4.6.7 and K2 smartforms 1.0.6 it is now possible to configure a single K2 smartforms site with multiple authentication providers. This document describes what Multi-Auth is and how it works.

What is Multi-Auth?

Multi-Auth is the ability for users to authenticate to a single SmartForms site using different authentication methods. Why is this important?

In most Microsoft-centric organizations Windows Authentication is used as the default authentication provider because it works seamlessly with Active Directory and does not require users to enter login details when accessing the site. The credentials used to log in to Windows are able to be passed to the site through Internet Explorer.

In some organizations Active Directory is not used so a different security provider is registered with K2 which points to a different service for authenticating users. In this case the users will need to provide a username and password to log in to the SmartForms site.

It is becoming increasingly common, however, for organizations to require a mixture of authentication schemes. It might be a scenario where Vendors or Partner organizations need access to internal resources or a scenario where a merger has brought in users from a different directory service. Whatever the case may be these organizations require an easy way to allow all their users access to the same resources.

Multi-Auth in K2 has added the ability for the user accessing the site to indicate how they would like to authenticate. The site then knows how to properly switch between authentication schemes depending upon the user's selection.

One side effect to Multi-Auth is that Integrated Windows Authentication users, who are not used to any kind of intermediate step between clicking a link and seeing a form, will now occasionally need to select their authentication mode. Authentication is cached by the browser so they will not need to make their choice each time, but if the credentials are removed from the cache they will need to make the selection again.

How does Multi-Auth Work?

The following K2 sites now support Multi-Auth.

- K2 smartforms Runtime
- K2 smartforms Designer
- View Flow

To enable Multi-Auth some changes were made to how these sites work as well as how the K2 server handles authentication.

Claims Authentication

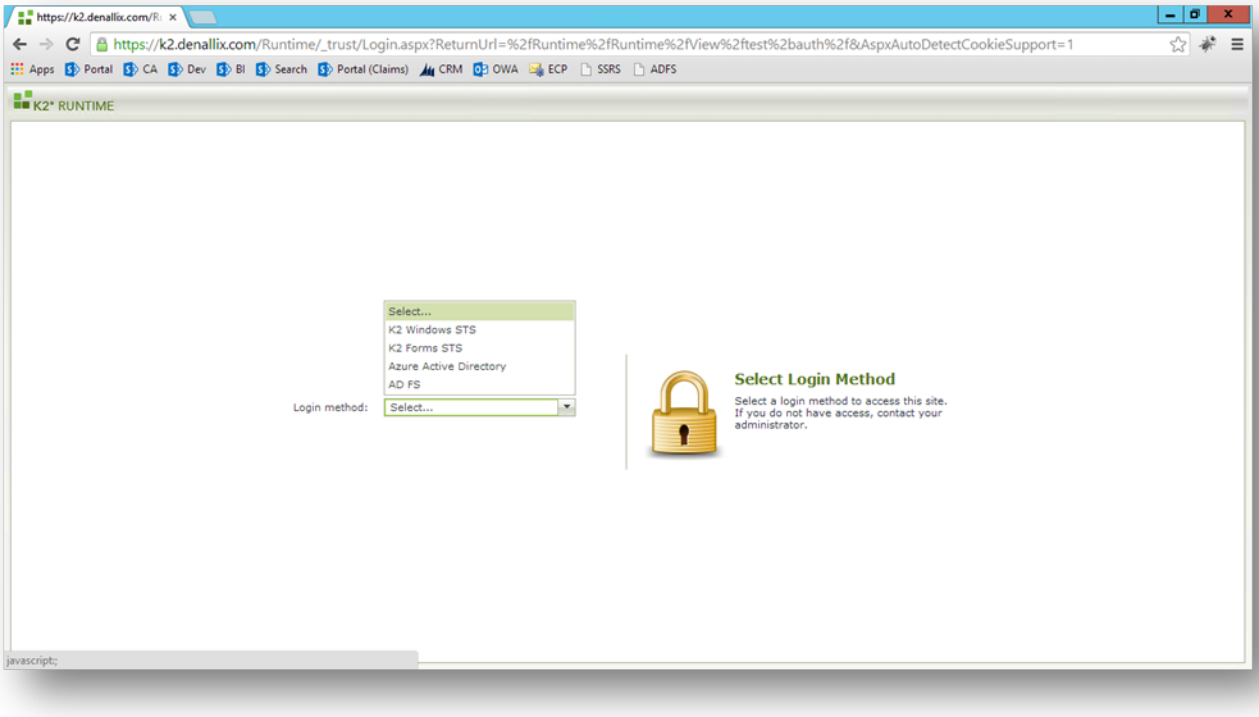
All authentication between SmartForms sites and K2 is done using Claims Authentication. You can read more about claims authentication in the K2 blackpearl Getting Started Guide. The basic idea is that when a user authenticates against a particular provider, that provider can issue a set of standardized claims which tells others systems who the user is and what system authenticated them. Claims can also be issued to give a lot more information about a user, but for our purposes only the Identity Provider, and Security Label claims are used.

Not all Identity Providers issue claims based security tokens today. Most notably is Active Directory. AD only issues Windows Authentication tokens. To solve this problem K2 has provided a Security Token Service (STS) that converts Windows tokens into claims known as the K2 Windows STS. Similarly, since Forms Authentication also does not issue claims a K2 Forms STS has also been provided. Systems like Active Directory Federation Services (AD FS) and Azure Active Directory (AAD) issue claims through their own STSs.

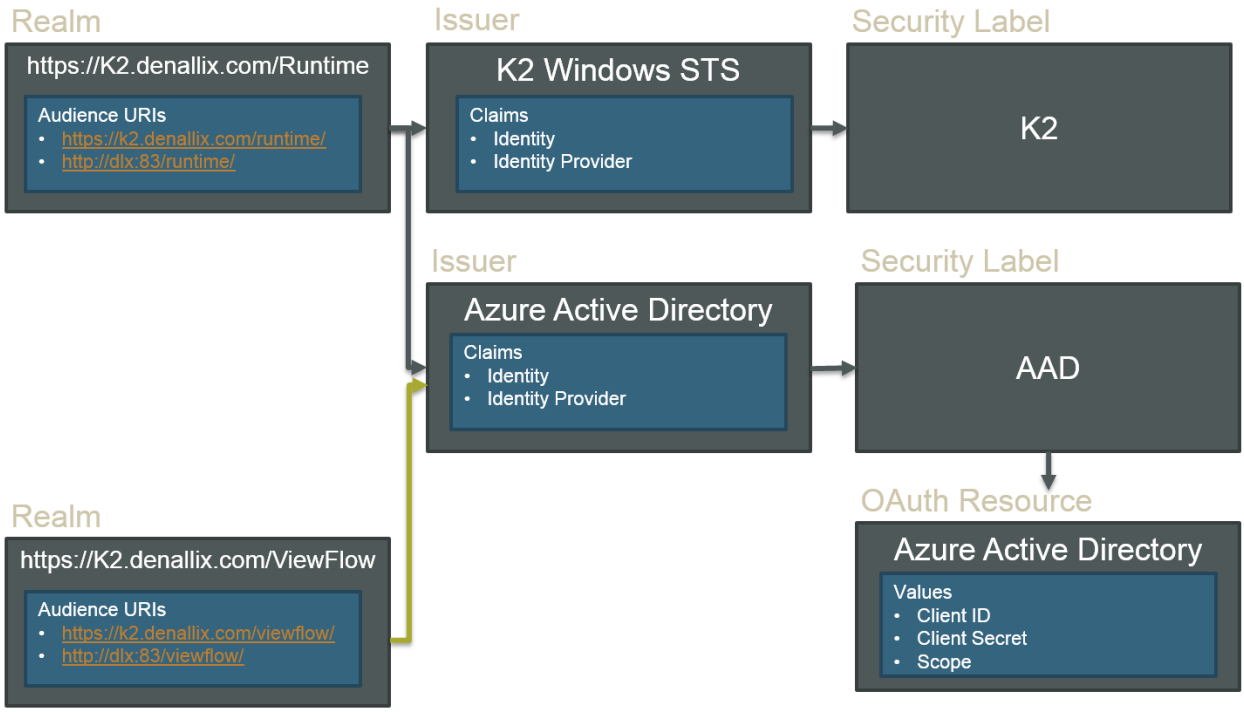
Once a claims based security token comes into the K2 Server from SmartForms, the K2 Server checks a list of known Claim Issuers to see if there is a match. If there is no match an exception is raised. If there is a match in the Issuer list, the security token is "cracked" so that the claims can be read. K2 checks a list of claims that have been registered with the Issuer to determine who the user is. Once the identity of the user has been found the K2 security label that is mapped to the Issuer is added to the identity and the user is authenticated to K2.

Realms

Claims Authentication takes care of authenticating the user once the user has a security token issued from an STS. Getting the user to the right STS is all about Realms. A Realm is an identifier for a site. K2 uses a URI based identifier which is the URL of the site. Other systems might use other formats such as URN.
Each SmartForms site is configured to map to a Realm. The list of Realms to choose from is configured in K2. When the SmartForms site loads it checks with the K2 Server to understand what Issuer configurations are mapped to the Realm of the SmartForms site. If there is more than one Issuer mapped to the Realm, then the SmartForms site is in Multi-Auth mode and will present a page to the user for them to select which Issuer to use to log in. However, if there is already a security token cached in the user's browser, the security token is sent through so that the user can be authenticated automatically without having to select an Issuer. Also, if there is only one Issuer configured for the Realm, then no login method selection is necessary and the user will be directed to the log in screen.



When the user selects an Issuer, or there is only one Issuer for the Realm, the URI that is stored with the Issuer configuration is used to route the user to the appropriate STS for log in. In the case of K2 Windows STS, which uses Windows Authentication, the log in happens without any user interaction. In the case of other STSs like AAD, AD FS, K2 Forms STS, etc. a login screen will be presented. The user enters valid credentials and a Security Token is attached to the redirect back to SmartForms and Claims Authentication takes over.
Below is a diagram that explains the relationship between Realms, Issuers, and Security Labels.



What about Anonymous?

It is not possible in K2 smartforms 1.0.6 to have anonymous authentication coexist with other authentication schemes using the new Multi-Auth model. If you require an anonymous SmartForms runtime site follow the steps in the Configure a Secondary SmartForms Runtime Site for Anonymous Access document.

Configuration

Authentication and Claims configuration was previously done in a combination of web.config files and the K2host-server.exe.config file. With K2 blackpearl 4.6.7, this configuration has all moved to the database. K2 smartforms 1.0.6 ships with management pages that allow you to set up and manage the configuration quickly and easily from a browser.

The following articles describe the configuration steps for the various available authentication options.

- Configure SmartForms for AAD
- Configure SmartForms for AD FS
- Configure SmartForms for SQLUM
- Consolidation to Multi-Auth

K2 Multi-Auth: Configure SmartForms for Azure Active Directory (ADD)

With the introduction of K2 blackpearl 4.6.7 and K2 smartforms 1.0.6 it is now possible to configure a single K2 smartforms site with multiple authentication providers. This document outlines the configuration steps necessary for enabling Azure Active Directory (AAD) for K2 smartforms sites.

Configuration

Prerequisites

The following prerequisites are required for configuring SmartForms for AAD:

- K2 blackpearl 4.6.7
- K2 smartforms 1.0.6
- Microsoft .NET Framework 4.5
- Azure Active Directory
- SSL-enable the web site that hosts the K2 smartforms virtual directories

High Level Configuration Steps

These high-level steps are provided for those familiar with configuring claims integration. For a detailed guide, see the Detailed Steps topic below.

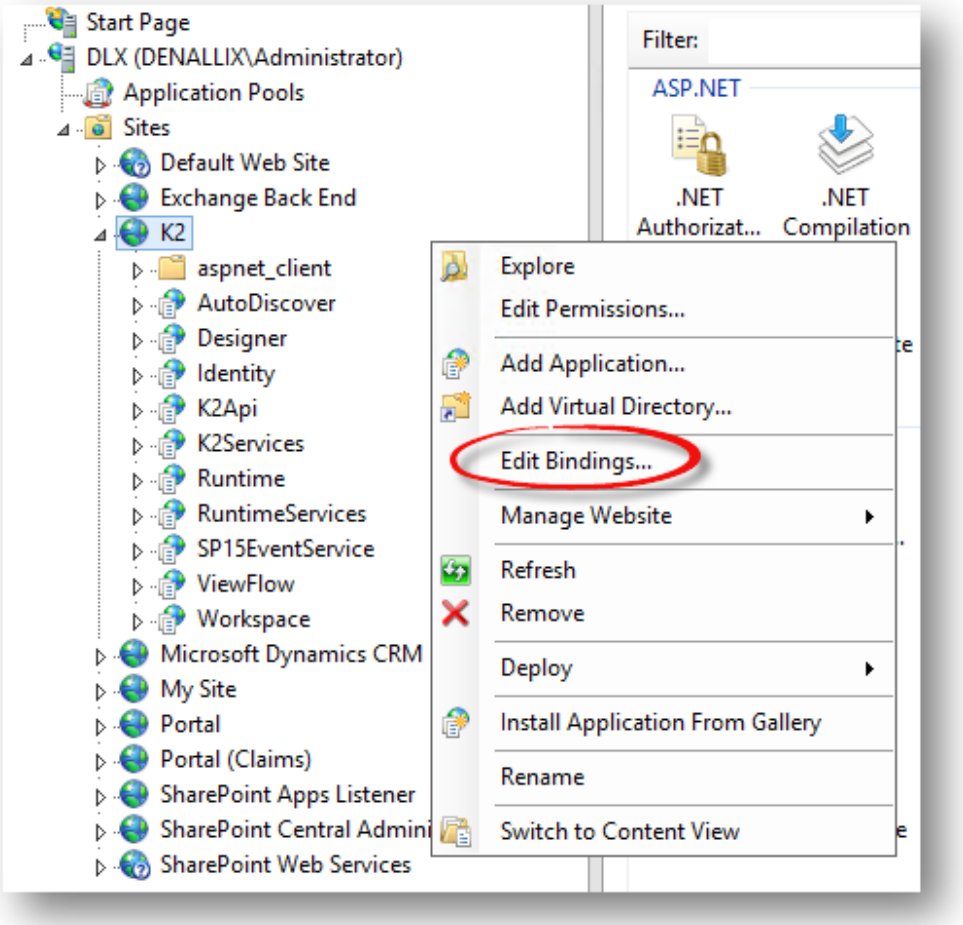
1. SSL-enable the web site that hosts the K2 smartforms virtual directories
2. Create an App in AAD for your SmartForms site
3. Register an OAuth resource in K2 for AAD
4. Add the AAD Security Label in K2
5. Configure the Claim Issuer in K2
6. Configure the Claim Mappings in K2
7. Configure the Realm and Audience URIs to Issuer mapping

Detailed Steps

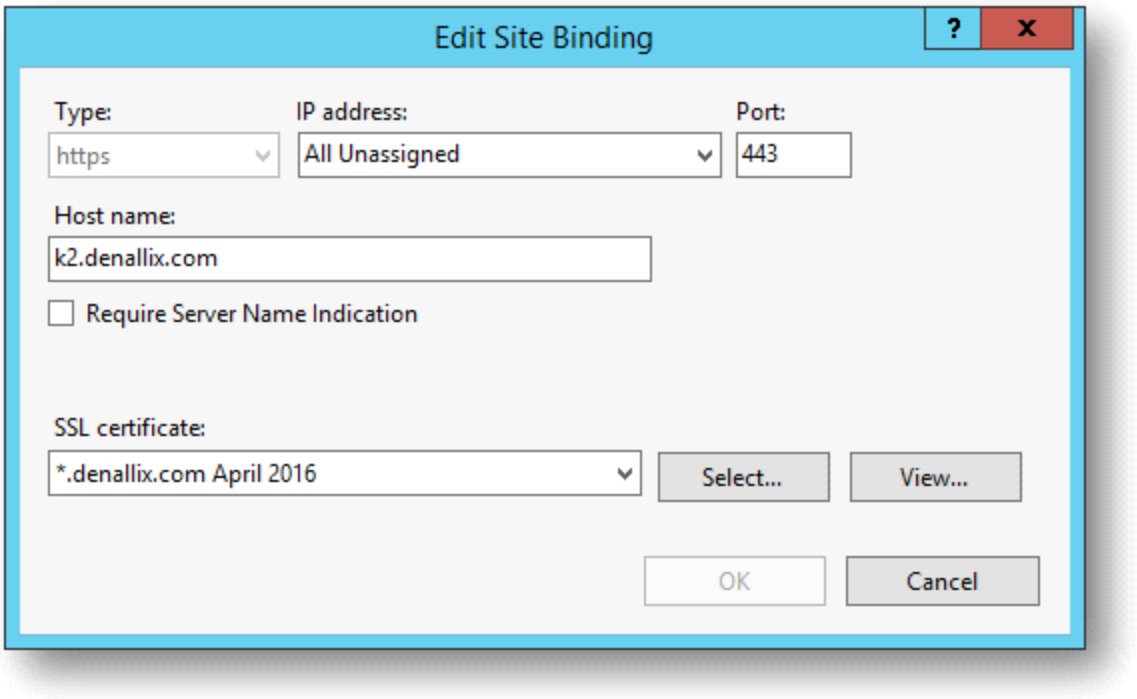
Step 1 – Create an SSL-enabled site for SmartForms

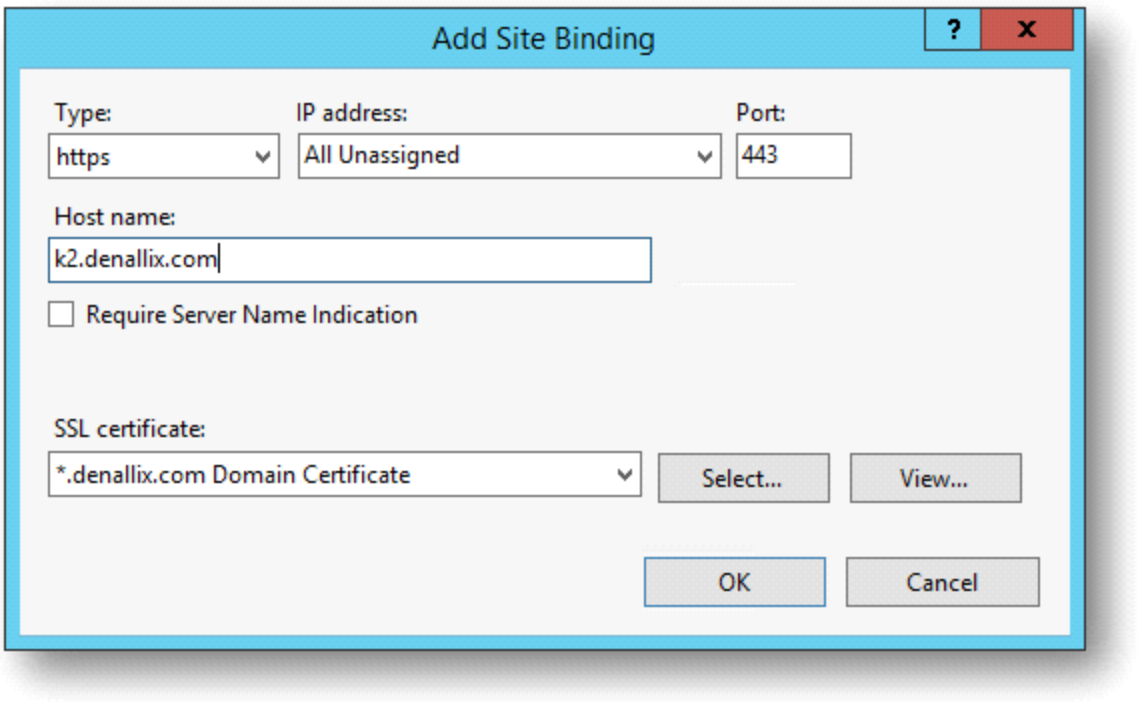
When configuring SmartForms to work with AAD it is required to have your SmartForms site enabled for SSL.

1. Open Internet Information Services (IIS) Manager
2. In this example the K2 site is used. Right click the K2 site and select Edit bindings



3. Click Add
4. Select https from the Type drop down list and type a new number into the Port field
5. Select the certificate to use for your site. In this example it is the *.denallix.com April 2016 Certificate, which was purchased from a valid Certificate Authority.
   Note - When working with systems like AAD that are external to the organization, Self-signed and Domain Certificates will not work.



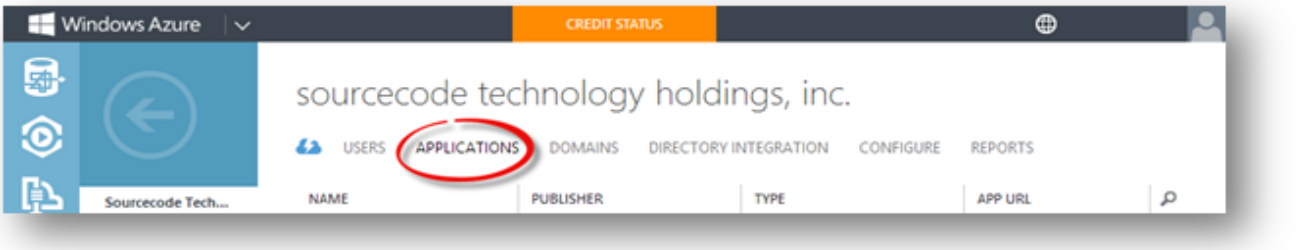


6. To make sure that the new HTTPS configuration is updated properly in the K2 configuration rerun the K2 black-pearl Setup Manager and the K2 smartforms Setup Manager.

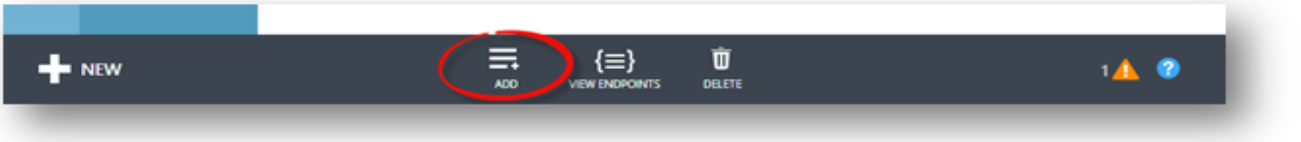
Step 2 - Create an App in Azure Active Directory for your SmartForms site

Like most OAuth based resources, Azure Active Directory uses an application registration model. In order for your SmartForms sites to be able to connect to your AAD tenant for login you must first configure an AAD application.

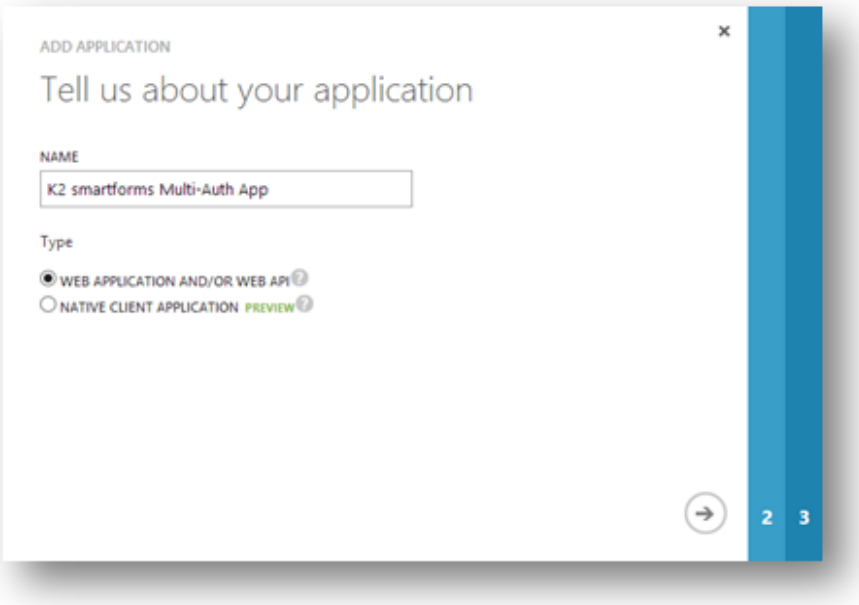
1. Log in to https://manage.windowsazure.com and navigate to your Active Directory Domain and click on the Applications tool bar



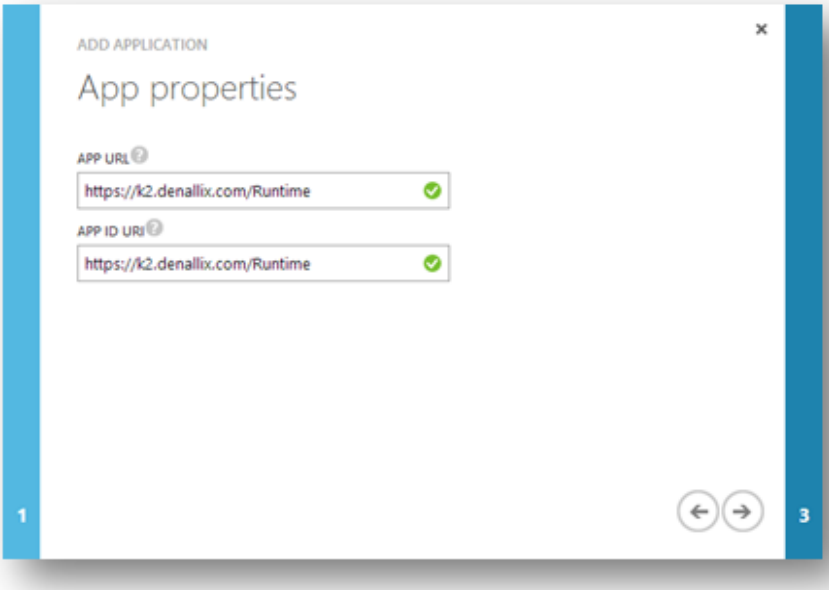
2. At the bottom of the applications screen click on ADD to add a new application



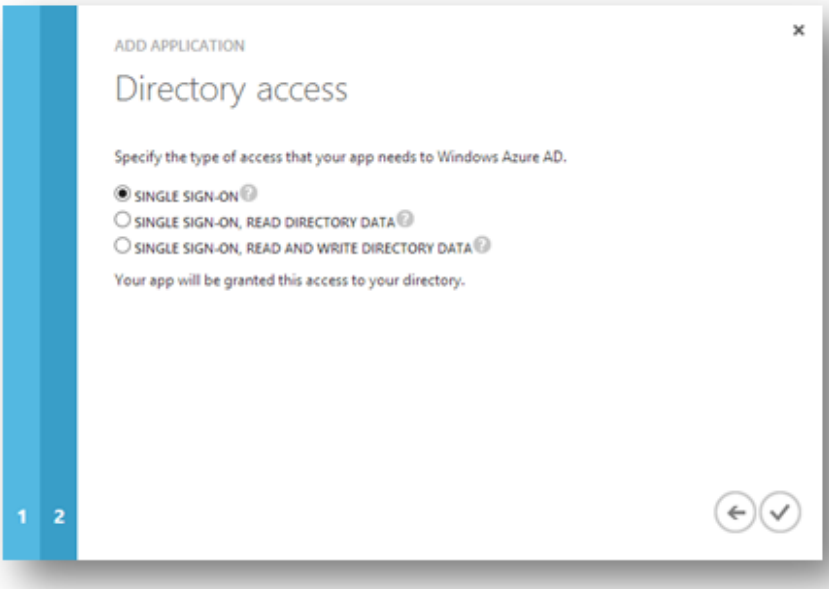
3. Select the Add an application my organization is developing option in the popup window
4. Give the application a name and select the Web Application and/or Web API option



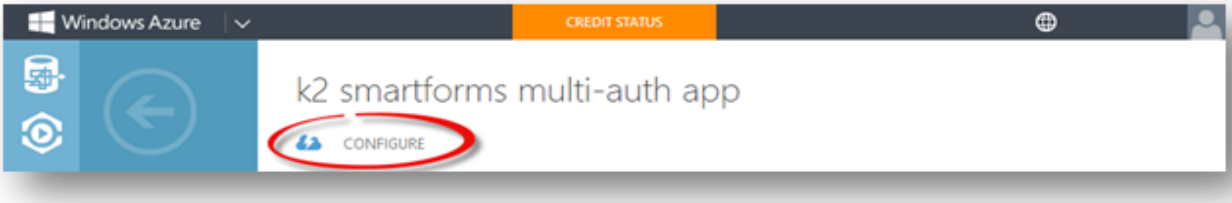
5. Enter the URL to your SmartForms site for the APP URL and APP ID URI fields and click next



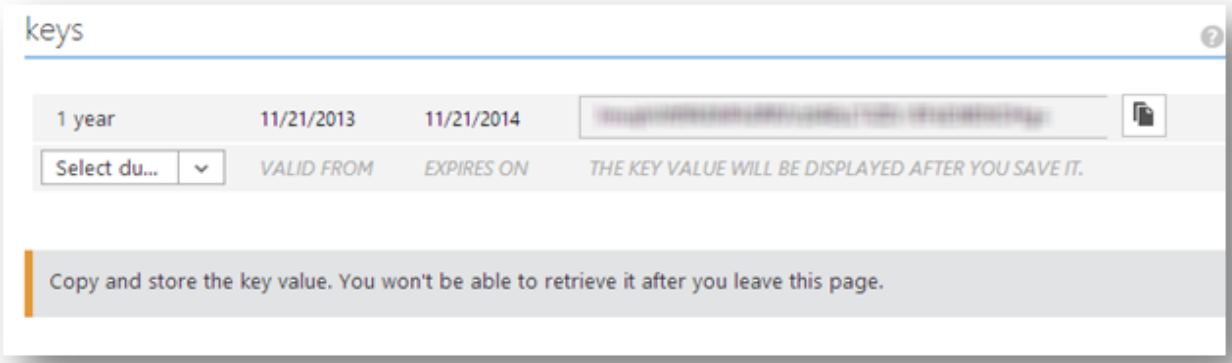
6. Select Single Sign-On as the type of directory access required and click the check mark to complete the wizard



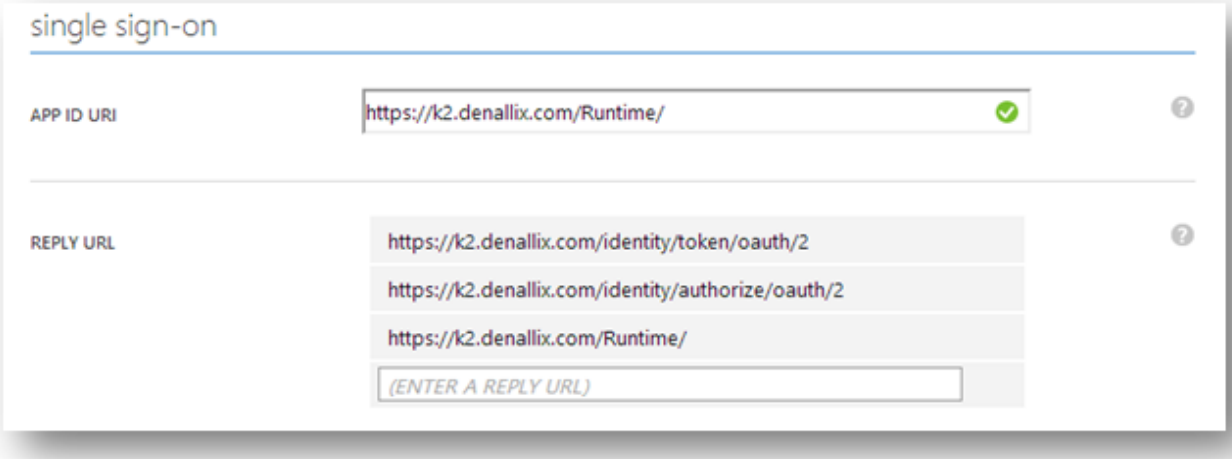
7. Once the App has been created click on the Configure link at the top of the page



8. Copy and save the Client ID from the configuration page
9. Under the keys section select a duration for the key and click save at the bottom of the page to retrieve the key value. Copy and save the key.



10. Under the single sign-on section add two additional reply URLs:
    a. Token Endpoint Reply URL - https://{YourK2Server}/identity/token/oauth/2
    b. Authorization Endpoint Reply URL - https://{YourK2Server}/identity/authorize/oauth/2



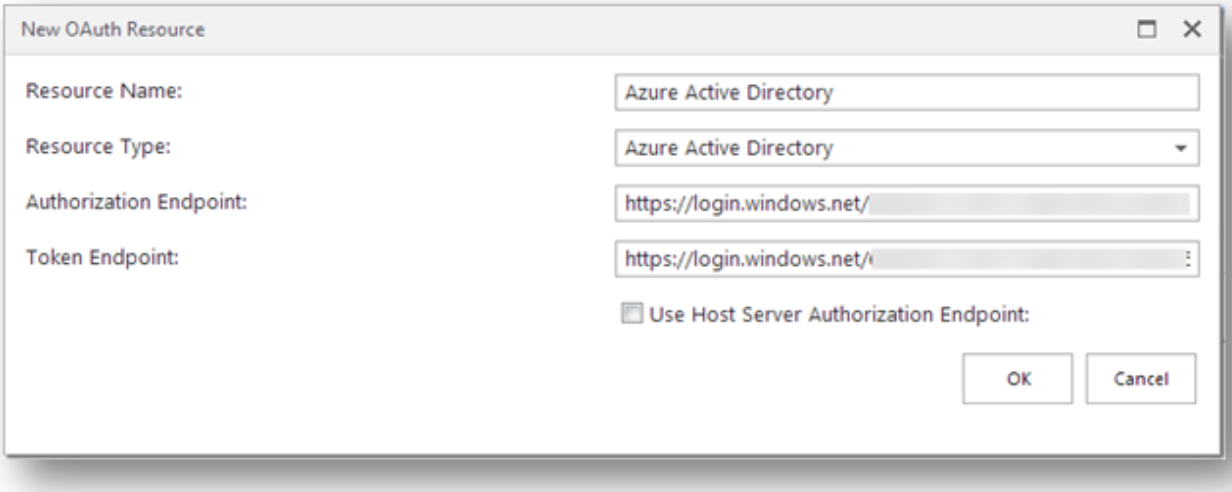
11. Click on the Manage Endpoints link at the bottom of the page and Copy and Save the Federation Metadata Document URL, the OAuth 2.0 Token Endpoint, and the OAuth 2.0 Authorization Endpoint URL.

Step 3 - Register an OAuth Resource in K2

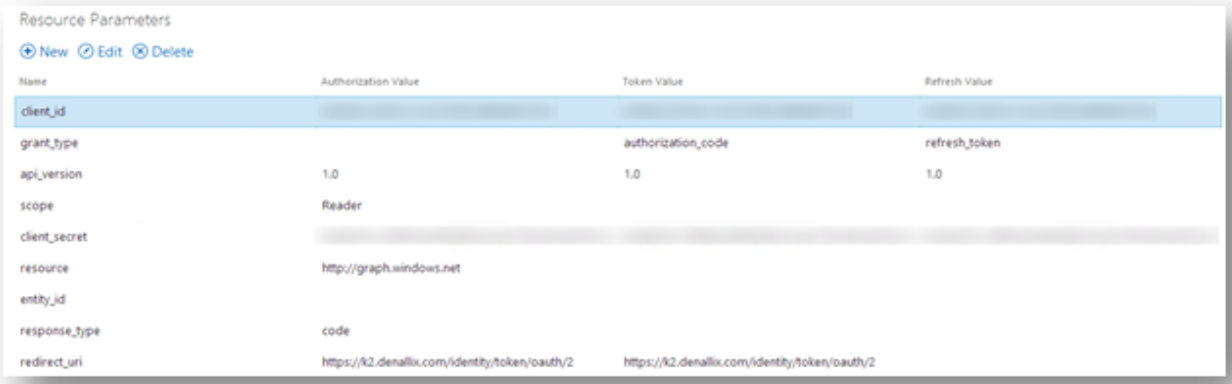
Now that you have an App configured in AAD you need to tell K2 how to request authorization tokens from AAD. This is done by configuring an OAuth resource.

1. Open the Manage OAuth Resources form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{YourK2Server}/Runtime/Form/Manage+OAuth+Resources/
2. Under Resources click New
3. Enter a Resource Name
4. Select Azure Active Directory for the Resource Type
5. Enter the Authorization Endpoint, minus the query string parameters, from the AAD App you created in Step 2 (Example: https://login.windows.net/{YourTenantID}/oauth2/authorize)

6. Enter the Token Endpoint, minus the query string parameters, from the AAD App you created in Step 2 (Example: https://login.windows.net/{YourTenantID}/oauth2/token)
7. Leave the Use Host Server Authorization Endpoint checkbox unchecked and click OK



8. Select the newly created resource from the Resources list
9. Select client_id from the Resource Parameters list and click edit
10. Enter the Client ID from the AAD app you created in Step 2 for the Authorization Value, the Token Value and the Refresh Value and click OK
11. Select api_version from the Resource Parameters list and click edit
12. Enter 1.0 for the Authorization Value, the Token Value and the Refresh Value and click OK
13. Select scope from the Resource Parameters list and click edit
14. Enter Reader for the Authorization Value and click OK
15. Select client_secret from the Resource Parameters list and click edit
16. Enter the Client Secret from the AAD app you created in Step 2 for the Authorization Value, the Token Value and the Refresh Value and click OK
17. Select resource from the Resource Parameters list and click edit
18. Enter http://graph.windows.net for the Authorization Value and click OK
19. Select response_type from the Resource Parameters list and click edit
20. Enter code for the Authorization Value and click OK
21. Select redirect_uri from the Resource Parameters list and click edit
22. Enter https://{YourK2Server}/identity/token/oauth/2 for the Authorization Value and the Token Value and click OK (replace [YourK2Server] with the server name or host header value to your K2 site)



Step 4 - Add the AAD Security Label in K2

K2 provides an out of the box security provider for AAD. Follow the steps below to configure a K2 Security Label for this security provider that uses the OAuth resource that you just created.

1. Open SQL Manager and connect to the SQL server where the K2 databases are hosted
2. Right click on the Authorization.OAuthResource table in the K2 Database and select the Select top 1000 Rows option.
3. Copy the ResourceID for the resource you added in Step 3
4. Replace the OAuthResourceID value in the RoleXmlConfig parameter of the following SQL with your ResourceID and execute the query.

```
UseK2
-- DECLARATIONS - Update as needed
 DECLARE@SecurityLabelName  NVARCHAR(20)= 'AAD'; -- the label value that will be prepended to users
and groups for the user manager
 DECLARE@XmlConfig XML=
 '<AuthInit>
 </AuthInit>'
 DECLARE@RoleXmlConfig XML=
 '<RoleInit>
<OAuthResourceID>YOUR RESOURCE ID HERE</OAuthResourceID>
 </RoleInit>'
 DECLARE@SecurityLabelID  UNIQUEIDENTIFIER ='e02d4aa0-f87a-4b5d-90f3-f03ce6c7af55'; -- GUID of Secur-
ityLabel for user manager
 DECLARE@AuthSecurityProviderID  UNIQUEIDENTIFIER = (SELECT SecurityProviderIDFROM [HostServer].
[SecurityProvider]WHERE ProviderClassName=  'SourceCode.Se-
curity.Providers.AzureActiveDirectory.SecurityProvider');
-- GUID of SecurityProvider for Authentication Services(IAuthenticationProvider)
 DECLARE@AuthInit XML= @XmlConfig-- XML initialization data for the Authentication Provider
 DECLARE@RoleSecurityProviderID  UNIQUEIDENTIFIER = (SELECT SecurityProviderIDFROM [HostServer].
[SecurityProvider]WHERE ProviderClassName=  'SourceCode.Se-
curity.Providers.AzureActiveDirectory.SecurityProvider');
 -- GUID of the SecurityProvider for User and Group Listing services (IRoleProvider)
 DECLARE@RoleInit XML= @RoleXmlConfig-- XML initialization data for the Role Provider
 DECLARE@DefaultLabel BIT= 0;--1 = true, NULL and 0 = false
```

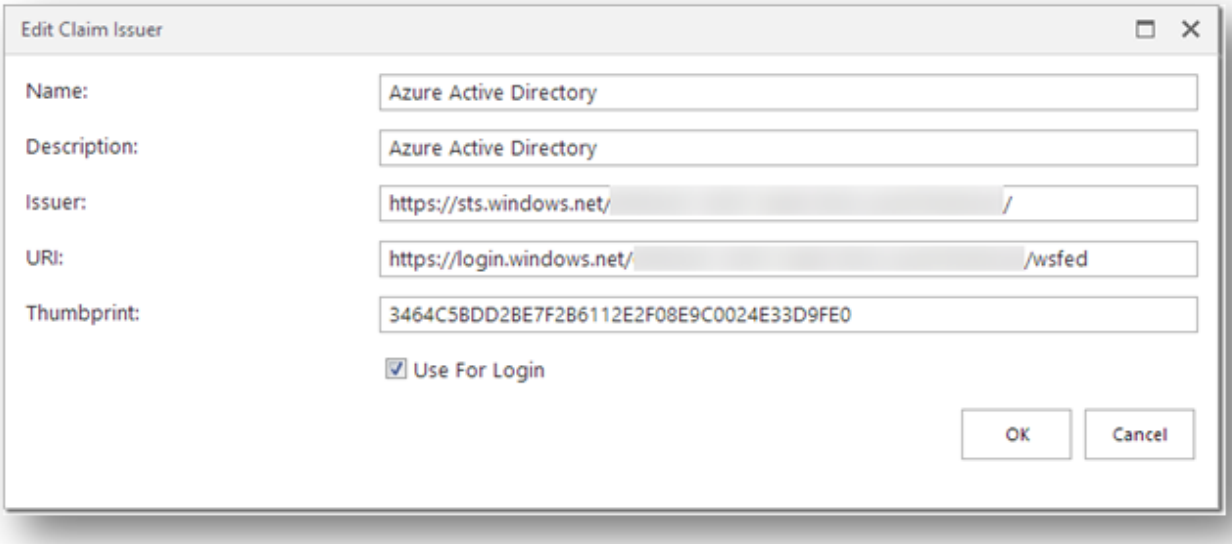
```
 DECLARE@ProviderClassName  NVARCHAR(200)=  'SourceCode.Security.Providers.AzureActiveDirectory'; --
the full .NET name of the Security Provider class

-- UPDATE TABLES
USEK2
 DELETEFROM [SecurityLabels]WHERE SecurityLabelName= @SecurityLabelName;
 INSERTINTO [SecurityLabels]VALUES
(
@SecurityLabelID
,
@SecurityLabelName,@AuthSecurityProviderID,@AuthInit,@RoleSecurityProviderID,@RoleInit,@DefaultLabel)
```

5. Restart the K2 blackpearl service

Step 5 – Configure the Claim Issuer in K2

A Claim Issuer is required for K2 to map the Security Token Service (STS) signing certificate with the security tokens.

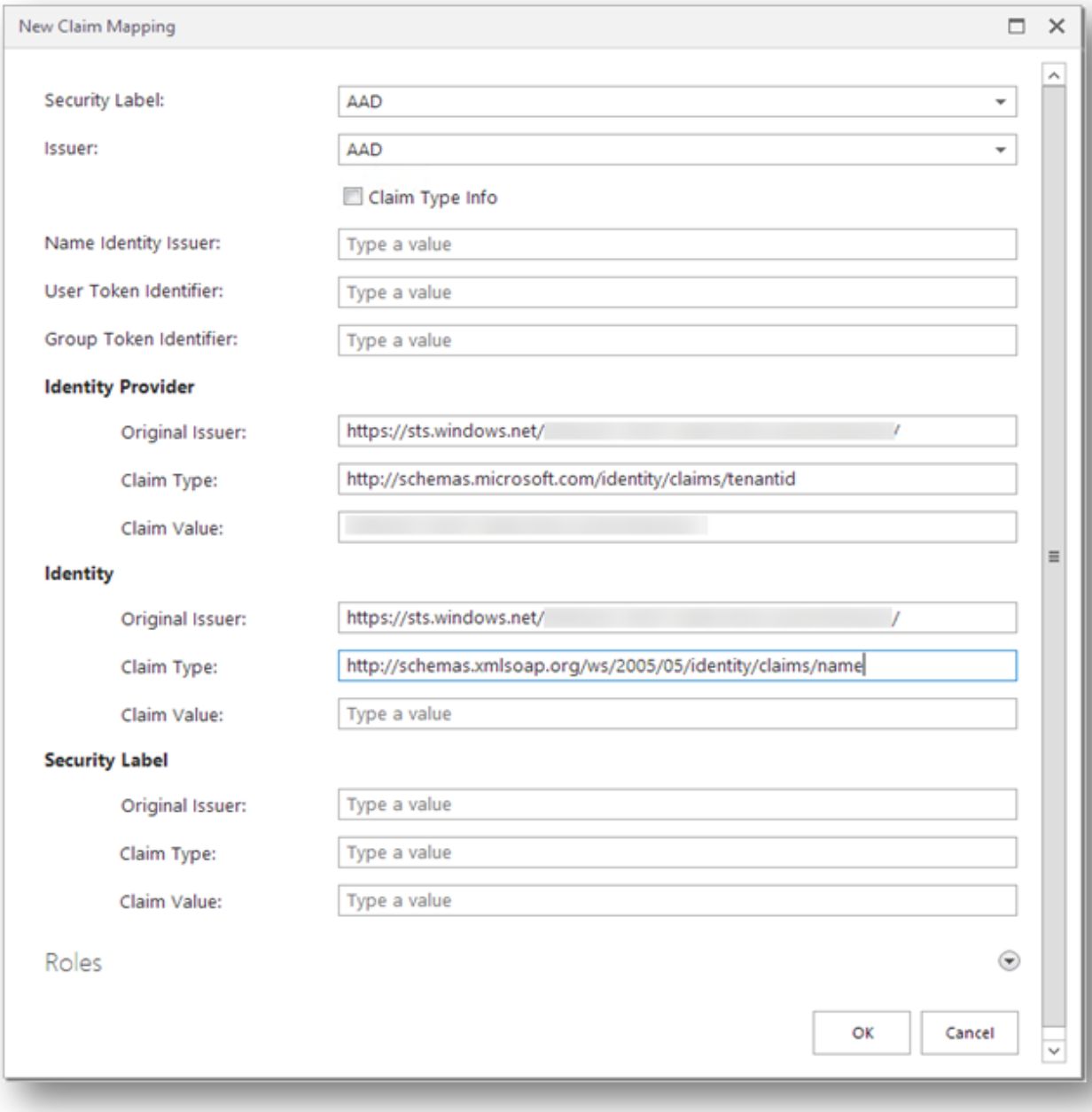
1. Open the Manage Issuers form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{YourServer}/Runtime/Form/Manage+Issuers/
2. Click New
3. Enter a Name and Description for the AAD Issuer
4. Enter the Token Issuer endpoint for AAD in the Issuer textbox. (Example: https://sts.windows.net/{YourTenantID}/)
5. Enter the Login URL for AAD in the URI textbox. From Step 2 (Example: https://login.windows.net/{YourTenantID}/wsfed)
6. Enter the Thumbprint of the Token-signing certificate for AAD. Currently all AAD instances use the same certificate so the thumbprint should be: 3464C5BDD2BE7F2B6112E2F08E9C0024E33D9FE0
7. Check the Use for Login check box and click OK to add the AAD issuer



Step 6 – Configure the Claim Mappings in K2

Claim mappings are used to identify the incoming claims and map them to the appropriate K2 security label.

1. Open the Manage Claims form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{YourServer}/Runtime/Form/Manage+Claims/
2. Click New on the Security Label view
3. Select the Security Provider you configured in Step 4 from the Security Label drop down
4. Select the Issuer you configured in Step 5 from the Issuer drop down
5. For the Identity Provider Original Issuer textbox enter the Original Issuer value for AAD (Example: https://sts.windows.net/{YourTenantID}/)
6. For the Identity Provider Claim Type textbox enter http://schemas.microsoft.com/identity/claims/tenantid
7. For the Identity Provider Claim Value textbox enter your Tenant ID for AAD
8. For the Identity Original Issuer textbox enter the Original Issuer value for AAD (Example: https://sts.windows.net/{YourTenantID}/)
9. For the Identity Claim Type textbox enter the Claim Type value for AAD (Example: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name)
10. Leave the Identity Claim Value textbox empty as this claim will be different for each user that logs in at runtime.
11. Click OK to add the mapping.

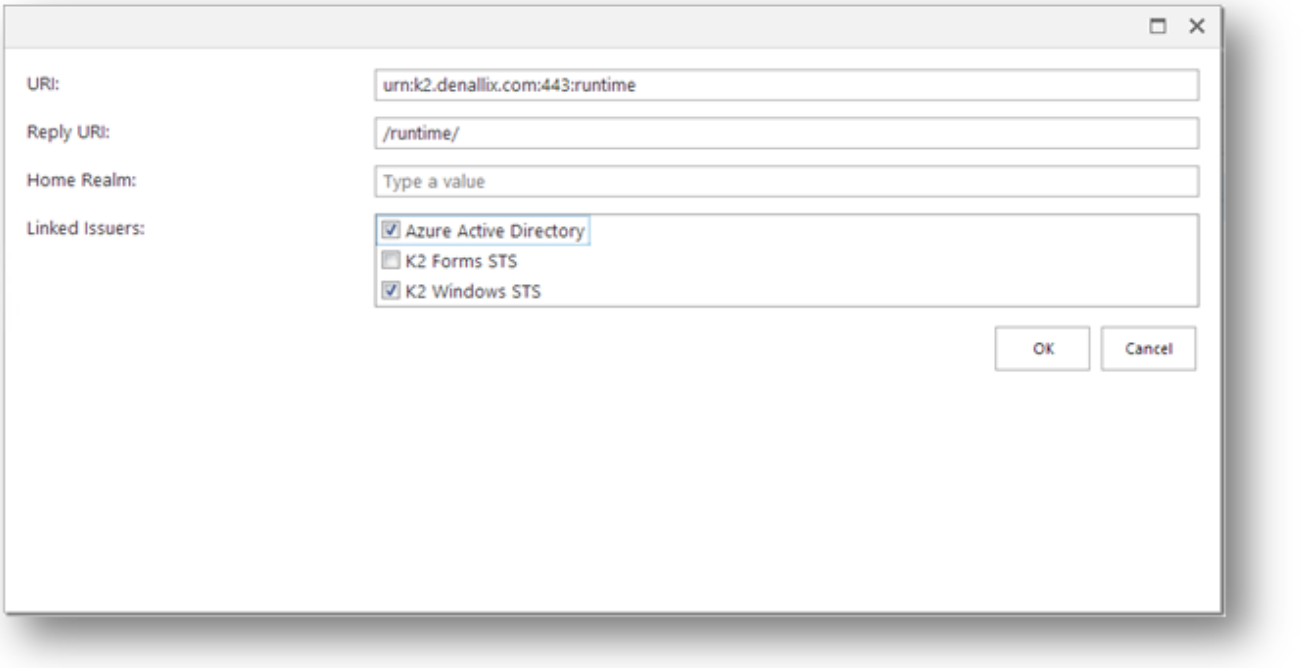


Step 7 - Configure the Realm and Audience URIs to Issuer mapping

The Realm is the unique value that associates the SmartForms site with the claims authentication options. Audience URIs are the actual URLs that will be used to access the SmartForms site. Additional Audience URIs can be specified for a single Realm. For example if you use https://k2.denallix.com/Runtime and http://dlx:81/Runtime to access the SmartForms site you will need both URLs registered as Audience URIs.

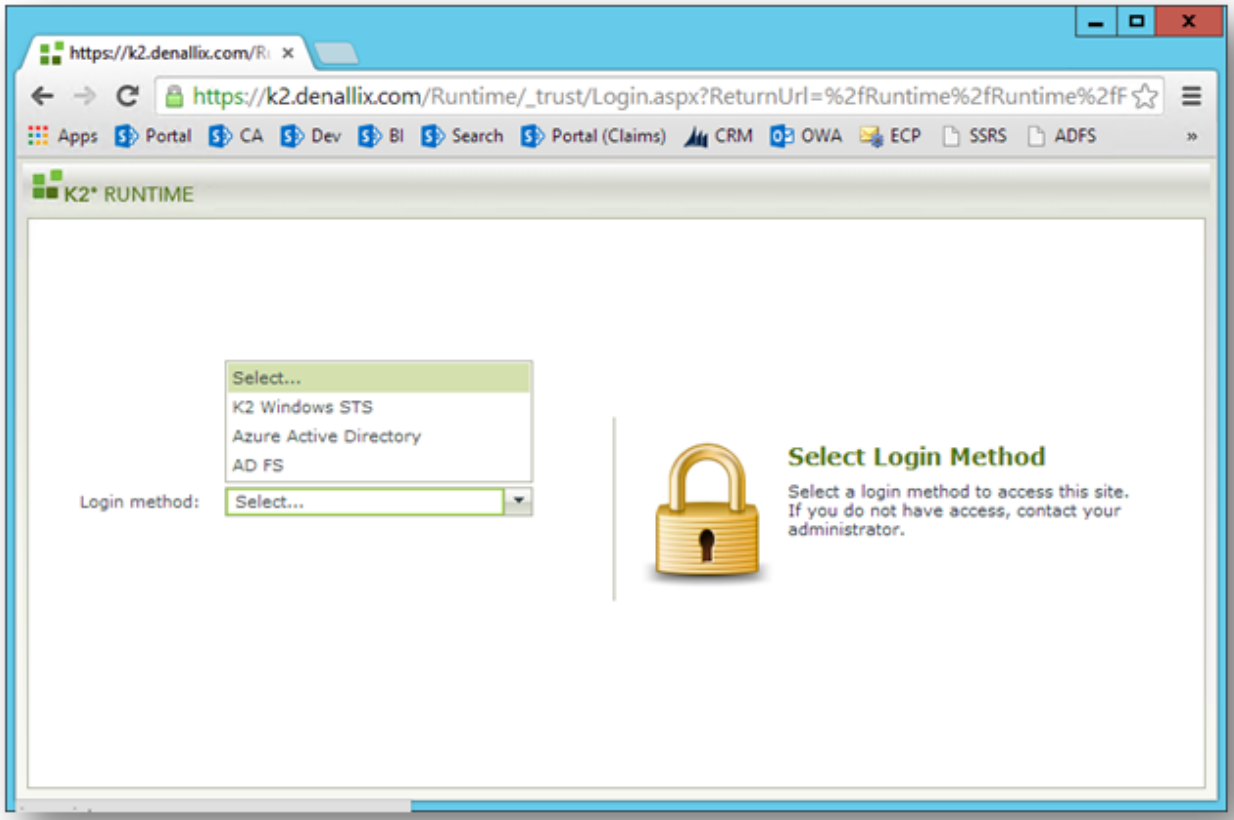
1. Open the Manage Site Realms form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{YourServer}/Runtime/Form/Manage+Site+Realms/

2. The list of Realms should be pre-configured with your Runtime, Designer, and View Flow realms and Audience URIs. For each realm that you wish to enable AAD authentication for follow the steps below
    a. Select the desired Realm and click edit
    b. In the Edit Realm dialog use the checkbox list control to select the desired issuers for the realm.
    c. Click OK



Step 8 - Navigate to the SmartForms site.

If AAD is the only authentication configured in the Realm to Issuer mapping then you should be redirected to the Azure Active Directory login screen. If there are other authentication modes configured then you will see a page with a dropdown that lets you select the authentication method that you want to use. Authentication that you have previously been using to access your smartforms sites will likely be cached by your browser. To enable you to log in using AAD you will first need to clear your browser cache.



K2 Multi-Auth: Configure SmartForms for Active Directory Federation Services (AD FS)

With the introduction of K2 blackpearl 4.6.7 and K2 smartforms 1.0.6 it is now possible to configure a single K2 smartforms site with multiple authentication providers. This document outlines the configuration steps necessary for enabling Active Directory Federation Services (AD FS) for K2 smartforms sites.

Configuration

Prerequisites

The following prerequisites are required for configuring SmartForms for AD FS:

- K2 blackpearl 4.6.7 (Refer to the SmartForms and Claims document to configure SmartForms for AD FS in prior K2 blackpearl releases)
- Microsoft .NET Framework 4.5
- Token signing certificate from your Identity Provider
- Active Directory Federation Service (AD FS) installed and configured
- SSL-enable the web site that hosts the K2 smartforms virtual directories

High Level Configuration Steps

These high-level steps are provided for those familiar with configuring claims integration. For a detailed guide, see the Detailed Steps topic below.

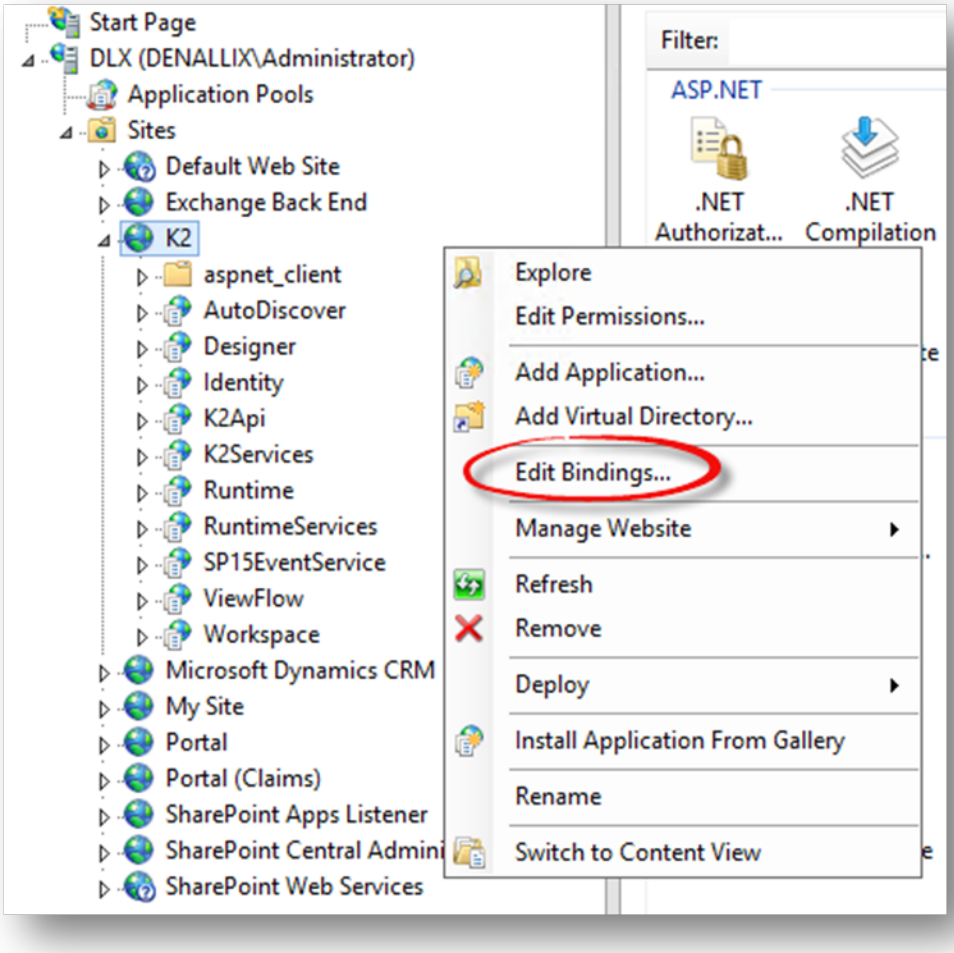
1. SSL-enable the web site that hosts the K2 smartforms virtual directories
2. Install the Identity Provider Certificate on the K2 server
3. Configure the K2 Security Provider
4. Configure the Claim Issuer in K2
5. Configure the Claim Mappings in K2
6. Configure the Realm to Issuer Mappings in K2
7. Configure K2 as a Relying Party Trust in ADFS for each K2 smartforms site

Detailed Steps

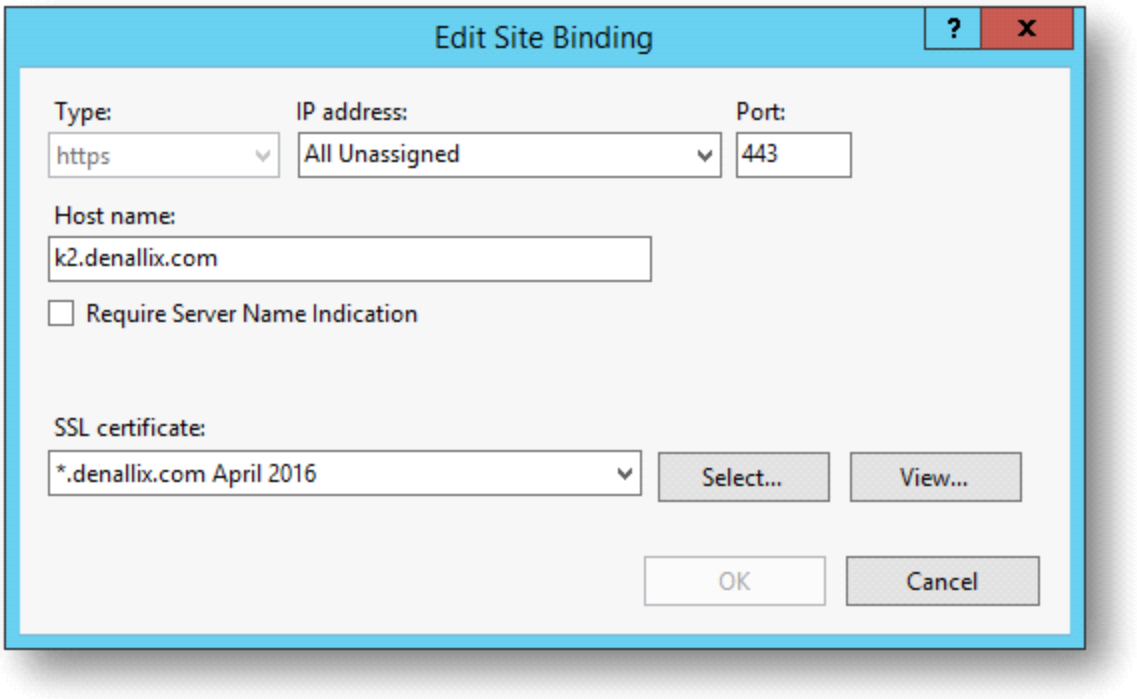
Step 1 – Create an SSL-enabled site for SmartForms

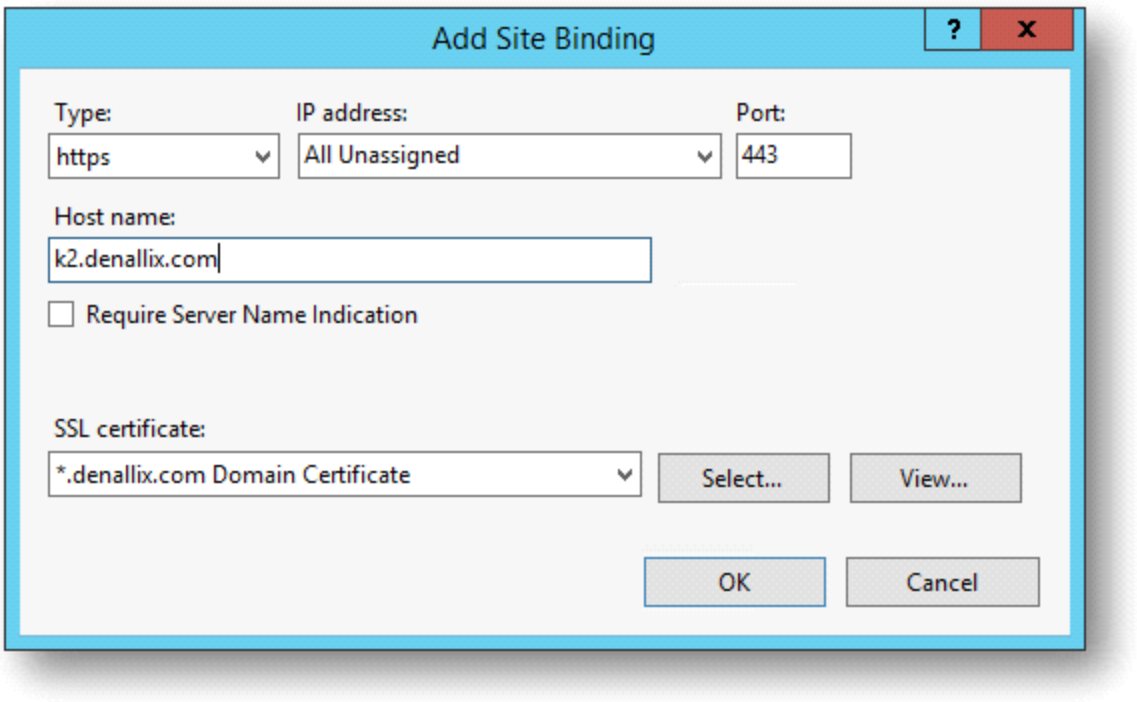
When configuring SmartForms to work with AD FS it is required to have your SmartForms site enabled for SSL.

1. Open Internet Information Services (IIS) Manager
2. In this example the K2 site is used. Right click the K2 site and select Edit bindings



3. Click Add
4. Select https from the Type drop down list and type a new number into the Port field
5. Select the certificate used for your site. In this example it is the *.denallix.com April 2015 Certificate, which was purchased from a valid Certificate Authority.
   Note -When working with systems like AD FS that are internal to the organization, Self-signed and Domain Certificates can work in the place of purchased certificates. If you plan to also integrate with an online system like Azure Active Directory you will need to use a purchased certificate.



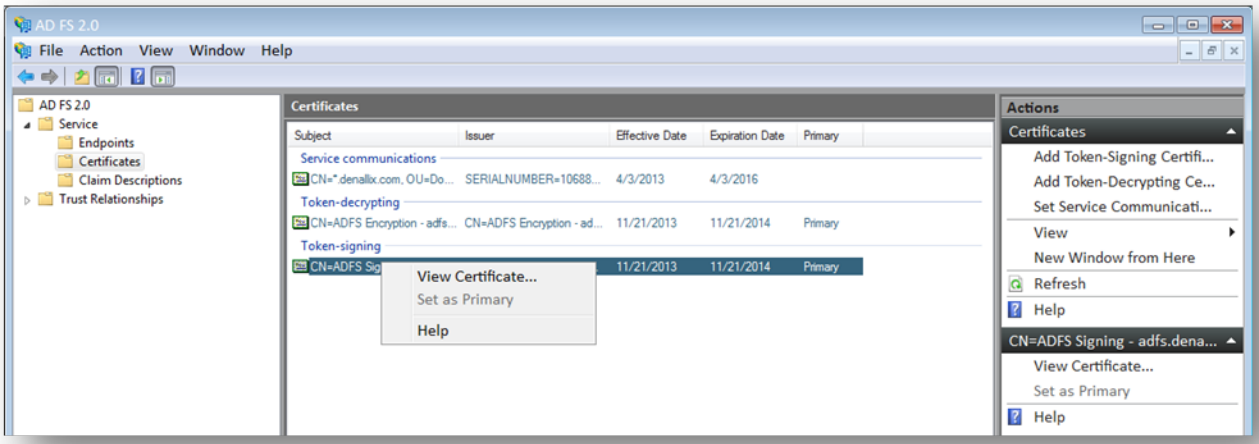


6. To make sure that the new HTTPS configuration is updated properly in the K2 configuration rerun the K2 black-pearl Setup Manager and the K2 SmartForms Setup Manager.

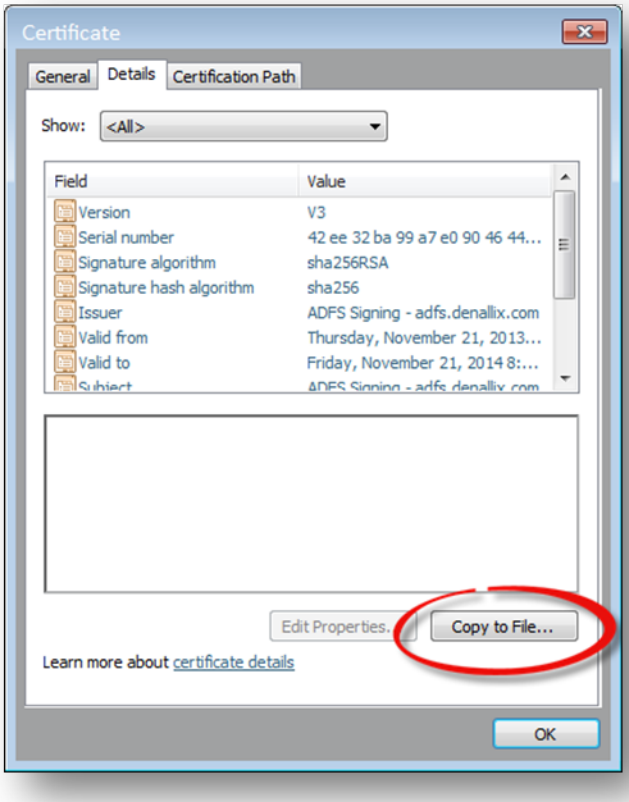
Step 2 – Install the Identity Provider Certificate on the K2 server

The Token-signing certificate from ADFS must be installed in the Trusted Root Certification Authorities location on the K2 server.

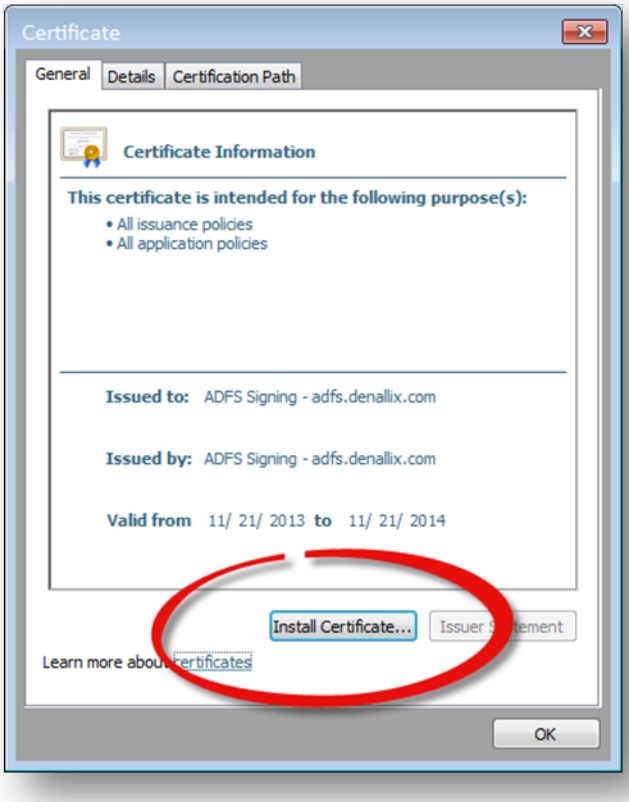
1. Open the AD FS Management Console
2. Expand the Service Node
3. Select the Certificates Node
4. Right-click the Token-signing certificate and select View Certificate



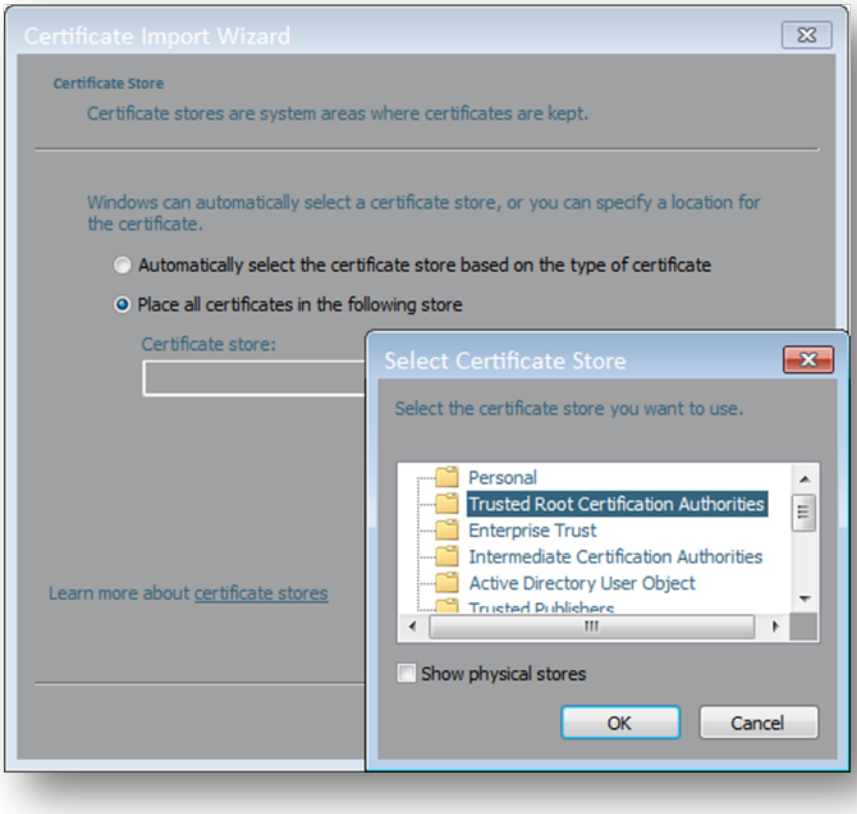
5. Select the Details tab and click the Copy to File button



6. Walk through the Certificate Export Wizard, creating a DER encoded binary X.509 certificate file
7. Copy the .CER file to the K2 server
8. On the K2 server double-click the .CER file and click the Install Certificate button on the General Tab



9. Select the Place all certificates in the following store option on the Certificate Import Wizard and choose the Trusted Root Certificate Authorities store



10. Click OK, Next and then Finish to complete the wizard.

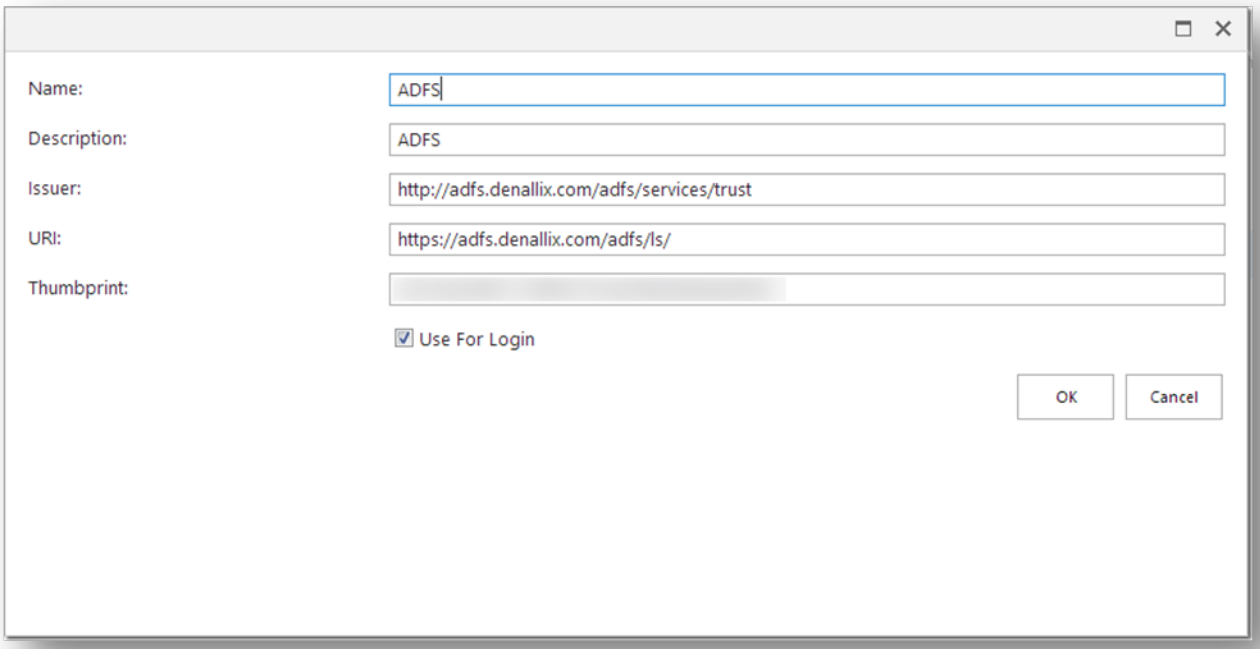
Step 3 – Configure the K2 Security Provider

In order for K2 to authenticate the users from AD FS a K2 Security Provider needs to be installed and configured. If AD FS is configured to use Active Directory or another LDAP based system as its Attribute Store, you can configure the K2 LDAP Security Provider by following the steps in the K2 blackpearl Getting Started Guide.

Step 4 – Configure the Claim Issuer in K2

A Claim Issuer is required for K2 to map the STS signing certificate with the security tokens.

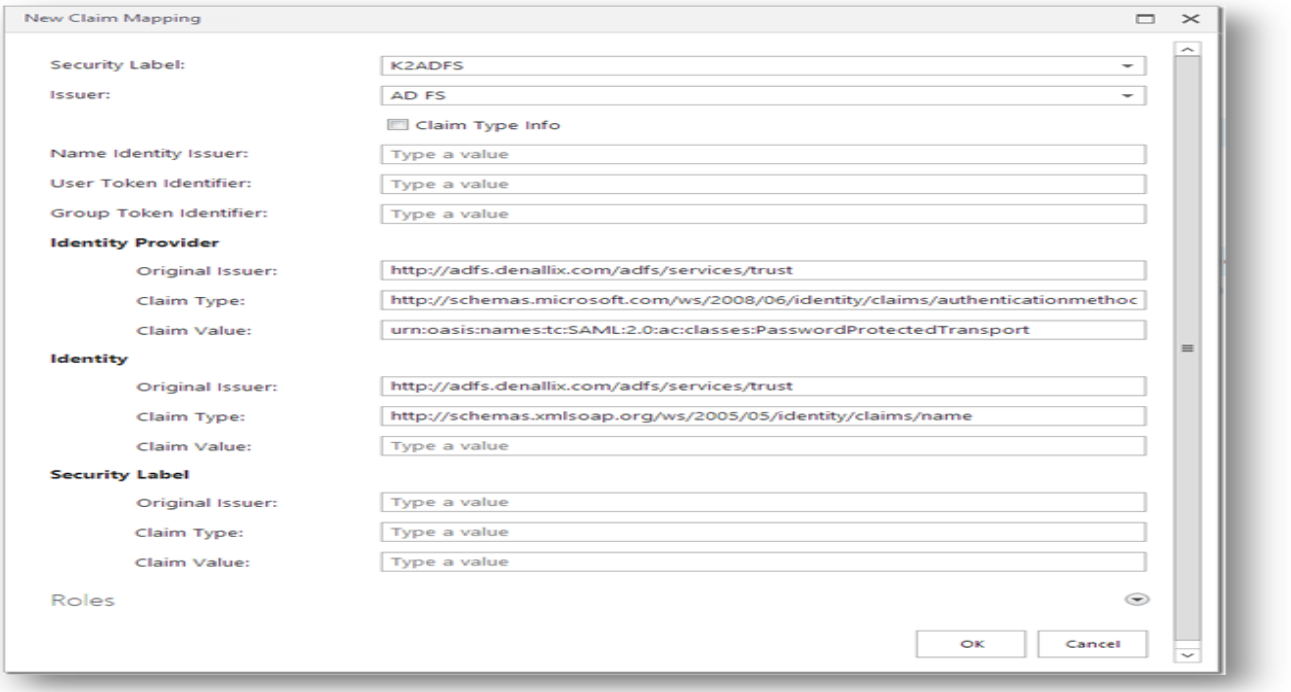
1. Open the Manage Issuers form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Issuers/
2. Click New
3. Enter a Name and Description for the AD FS Issuer
4. Enter the Token Issuer endpoint for AD FS in the Issuer textbox (Example: http://adfs.denallix.com/adfs/services/trust)
5. Enter the Login URL for ADFS in the URI textbox (Example: https://adfs.denallix.com/adfs/ls/)
6. Enter the Thumbprint of the Token-signing certificate for AD FS
7. Check the Use for Login check box and click OK to add the AD FS issuer



Step 5 – Configure the Claim Mappings in K2

Claim mappings are used to identify the incoming claims and map them to the appropriate K2 security label.

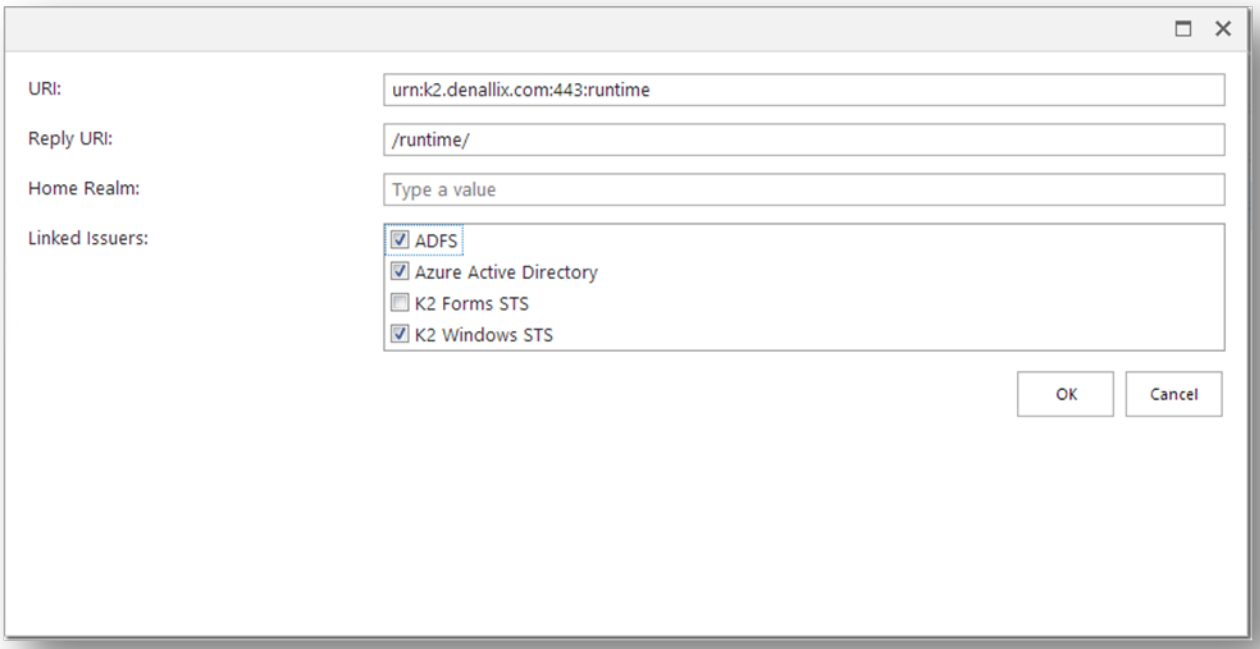
1. Open the Manage Claims form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Claims/
2. Click New on the Security Label view
3. Select the Security Provider you configured in Step 3 from the Security Label drop down
4. Select the Issuer you configured in Step 4 from the Issuer drop down
5. For the Identity Provider Original Issuer textbox enter the Original Issuer value for AD FS (Example: http://adfs.denallix.com/adfs/services/trust)
6. For the Identity Provider Claim Type textbox enter the Claim Type value for AD FS (Example: http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod)
7. For the Identity Provider Claim Value textbox enter the Authentication Method Claim Value for AD FS (Example: urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport)
8. For the Identity Original Issuer textbox enter the Original Issuer value for AD FS (Example: http://adfs.denallix.com/adfs/services/trust)
9. For the Identity Claim Type textbox enter the Claim Type value for AD FS (Example: http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name)
10. Leave the Identity Claim Value textbox empty as this claim will be different for each user that logs in at runtime.
11. Click OK to add the mapping.



Step 6 – Configure the Realm to Issuer Mappings in K2

The Realm is the unique value that associates the SmartForms site with the claims authentication options. Audience URIs are the actual URLs that will be used to access the SmartForms site. Additional Audience URIs can be specified for a single Realm. For example if you use https://k2.denallix.com/Runtime and http://dlx:81/Runtime to access the SmartForms site you will need both URLs registered as Audience URIs.

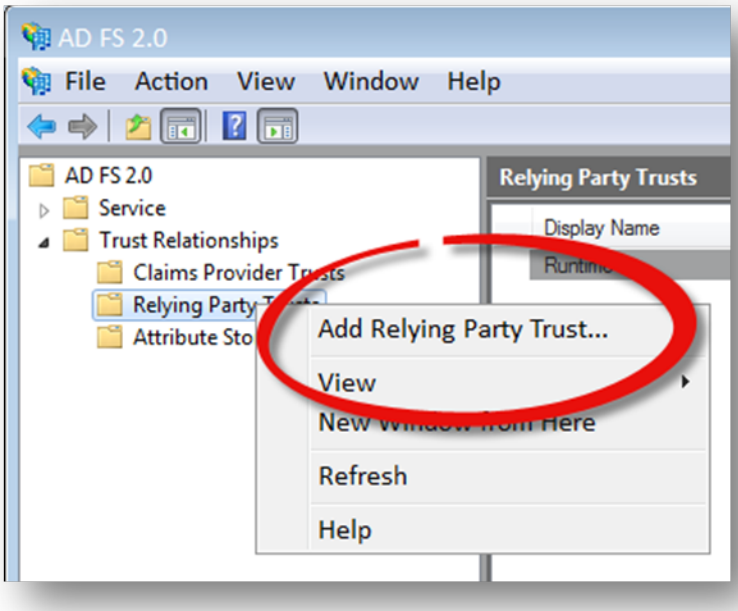
1. Open the Manage Site Realms form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Site+Realms/
2. The list of Realms should be pre-configured with your Runtime, Designer, and View Flow realms and Audience URIs. For each realm that you wish to enable AD FS authentication for follow the steps below:
    a. Select the desired Realm and click edit
    b. In the Edit Realm dialog use the checkbox control to select the desired issuers to map to the realm.
    c. Click OK



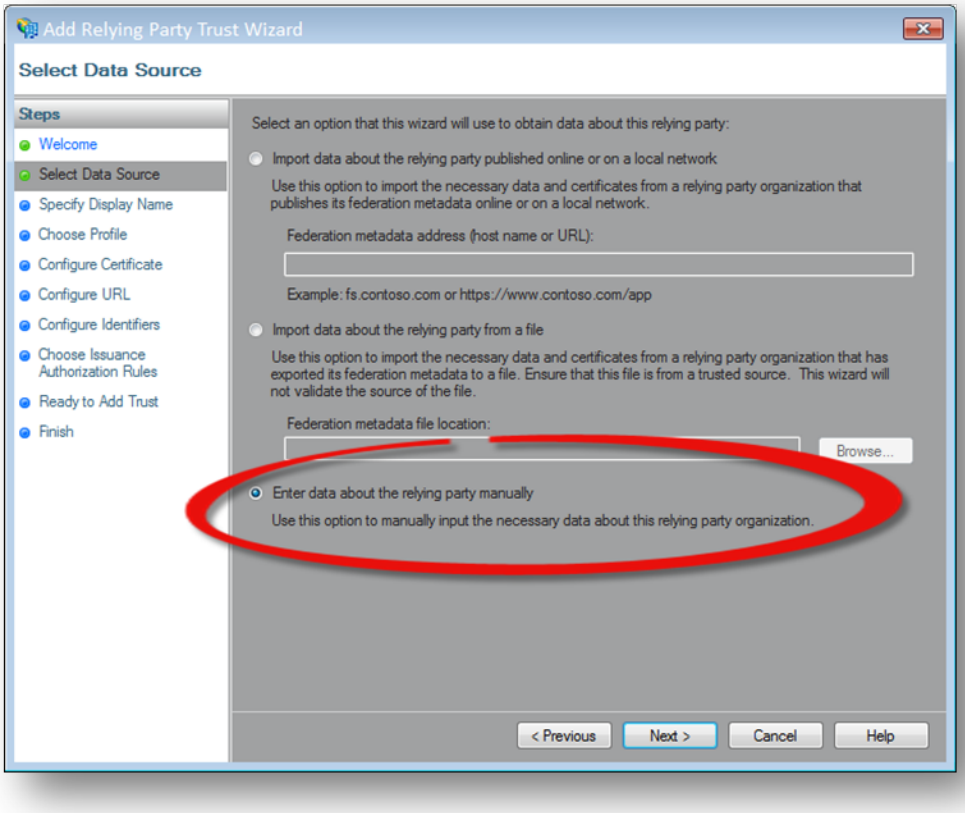
Step 7 - Configure K2 as a Relying Party Trust in AD FS for each K2 smartforms site

The following steps are required to add a relying party trust to the token issuer. This must be done for each site that was configured above.

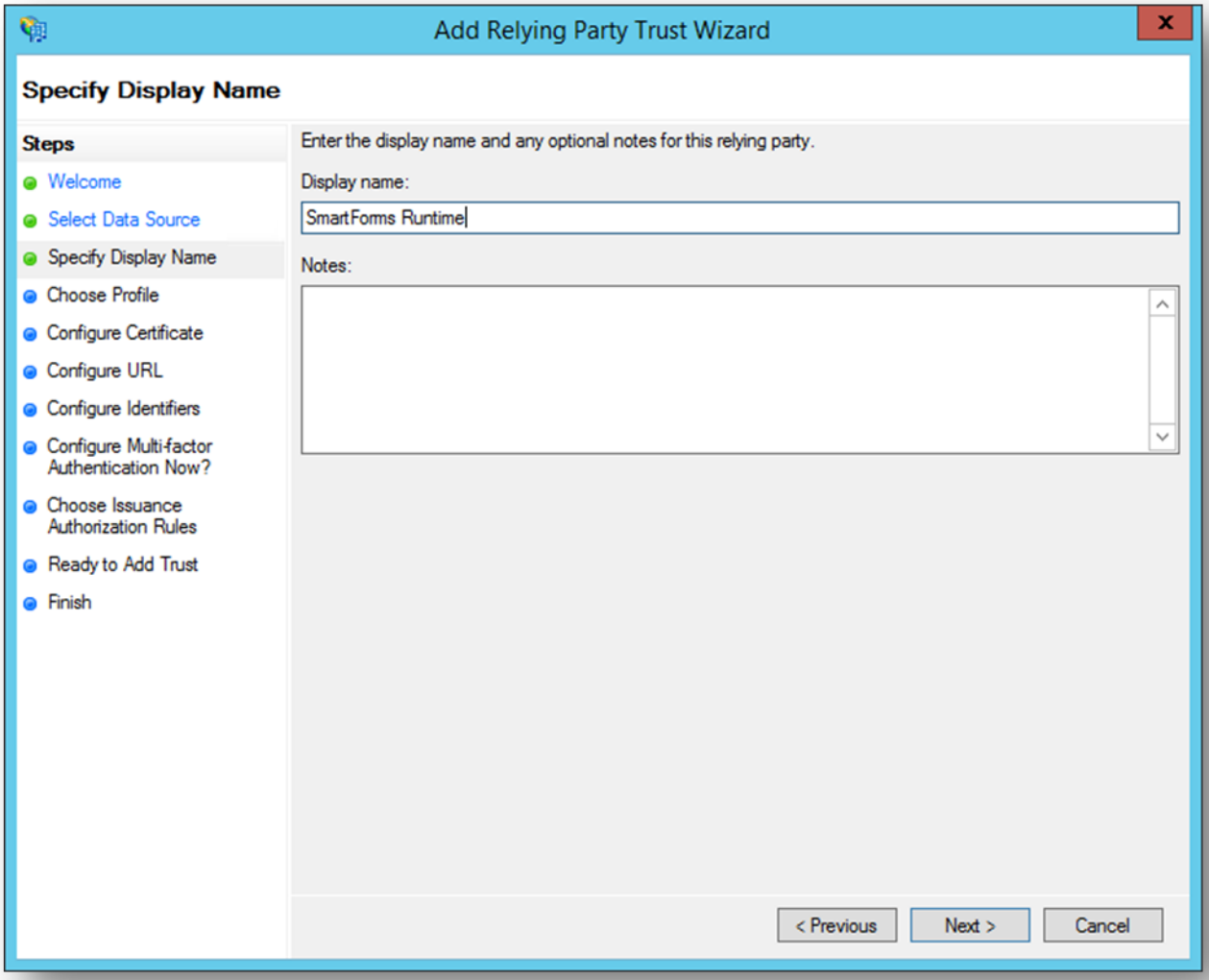
1. Launch the AD FS Management console
2. Expand the Trust Relationships node
3. Right-click on the Relying Party Trusts node and select Add Relying Party Trust...



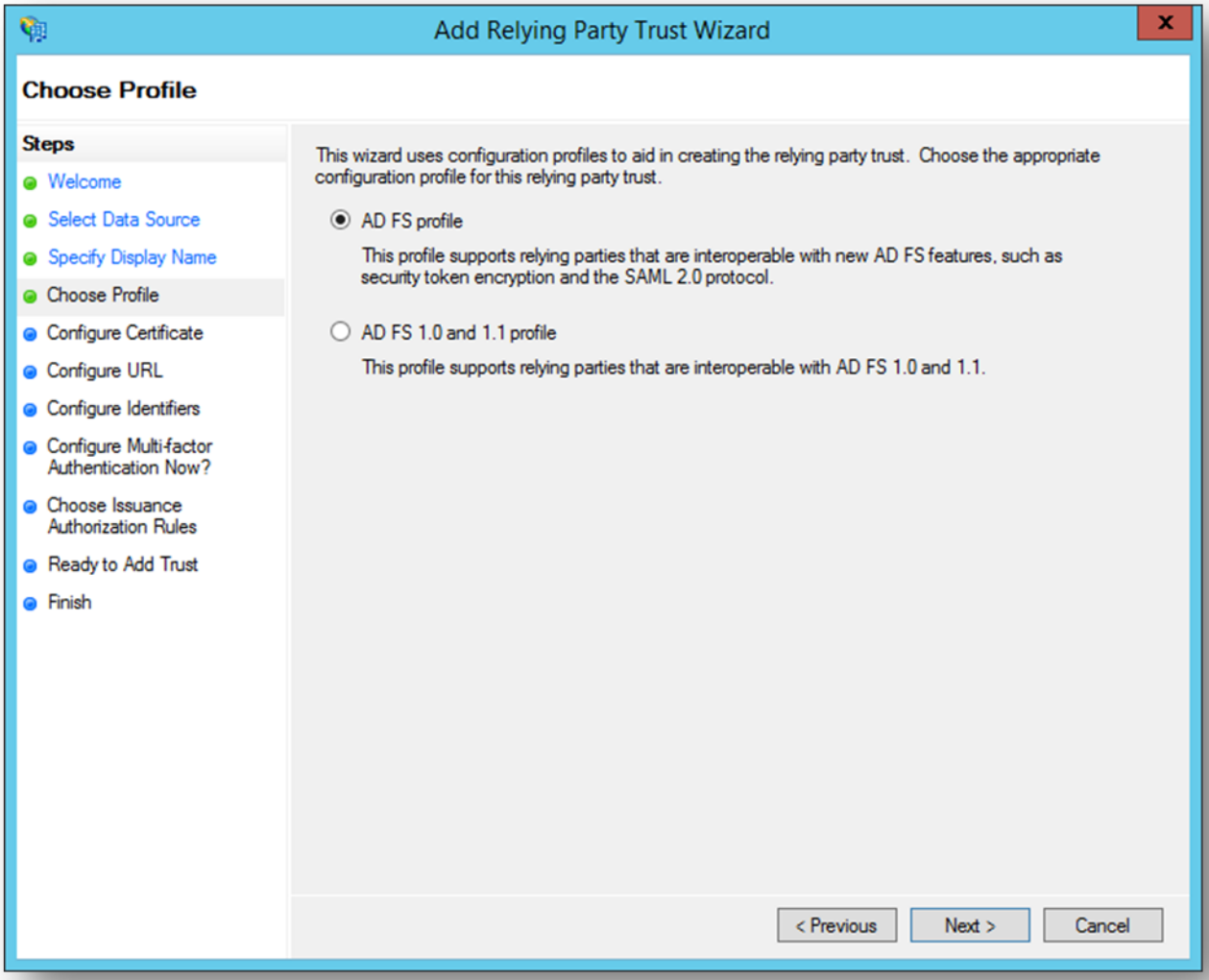
4. Click Start
5. Select Enter data about the relying party manually and click next



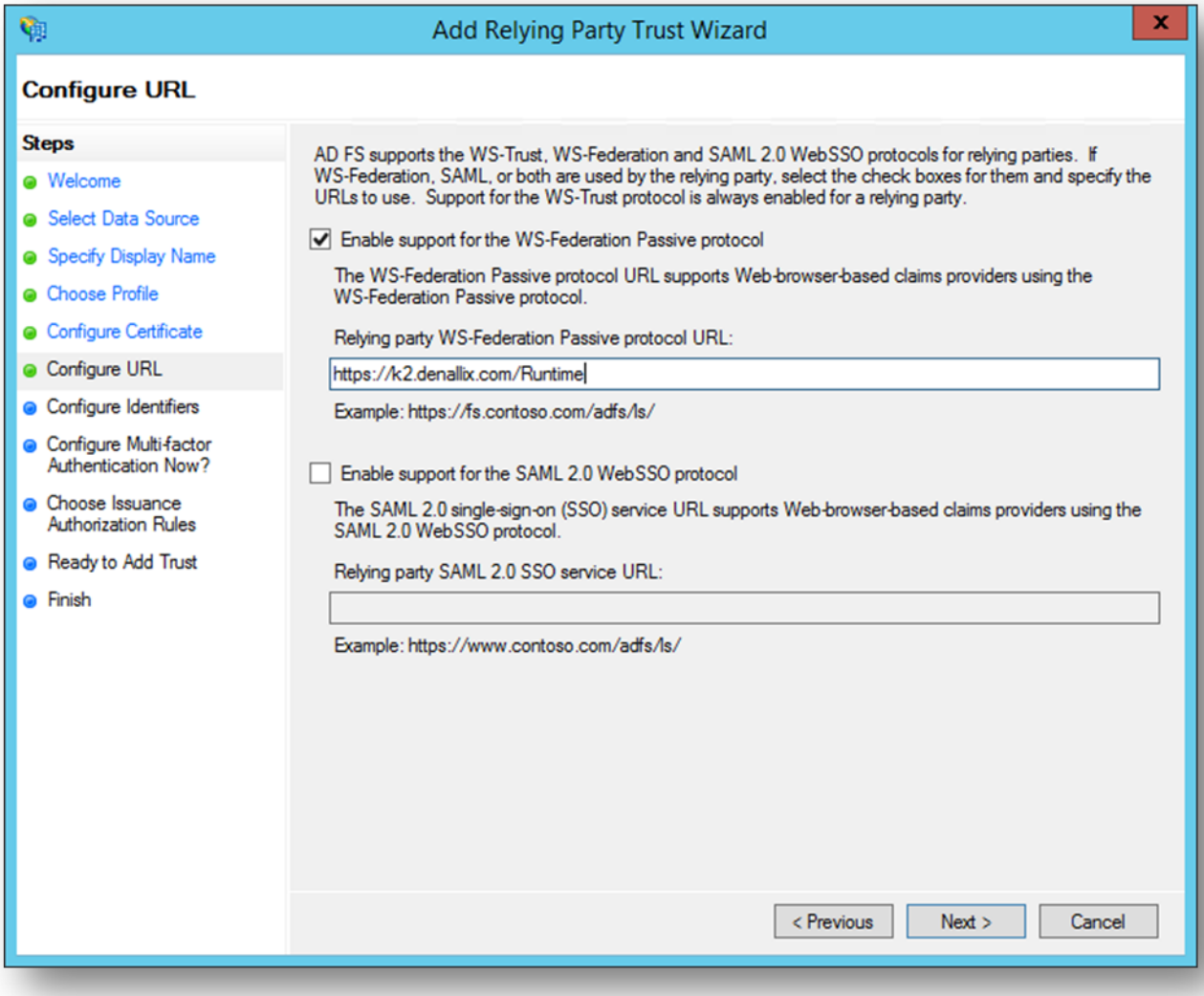
6. Enter a display name and description and click Next



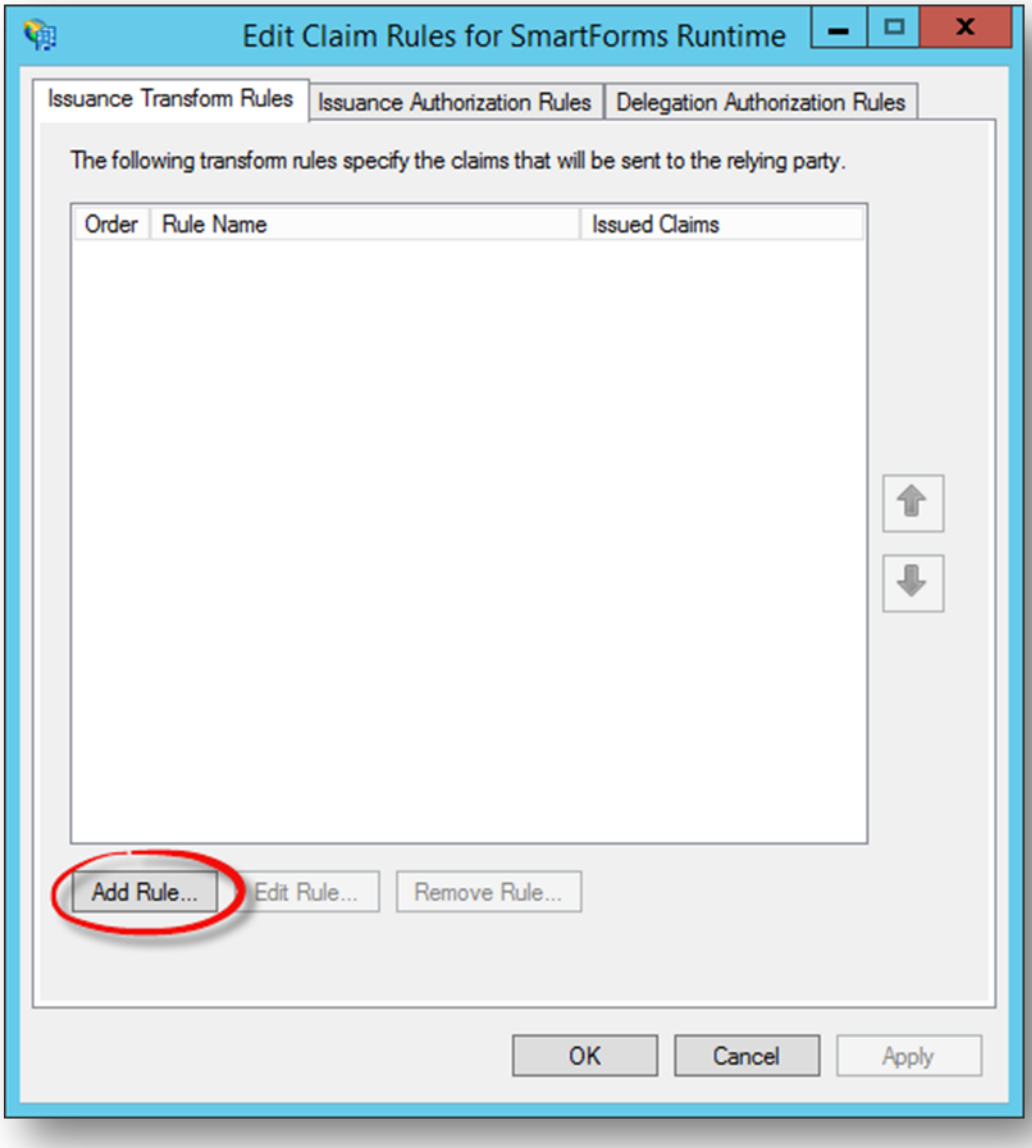
7. Select AD FS profile and click Next



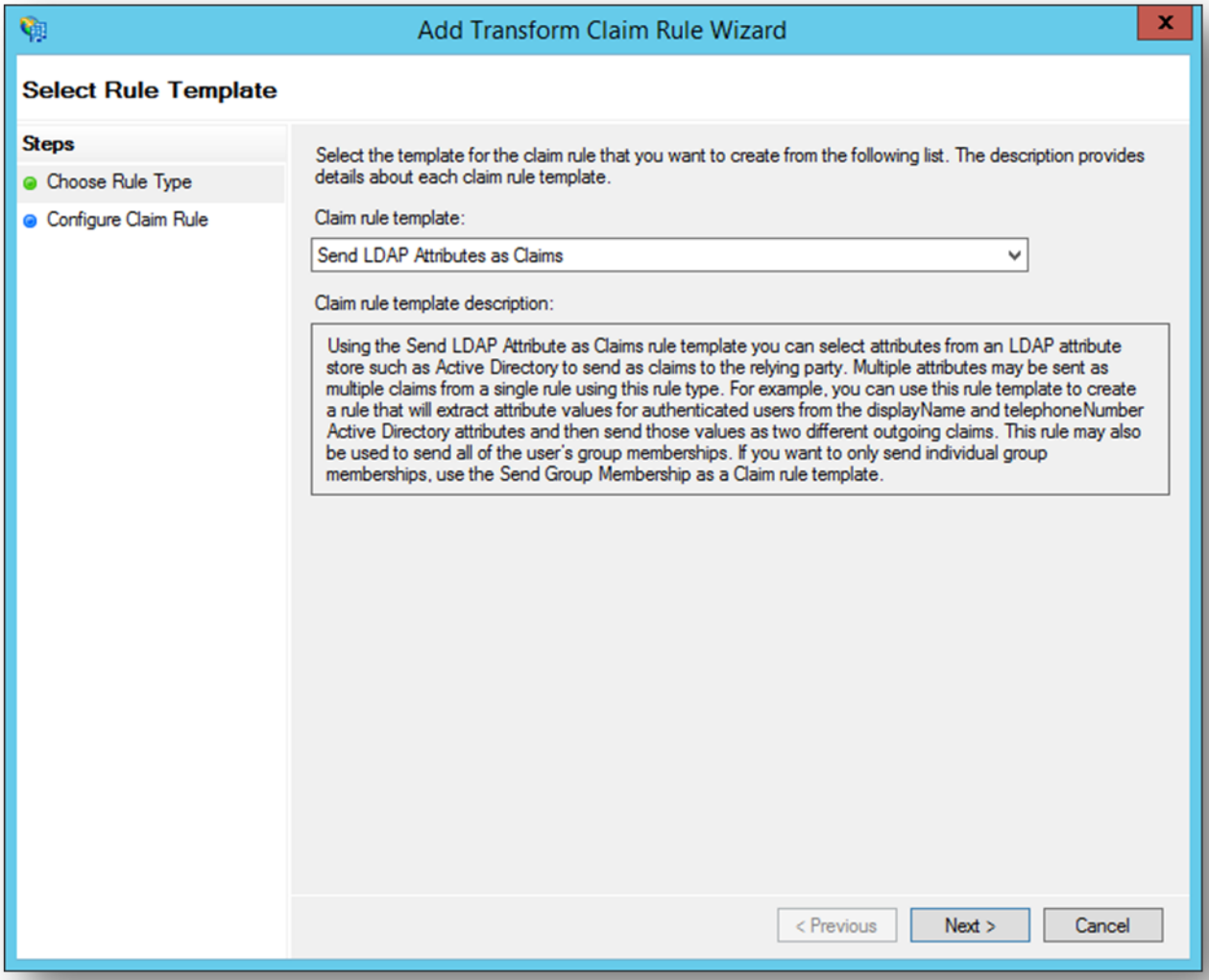
8. Optionally specify a token encryption certificate. Since SSL is being used for the site it is not necessary to also encrypt the token unless desired. Click next
9. Check the Enable support for the WS-Federation Passive protocol option
10. Enter the SmartForms site URL (Example: https://k2.denallix.com/Runtime/) and click next



11. Click next on the Configure Identifiers page
12. Click next on the Multi-Factor authentication page
13. Select the desired Issuance authorization rules option and click next
14. Click next and Close
15. The Edit Claim Rules dialog will open
16. On the Issuance Transform Rules tab, click Add Rule…

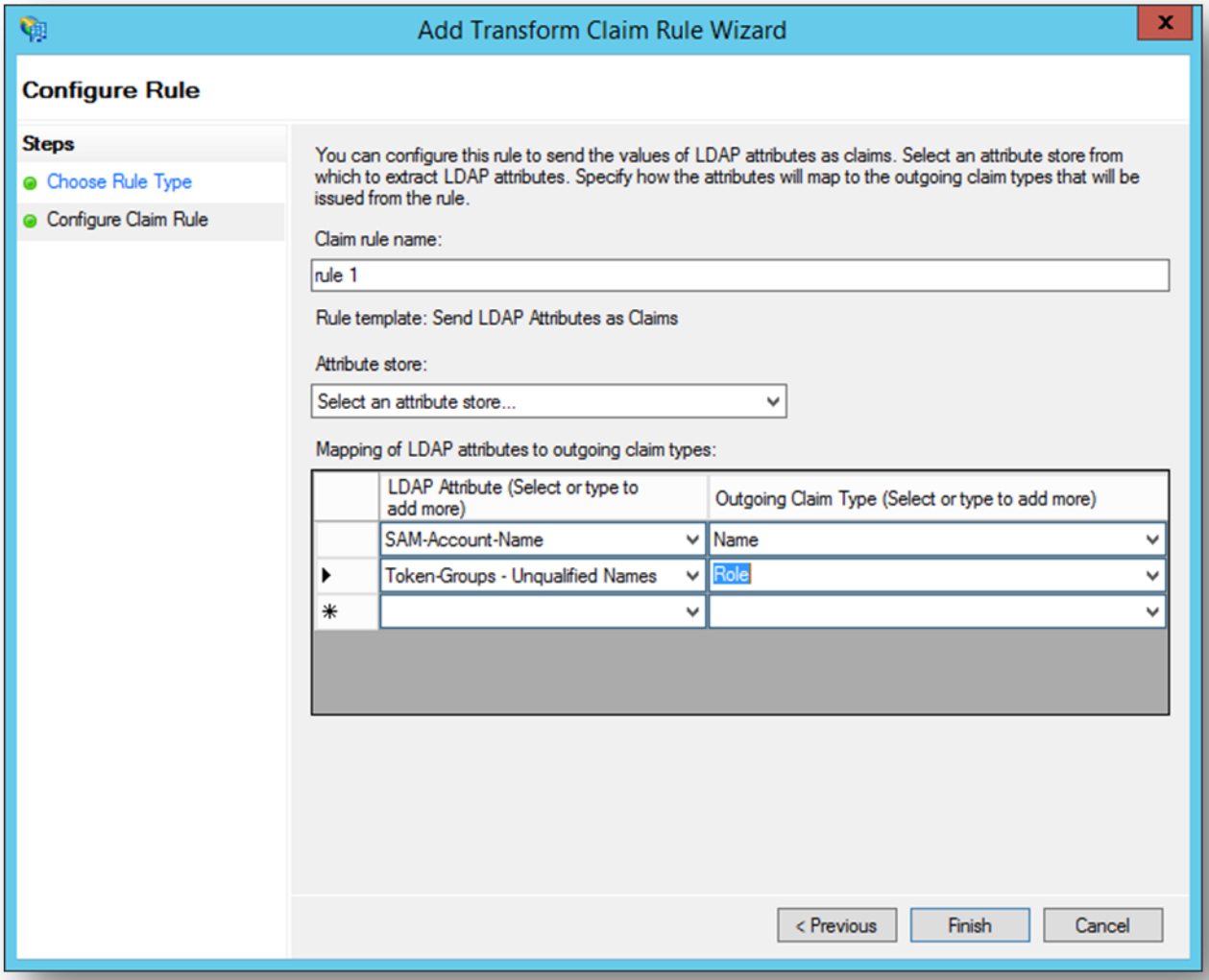


17. Select Send LDAP Attributes as Claims and click next



18. Supply a name for the rule and select Active Directory as the Attribute store
19. Configure the required and optional claim mappings:

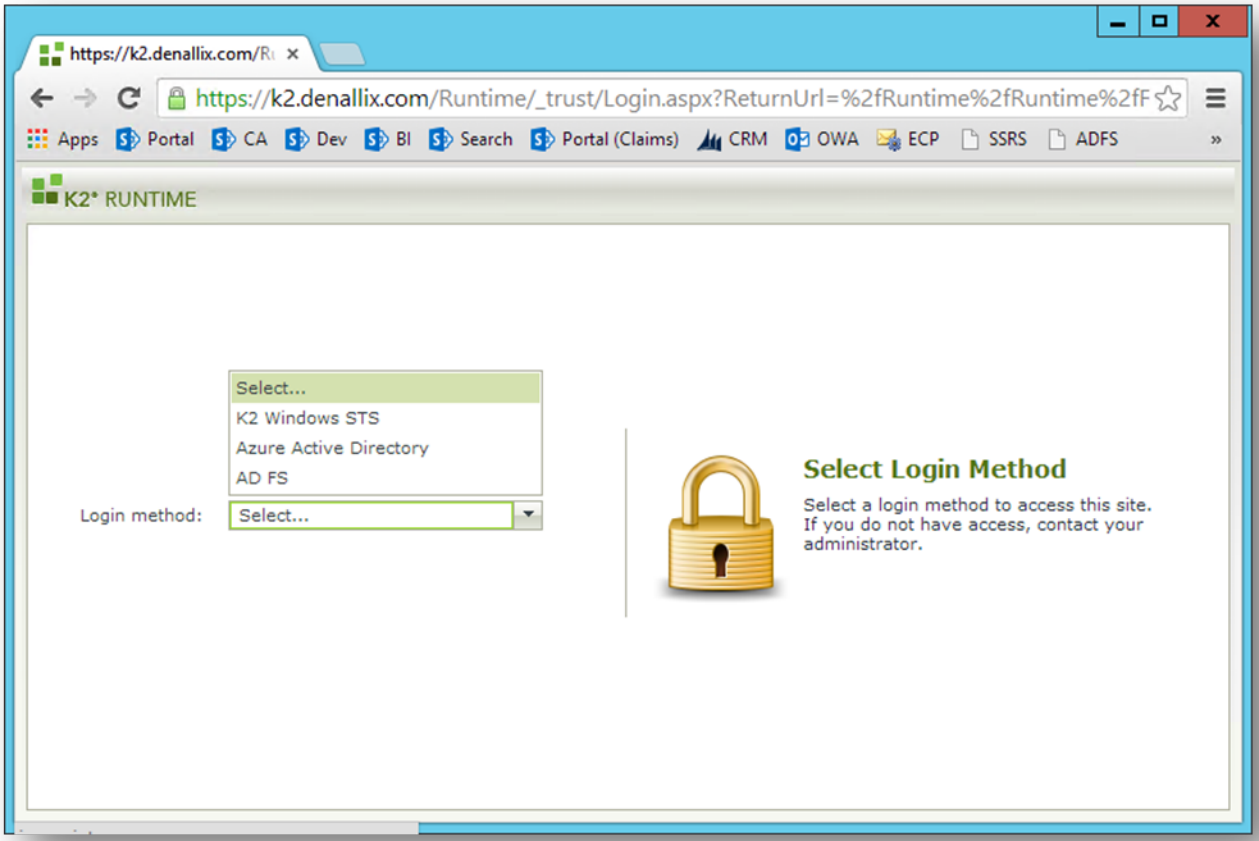
LDAP Attribute	Outgoing Claim Type
SAM-Account-Name	Name
Token-Groups - Unqualified Names	Role



20. Click Finish
21. Click OK to complete the Claim Rule mappings
22. Restart the K2 Host Server and navigate to a site that you configured for ADFS authentication. You may also need to clear your browser cache in order to clear any cached authentication.

Step 8 - Navigate to the SmartForms site.

If ADFS is the only authentication configured in the Realm to Issuer mapping then you should be redirected to the ADFS log in screen. If there are other authentication modes configured then you will see a page with a dropdown that lets you select the authentication method that you want to use. Authentication that you have previously been using to access your K2 smartforms sites will likely be cached by your browser. To enable you to login using AD FS you will first need to clear your browser cache.



K2 Multi-Auth: Configure SmartForms for SQL Server User Manager (SQLUM)

With the introduction of K2 blackpearl 4.6.7 and K2 smartforms 1.0.6 it is now possible to configure a single K2 smartforms site with multiple authentication providers. This document outlines the configuration steps necessary for enabling SQL Server User Manager (SQLUM) for K2 smartforms sites.

Configuration

Prerequisites

The following prerequisites are required for configuring SmartForms for SQLUM:

- K2 blackpearl 4.6.7 (Refer to the K2 blackpearl Getting Started Guide to configure SmartForms for SQLUM in prior K2 blackpearl releases)
- Microsoft .NET Framework 4.5

High Level Configuration Steps

These high-level steps are provided for those familiar with configuring claims integration. For a detailed guide, see the Detailed Steps topic below.

1. [Optional] SSL-enable the web site that hosts the K2 smartforms virtual directories
2. Configure the K2 SQLUM Security Provider
3. Configure the K2 Forms STS for login

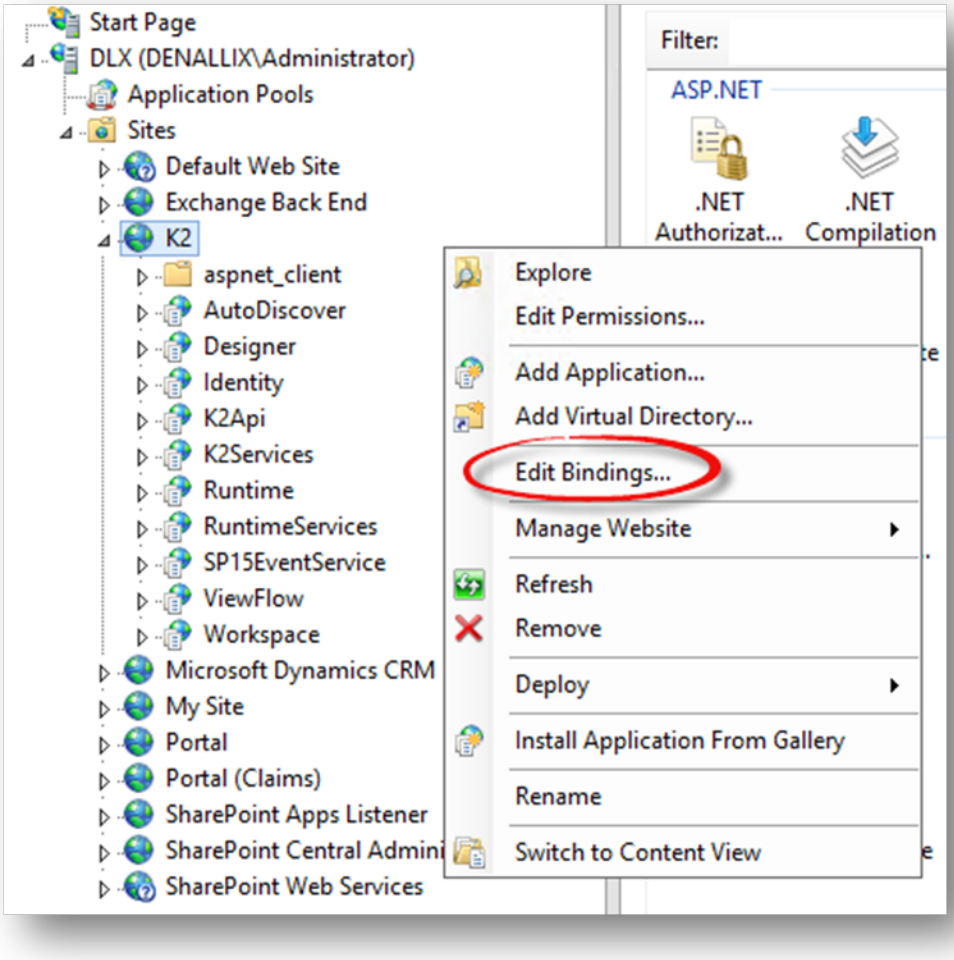
4. Configure the Claim Mappings in K2
5. Configure the Realm to Issuer Mappings in K2

Detailed Steps

Step 1 – Create an SSL-enabled site for SmartForms

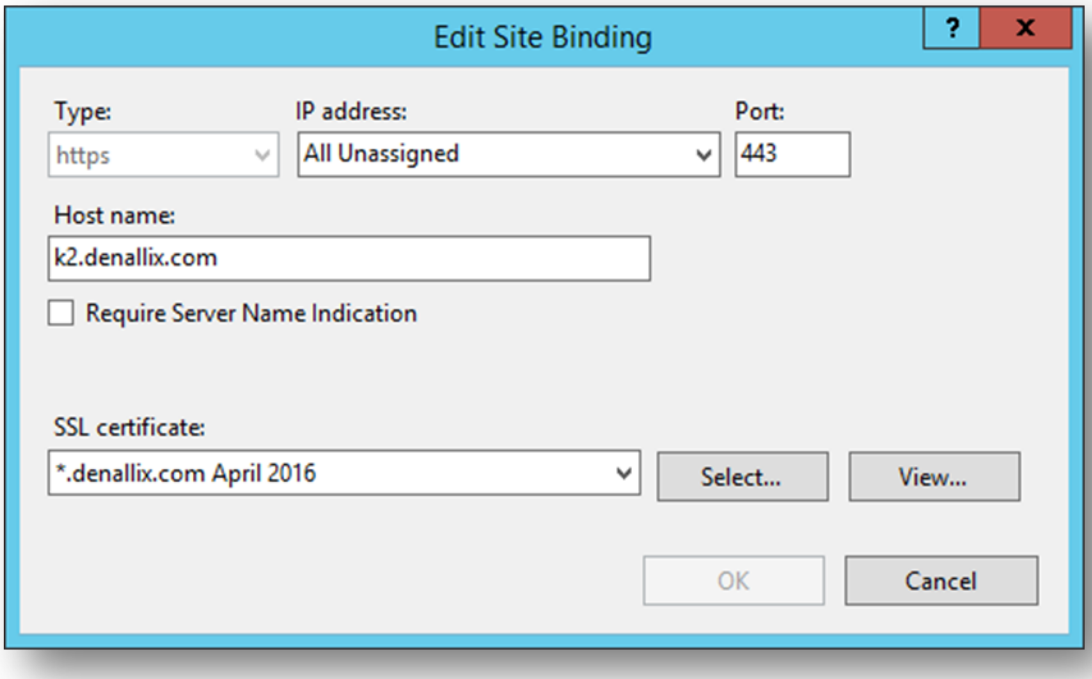
It is not required to have your SmartForms site enabled for SSL but it is highly recommended.

1. Open Internet Information Services (IIS) Manager
2. In this example the K2 site is used. Right click the K2 site and select Edit bindings



3. Click Add
4. Select https from the Type drop down list and type a new number into the Port field
5. Select the certificate used for your site. In this example it is the *.denallix.com April 2015 Certificate, which was purchased from a valid Certificate Authority.

NOTE: When working with systems that are internal to the organization, Self-signed and Domain Certificates can work in the place of purchased certificates. If you plan to also integrate with an online system like Azure Active Directory you will need to use a purchased certificate.



6. To make sure that the new HTTPS configuration is updated properly in the K2 configuration rerun the K2 black-pearl Setup Manager and the K2 SmartForms Setup Manager

Step 2 – Configure the K2 Security Provider

In order for K2 to authenticate the users against SQLUM a K2 Security Provider needs to be installed and configured.

1. Open SQL Manager and connect to the SQL Server where the K2 databases are hosted
2. Execute the following query to register the SQLUM security provider

```
USEK2
-- DECLARATIONS
 DECLARE@SecurityLabelName  NVARCHAR(20)= 'K2SQL'; -- Update as needed
 DECLARE@SecurityLabelID  UNIQUEIDENTIFIER ='8e8d5221-ee89-4cd7-99da-fcfcdf64abdb';
 DECLARE@AuthSecurityProviderID  UNIQUEIDENTIFIER ='fc1848e6-23f5-49d8-8c48-9f7b197c80b7';
 DECLARE@AuthInit XML=
 '<AuthInit>
<init>DLX,K2</init>
<login/>
<implementation assembly="SQLUM, Version=4.0.0.0, Culture=neutral, PublicKeyToken=16a2c5aaaa1b130d"
type="SQLUM.K2UserManager"/>
 </AuthInit>' -- XML configuration for the SQL provider, see K2 Help for more information on con-
figuration values
 DECLARE@RoleSecurityProviderID  UNIQUEIDENTIFIER =@AuthSecurityProviderID;
 DECLARE@RoleInit XML=
 '<roleprovider>
<init>DLX, K2</init>
<login/>
```

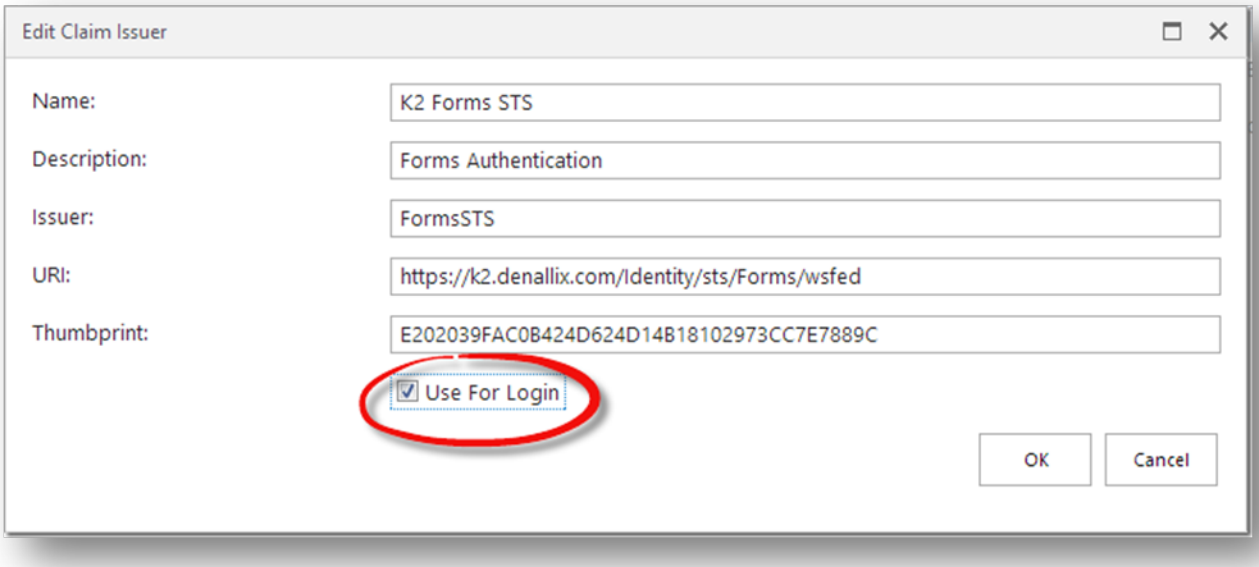
```
<implementation assembly ="SQLUM, Version=4.0.0.0, Culture=neutral, PublicKeyToken=16a2c5aaaa1b130d"
type ="SQLUM.K2UserManager"/>
 </roleprovider>' -- XML configuration for the SQL provider, see K2 Help for more information on con-
figuration values
 DECLARE@DefaultLabel BIT= NULL;--1 = true, NULL and 0 = false
-- UPDATE TABLES
 DELETEFROM [HostServer].[SecurityLabel]WHERE SecurityLabelName= @SecurityLabelName;
 INSERTINTO [HostServer].[SecurityLabel]VALUES (@SecurityLabelID,@SecurityLabelName,@AuthSe-
curityProviderID,@AuthInit,@RoleSecurityProviderID,@RoleInit,@DefaultLabel)
```

3. Restart the K2 blackpearl service

Step 3 - Configure the K2 Forms STS for Login

In order to be able to use the K2 Forms STS for login the Use for Login setting for the issuer needs to be set to true.

1. Open the Manage Issuers form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Issuers/
2. Select the K2 Forms STS and click edit
3. Check the Use for Login checkbox and click OK

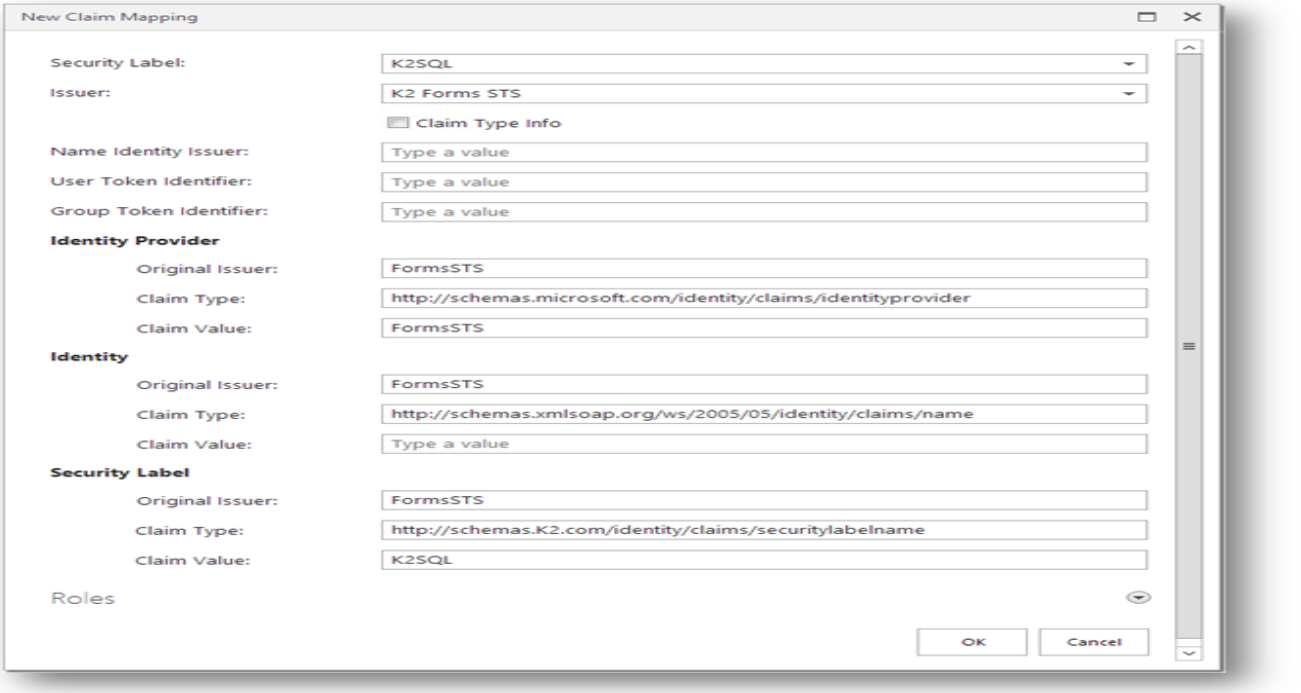


Step 4 – Configure the Claim Mappings in K2

Claim mappings are used to identify the incoming claims and map them to the appropriate K2 security label.

1. Open the Manage Claims form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Claims/
2. Click New on the Security Label view
3. Select the Security Provider you configured in Step 2 from the Security Label drop down

4. Select K2 Forms STS from the Issuer drop down
5. For the Identity Provider Original Issuer textbox enter FormsSTS
6. For the Identity Provider Claim Type textbox enter http://schemas.microsoft.com/identity/claims/identityprovider
7. For the Identity Provider Claim Value textbox enter FormsSTS
8. For the Identity Original Issuer textbox enter FormsSTS
9. For the Identity Claim Type textbox enter http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name
10. Leave the Identity Claim Value textbox empty as this claim will be different for each user at runtime.
11. Click OK to add the mapping.



Step 5 – Configure the Realm to Issuer Mappings in K2

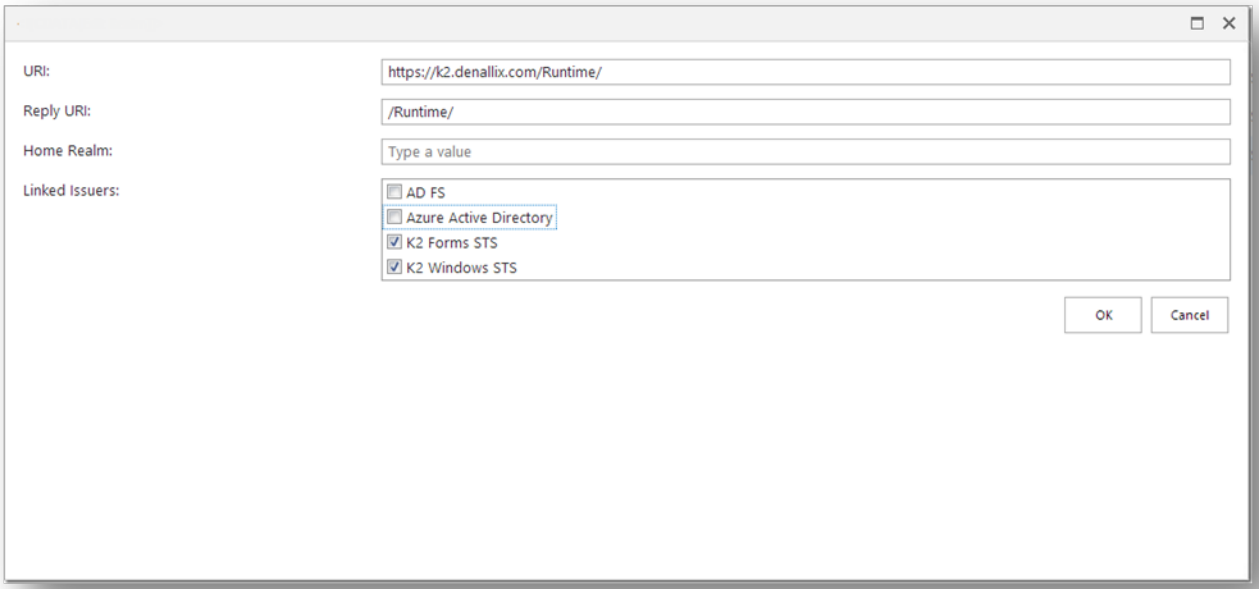
The Realm is the unique value that associates the SmartForms site with the claims authentication options. Audience URIs are the actual URLs that will be used to access the SmartForms site. Additional Audience URIs can be specified for a single Realm. For example if you use https://k2.denallix.com/Runtime and http://dlx:81/Runtime to access the SmartForms site you will need both URLs registered as Audience URIs.

1. Open the Manage Site Realms form on your SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Site+Realms/
2. The list of Realms should be pre-configured with your Runtime, Designer, and View Flow realms and Audience URIs. For each realm that you wish to enable SQLUM authentication for follow the steps below:

a. Select the desired Realm and click edit

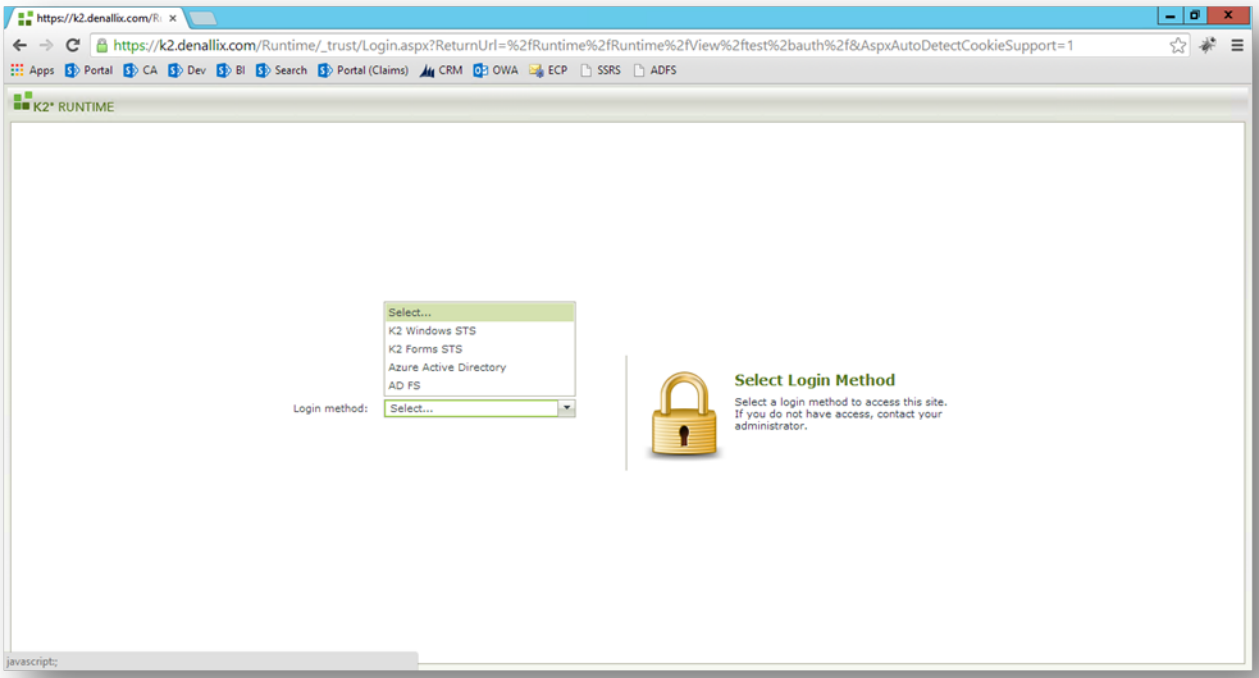
b. In the Edit Realm dialog use the checkbox list to select the desired issuers to map to the realm. For SQLUM select the K2 Forms STS

c. Click OK



Step 6 – Navigate to the SmartForms site

If K2 Forms STS is the only authentication configured in the Realm to Issuer mapping then you should be redirected to a login screen. If there are other authentication modes configured then you will see a page with a drop-down that lets you select the authentication type that you want to use. You may need to clear your browser cache to clear any currently cached credentials.



K2 Multi-Auth: Consolidation to Multi-Auth

With the introduction of K2 blackpearl 4.6.7 and K2 smartforms 1.0.6 it is now possible to configure a single K2 smartforms site with multiple authentication providers. In the past if you required multiple authentication methods for your SmartForms sites you needed to create a copy of your SmartForms site for each authentication method. This document describes the steps necessary to consolidate these multiple sites into a single site that uses multi-auth.

Should I consolidate?

Before you make the decision to consolidate your SmartForms sites to a single site, review the Introduction to Multi-Auth document to understand if a single Multi-Auth enabled site meets your needs. In some cases it may be desirable to continue having multiple sites rather than consolidate to a single site. If you prefer having separate sites, please refer to the Upgrading Copied SmartForms Sites to 1.0.6 document.

Configuration

High Level Steps

These high-level steps are provided for those familiar with upgrading K2 smartforms and configuring Multi-Auth. For a detailed guide, see the Detailed Steps topic below.

1. Upgrade your primary SmartForms site to 1.0.6
2. Configure your primary SmartForms site with each desired Authentication mode
3. Confirm that your Multi-Auth configuration is working
4. Remove the old secondary SmartForms sites

Detailed Steps

Step 1 - Upgrade your Primary SmartForms site to 1.0.6

The K2 smartforms 1.0.6 installer will upgrade your primary SmartForms site, both design-time and runtime. Any SmartForms sites that were copied in order to facilitate different authentication modes will not be upgraded by installer. Only those sites that were created through a K2 smartforms installation will be upgraded.
Once the upgrade of the primary SmartForms site has been completed, the site will be able to be configured for Multi-Auth.

Step 2 - Configure your primary SmartForms site for Multi-Auth

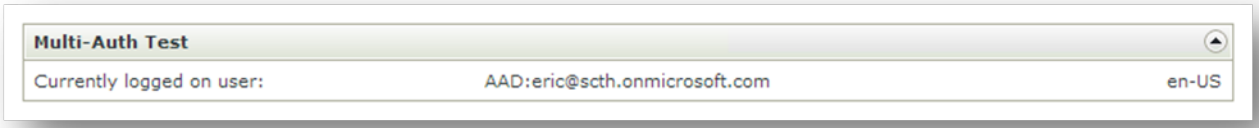
The following documents describe in detail how to configure your SmartForms site for various authentication modes.

- Configure SmartForms for Active Directory Federation Services (AD FS)
- Configure SmartForms for Azure Active Directory (AAD)
- Configure SmartForms for Forms Authentication using SQLUM

Follow the configuration documents for each type of authentication method you want to enable. Complete the configuration before continuing on to Step 3.

Step 3 - Confirm that your Multi-Auth configuration is working

Before removing the copied SmartForms sites that you created to support additional authentication providers, ensure that the Multi-Auth configuration on your primary SmartForms site is able to properly authenticate users from each of your configured authentication modes.
To ensure that the users are authenticating correctly it can be helpful to configure a Multi-Auth test form that upon initialize transfers the current user context to a data label.



Step 4 - Remove the secondary SmartForms sites

Once you are satisfied that the Multi-Auth functionality on the primary site is working properly you can remove the old copied SmartForms sites from IIS and the file system.

K2 Multi-Auth: Upgrading Secondary SmartForms Sites to 1.0.6

Overview

With the introduction of K2 blackpearl 4.6.7 and K2 smartforms 1.0.6 it is now possible to configure a single K2 smartforms site with multiple authentication providers. In the past if you required multiple authentication methods for your SmartForms sites you needed to create a copy of your SmartForms site for each authentication method. In some cases it may be desirable to continue having multiple sites rather than consolidate to a single site. This document describes the steps necessary to upgrade these copied sites to 1.0.6 and reconfigure them for the appropriate authentication method using the new Claims based / Realm model.

Should I consolidate instead?

Before you make the decision to upgrade your copied sites, review the Introduction to Multi-Auth document to understand if it would a single Multi-Auth enabled site would meet your needs. If you wish to consolidate please refer to the Consolidation to Multi-Auth document for detailed steps.

Configuration

High Level Steps

These high-level steps are provided for those familiar with upgrading K2 smartforms and configuring authentication methods. For a detailed guide, see the Detailed Steps topic below.

1. Upgrade your primary SmartForms site to 1.0.6
2. Copy the upgraded primary site over the secondary sites
3. Configure Realms for the secondary sites
4. Configure Claims for the secondary sties

Detailed Steps

Step 1 - Upgrade your primary SmartForms site to 1.0.6

The K2 smartforms 1.0.6 installer will upgrade your primary SmartForms site, both design-time and runtime. Any SmartForms sites that were copied in order to facilitate different authentication modes will not be upgraded by the installer. Only those sites that were created through a K2 smartforms installation will be upgraded.

Step 2 - Copy the upgraded primary site over the secondary sites

Since the SmartForms installer does not upgrade the copied sites, you will need to manually copy the files from the upgraded primary SmartForms site over the files in the copied SmartForms sites. Once this is done the sites will no longer be configured for the desired authentication modes. The subsequent steps will guide you on configuring the appropriate authentication mode for each site using the new Realm model.

Step 3 - Configure Realms for the secondary sites

In the new Realm configuration model all the SmartForms authentication configuration is stored centrally in the K2 database. Realms are used in order to distinguish which sites use which types of authentication. For each secondary SmartForms site that you are upgrading follow the steps below to create the Realm and configure the site to map to the Realm.

1. Open the Manage Site Realms form on your primary SmartForms runtime site. The form is located in the System -> Management -> Security -> Forms category. You can also run it directly using the following URL: https://{SmartFormsServer}/Runtime/Form/Manage+Site+Realms/
2. Under Realms click New
3. For the URI field enter the URL to your secondary SmartForms site (Example: https://k2.denallix.com/RuntimeSQLUM/)
4. For the Reply URI field enter the relative URL to the SmartForms site (Example: /RuntimeSQLUM/)
5. For the Linked Issuers you can select the appropriate option if it is available. If you don't see the appropriate option leave the checkboxes blank. The configuration steps in Step 4 will guide you through configuring an appropriate Issuer.
6. Click OK to add the Realm
7. Open the web.config file for the secondary SmartForms site and locate the Federation Configuration section
8. For the cookieHandler property set the path value to be the same as the Reply URI you configured above but without the trailing slash (Example: /RuntimeSQLUM)

```
<system.identityModel.services>
  <federationConfiguration>
    <cookieHandler requireSsl="false" path="/RuntimeSQLUM" />
```

9. For the wsFederation property set the realm value to be the same as the Realm URI you configured above (Example: https://k2.denallix.com/RuntimeSQLUM/)

```
<wsFederation passiveRedirectEnabled="false" issuer="http://none"
realm="https://k2.denallix.com/RuntimeSQLUM/" requireHttps="false" />
```

10. Save and close the web.config file

Step 4 - Configure Claims for the secondary sites

The final step is to configure or clean up the imported Claims configuration depending upon the type of authentication you wish to enable for the site. The documents listed below have all the necessary steps for configuring a SmartForms site to use each authentication mode. You may have already completed some of the steps in these documents as they assume that you are starting from scratch rather than upgrading. Since you are upgrading existing sites, the most important parts are the steps around the Issuer and Claim Mapping configurations. If you are upgrading an AD FS configured site, some of the Claim configuration will have been automatically imported into the database but you will need to fill in some additional details. If you are upgrading a Forms Authentication site that used SQLUM, there will be no imported claims configuration and you will need to follow all the steps.

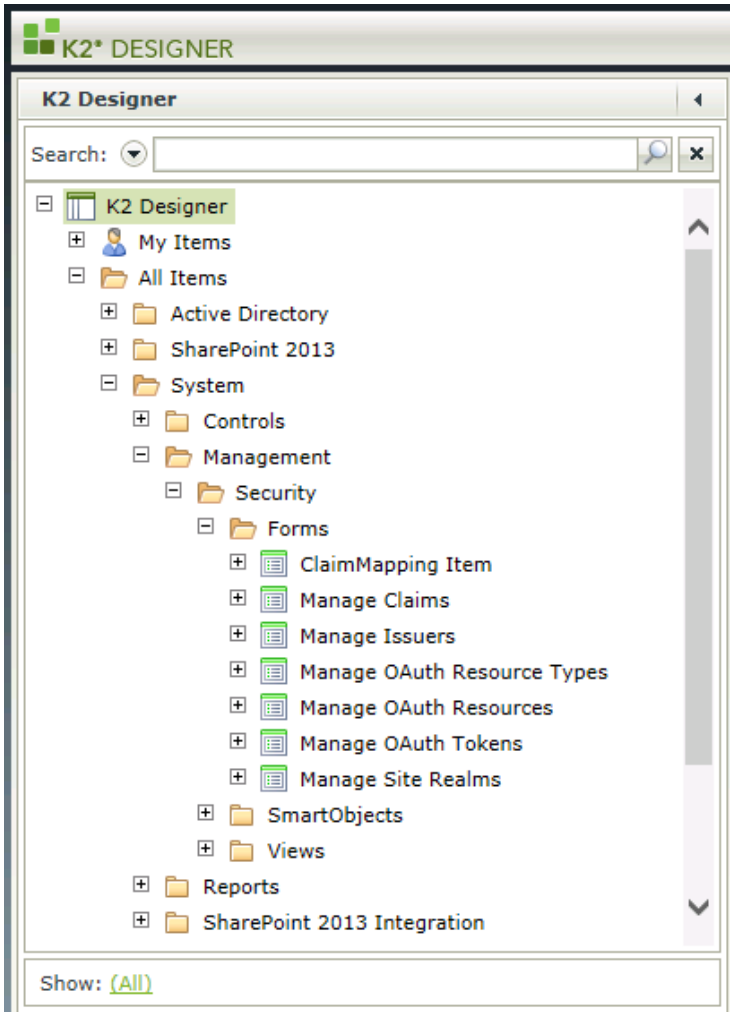
- Configure SmartForms for Active Directory Federation Services (AD FS)
- Configure SmartForms for Azure Active Directory (AAD)
- Configure SmartForms for Forms Authentication using SQLUM

K2 Authentication Management Settings

This document deals with the Authentication Management section of K2 smartforms, which provides configuration settings for Claims Authentication, Multiple Authentication, and the OAuth Token system used by K2 smartforms and the K2 for SharePoint App.

Caution: These settings are critical to the K2 system. An incorrect setting may cause errors and system instability. These settings should only be modified by an Administrator who is already familiar with Claims and OAuth configuration parameters and settings. The values for these settings are stored in the K2 database, so be sure that you have created a backup before editing them. If you mistakenly configure these settings, you will not be able to use the K2 Application because of various access and authentication errors.

To edit these settings, navigate in the K2 Designer to All Items > System > Management > Security > Forms. SharePoint Site Admin / Owner rights are needed in order to edit the Management settings.



Claims

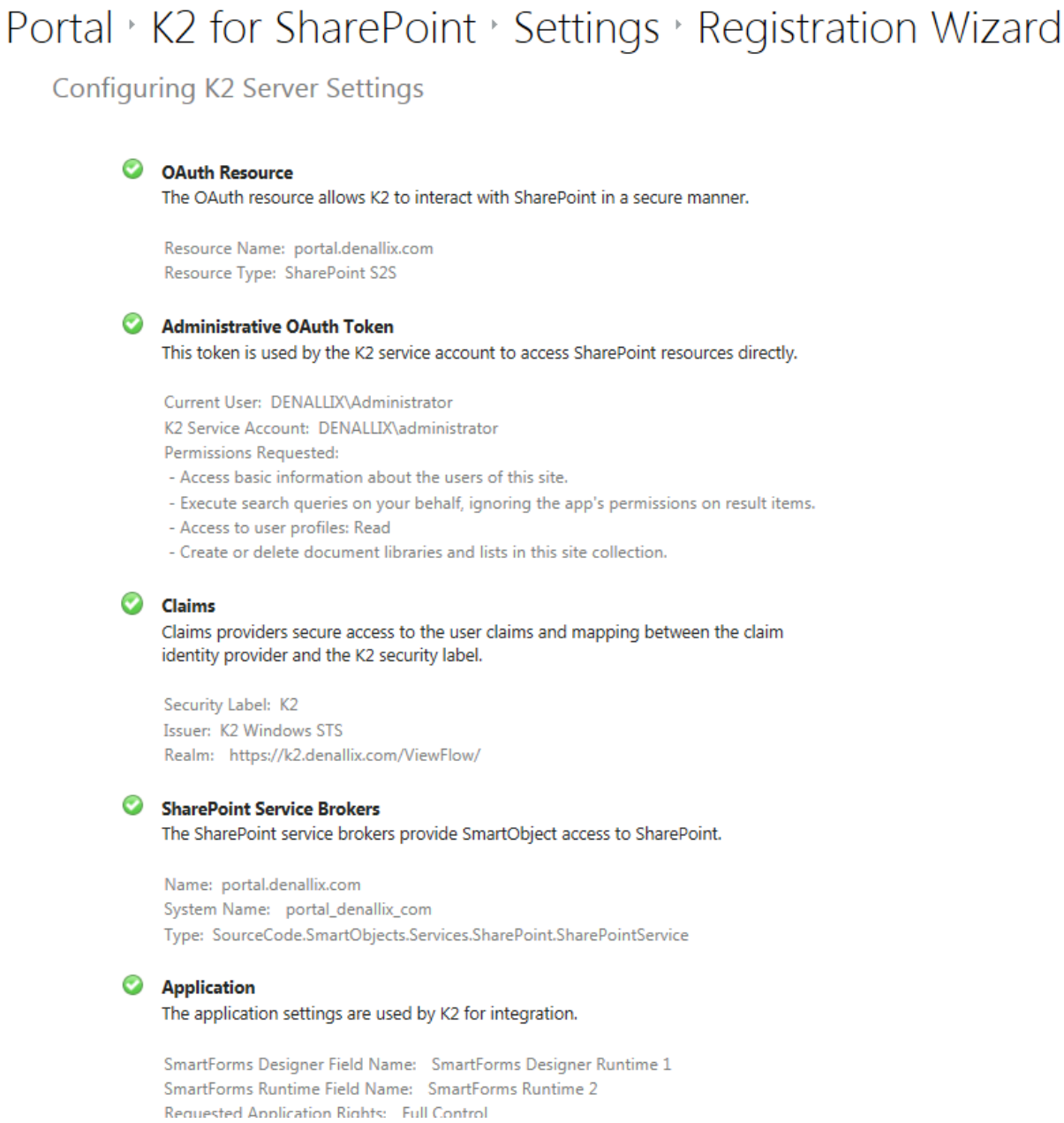
Claims-based authentication provides an industry standard security protocol to authenticate a user on a host computer. Claims-based authentication is a set of WS-* standards describing the use of a Security Assertion Markup Language (SAML) token in either passive mode (when used with a web application) or active mode (where WS-Trust in used with Windows Communication Foundation (WCF) clients).
Claims-based authentication requires the availability of a security token service (STS) running on a server. An STS server can be based on Active Directory Federation Services (AD FS) V2, or any platform that provides the official STS protocol.

Multiple User Authentication Managers

From K2 blackpearl 4.6.7 onwards, the K2 server supports multiple claims based user managers.
The K2 blackpearl installer creates two authentication managers; K2 Windows STS (uses the default Windows authentication login), and K2 Forms STS (creates a Claims based authentication for K2 smartforms so the user may login as a K2 user or with another ID). The identity settings for these managers are stored in the K2.

Default Installation Settings

The Registration Wizard sets and reports back the default configuration for Claims and OAuth, as shown in the screen capture below:



Claims Management

Manage Claims

An Administrator can edit, delete, or create a new Claims identity using the Manage Claims page. The following values can be created or edited:

- Security Label
- Issuer
- Claim Type Info
- Name Identity Issuer
- Identity Provider

    - Original Issuer
    - Claim Type
    - Claim Value
- Identity
    - Original Issuer
    - Claim Type
    - Claim Value

Identify Role values may be added, edited, or deleted.
By default, two Claims configurations are added when K2 blackpearl is installed; the K2 Windows STS , and the K2 Forms STS. An authentication query is performed using the first configuration in the list (Windows), and if that is unsuccessful then the next configuration is used (Forms). If other authentication configurations have been added then they are queried in the order they appear if the first two services are unsuccessful.

Manage Issuers

For each authentication manager in the Manage Claims page there exists a Claim Issuer configuration. The Manage Issuers page presents those Claim Issuers and allows the Administrator to create new issuers, edit the existing issuers, or delete the issuer configuration. The following values can be created or edited:

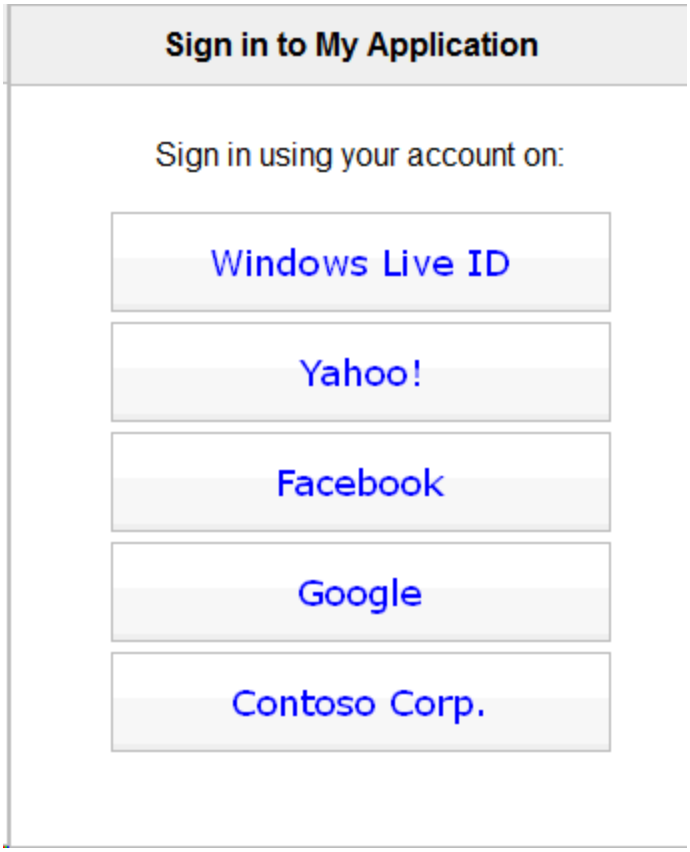
- Name
- Description
- Issuer
- URI
- Thumbprint
- Use for Login Checkbox

Manage Site Realms

The K2 Viewflow and the K2 smartforms design time and runtime sites are linked to claim issuers. Using the Mange Site Realms page, an Administrator can change the URI and Reply URI used for these sites, as well as the Linked Issuer. A ‘realm’ is the identifier for a site that is secured by claims (the Microsoft convention is to use a URL for this). An ‘audience’ property is the URI used to access a realm (the actual URLs used to access the sites in the realm).
K2 smartforms 1.0.6 design time and runtime sites may be configured to use different user authentication managers. This is done by adding Linked Issuers to the realm. Click on the realm that you want to add an authentication manager to. Click on Edit, and then add the Issuer from the Available list by highlighting it and using the arrows to add it to the Linked Column.

HomeRealm column in the Identity.ClaimRealm table

HomeRealm is available for customers who have configured multiple identity providers (IdPs), such as Facebook and Windows Live ID, via a Relying Party STS (RP-STS), such as AD FS or ACS, as the federated identity provider. When they do this, K2 will route the login request to the RP-STS and the user will be prompted to select the IdP they wish to sign into.



If a customer wishes to create a 1:1 mapping between the K2 web site and the IdP of the RP-STS, thisy can be done by specifying the HomeRealm value for the ClaimRealm entry. When this value is present, K2 will append whr=<Federation service URI of home realm> to the login requests.

Example: https://mydomain/myadfsapp/?whr=urn:federation:contoso

NOTE: The HomeRealm value in the db should only be the highlighted portion above.

CAUTION: Specifying the HomeRealm value will force all K2 users to use the same IdP. If you need your Relying Party STS to support multiple IdPs for the same K2 site, you cannot use HomeRealm (leave it blank).

OAuth Managment

The following three management pages deal with OAuth settings. The Registration Wizard of the K2 for SharePoint installer adds the OAuth authentication settings during installation. The management pages are then used if those settings need to be modified.

Manage OAuth Resource Types

This page allows the Administrator to create new OAuth parameters or edit the existing ones. By default, the following OAuth Type parameters are added during the installation of K2 for SharePoint:

- client_id
- grant_type
- api_version
- scope
- client_secret

- resource
- entity_id
- response_type
- redirect_uri

These parameters then have the following properties set:

- Uri Encode : true/false
- Authorization Request : true/false
- Authorization Response : true/false
- Authorization Default Value :i.e. code
- Token Request: true/false
- Token Response: true/false
- Token Default Value : i.e. authorization_code
- Refresh Request: true/false
- Refresh Default Value : i.e. refresh_token

These properties communicate the credentials request and return strings while using OAuth in K2 for SharePoint.

Manage OAuth Resources

The OAuth Resources page allows the Administrator to add, edit, or delete an OAuth resource and to add, edit, or delete Parameters of that OAuth resource. The following values may be configured for the OAuth resource:

- Name
- Type
- Authorization Endpoint
- Token Endpoint
- Use Host Server Authorization Endpoint

For each OAuth resource configuration, a list of resource parameters is created. Each of these parameters has a Name, Authorization Value, Token Value, and Refresh Value property. The default parameters are:

- certificate_password
- sharepoint_pid
- k2_static_claim_upn
- k2_static_claim_nii
- sharepoint_uri
- client_id
- certificate_path
- sharepoint_site_realm

- application_level_delegation
- trusted_for_impersonation_claim_type
- k2_static_claim_nameid

These parameters store the OAuth communication settings and identity strings.

Manage OAuth Tokens

This management page allows the Administrator to delete OAuth tokens. This is useful for dealing with tokens that were created for an employee who has left the company, or for situations where the OAuth configuration has been modified and the old tokens are no longer applicable. The stored tokens will display the following properties:

- Resource Type
- Primary Credential ID
- User Name
- Expires in Seconds

K2 for SharePoint

Overview and Architecture of K2 for SharePoint

The K2 for SharePoint app is a provider-hosted app for SharePoint 2013. Provider-hosted apps allow for server-side execution, and the K2 for SharePoint app takes advantage of this by surfacing artifacts and pages from a K2 server directly into SharePoint for a rich application experience.

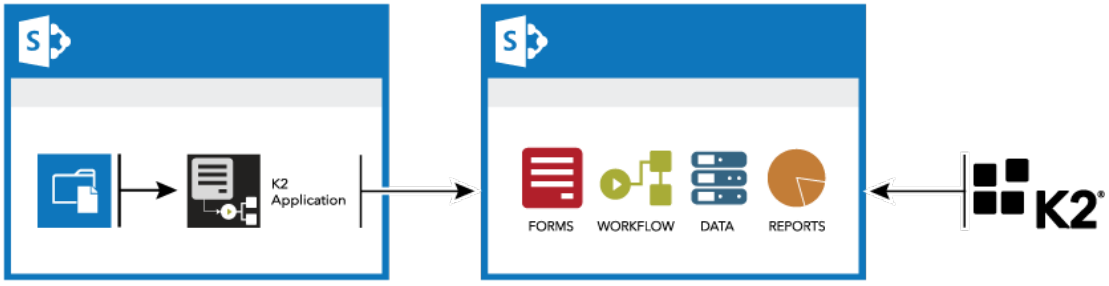
Important: Because K2 for SharePoint is an app, your SharePoint 2013 site must first be enabled for apps before you can install it. You must also have a fully-configured K2 server ready to register the app once it is installed and activated to the site. For more information about enabling your SharePoint 2013 site for apps, see the MSDN article Install and manage apps for SharePoint 2013. It is recommended to install a simple, free app from the SharePoint Store in order to test that your SharePoint 2013 site is ready for apps before attempting to install the K2 for SharePoint app.

Architecture

The K2 for SharePoint app itself is relatively lightweight and easy to install. It is the only thing that is required to be installed on the SharePoint server itself and includes:

1. A hidden K2 Settings list that stores the link between the app and the K2 server, along with a few other items like environment library fields.
2. A hidden K2Pages list that stores the pages used to host K2 content.
3. The K2 Worklist App Part which allows users to see their assigned K2 tasks.
4. The K2 Forms Viewer App Part which can be used to display a SmartForm.
5. The Ribbon Bar items for creating a K2 Application on a list or library and for viewing reports.

The K2 for SharePoint app must be registered with a K2 server to show K2 content and functionality in SharePoint. A SharePoint generated app domain hosts an iFrame that surfaces K2 content from the K2 Server. All interaction between SharePoint and K2 takes place via K2 SmartObjects. These, in turn, depend on the K2 for SharePoint Service Brokers that are registered with a K2 server during the Registration Wizard.



Note: The K2 for SharePoint app functions the same online and on-premises (on-prem), however there are different versions of the app. See Versions of the K2 for SharePoint App for more information.

Versions of the K2 for SharePoint App

There are two different versions of the K2 for SharePoint App.

1. The K2 for SharePoint app in the SharePoint Store: This app can be installed directly on any on SharePoint site that is enabled for apps, whether that site is online or on-prem. This app can only request Manage rights when it is installed, so items such as creating sites and SharePoint groups cannot be automated by K2 using this version of the app. See How To: Deploy a Full Control K2 Application to SharePoint Online for more information.
2. The K2 for SharePoint app in the K2 for SharePoint download: This app is installed with all on-prem versions of SharePoint 2013 using the AppDeployment.exe installer that is chained with the main K2 for SharePoint installer. It can also be run separately. This app requests Full Control rights on the site when it is installed, so all functionality included in the K2 Designer when designing a workflow is functional, such as creating sites and SharePoint groups.

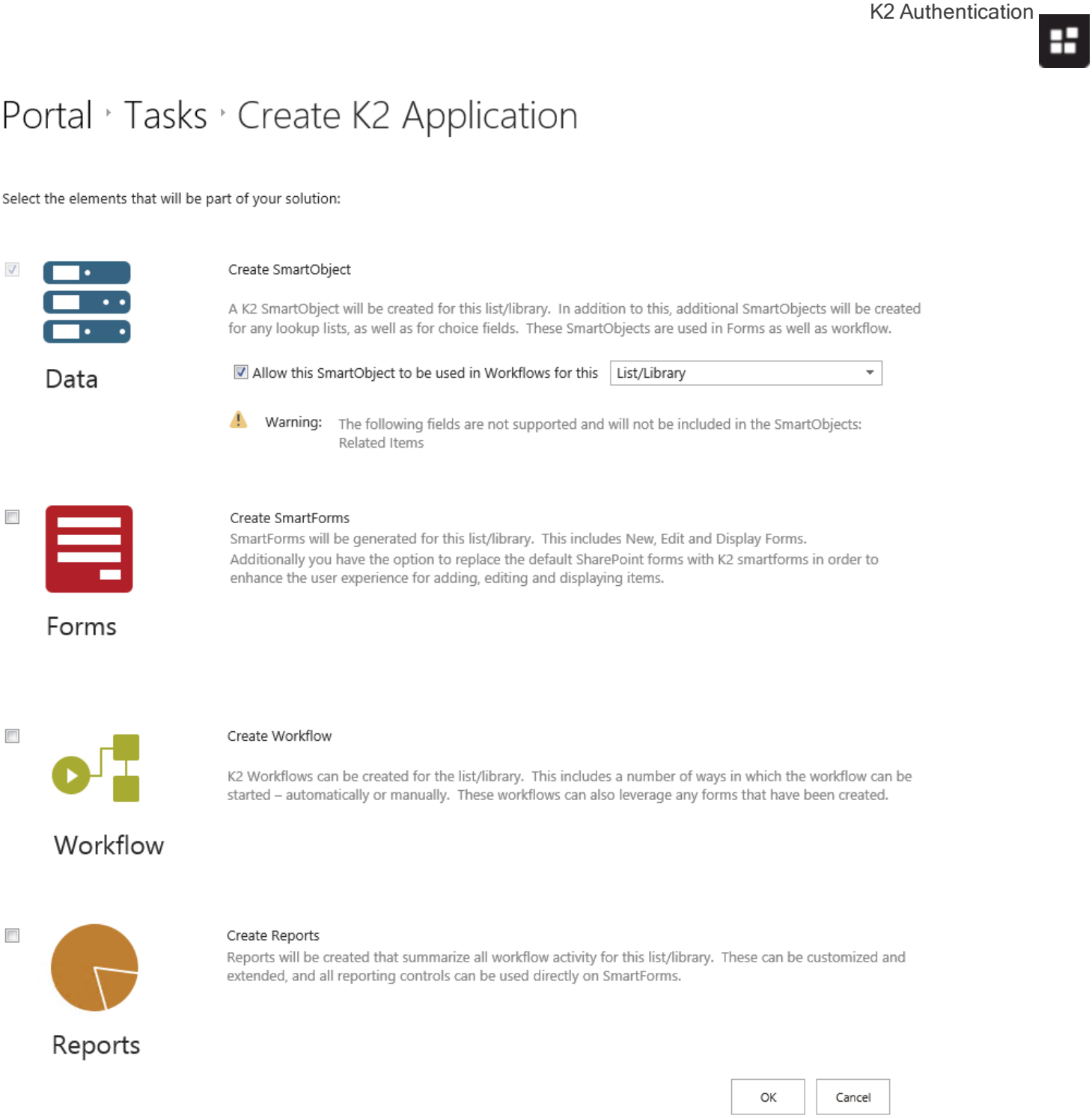
Registering the K2 for SharePoint App with a K2 Server

Once you have the K2 for SharePoint app installed, you must register it with a K2 server. The K2 server must have the version of components that the K2 for SharePoint app requires.
The K2 AutoDiscover service allows you to supply any K2-related URL and it will return what the K2 for SharePoint app needs, namely the K2 smartforms Designer and Runtime URLs. You can copy the AutoDiscover IIS application from the K2 server to make it accessible to the internet if necessary. The K2 server, however, must be accessible from that IIS location in order for the AutoDiscover service to query the K2 server for the latest values.
Notes:

- More than one K2 for SharePoint app can be registered with a single K2 server. This means that you can have all of your SharePoint 2013 farms and/or SharePoint Online tenancies being served by a single K2 server.
- You must install and register the K2 for SharePoint app on every site you wish to use it on. The registration wizard will automatically navigate between pages if it discovers that the root site in the site collection has already been configured.
- If you want to walk through the wizard manually, after registering the site with K2 you can go to Settings > Registration Wizard.

How the K2 for SharePoint App Works

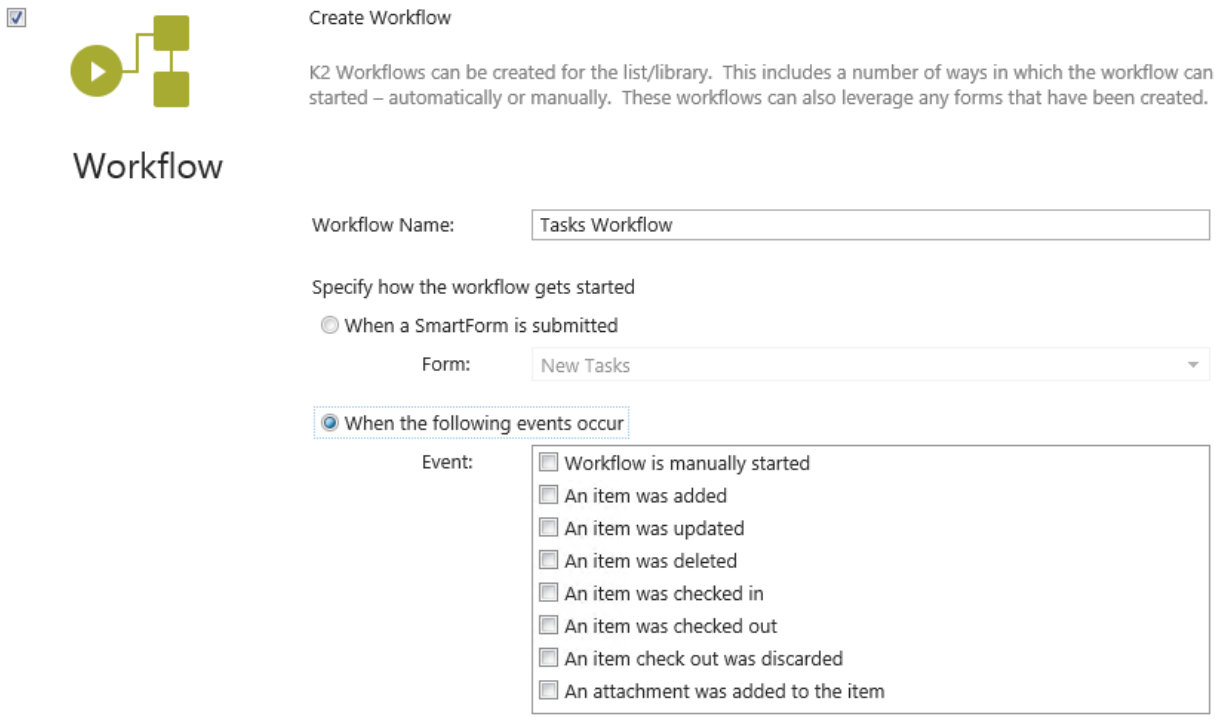
Once the K2 for SharePoint app is registered with a K2 server, the K2 Settings hidden list is populated. This list stores the link to the K2 server, what groups have K2 Designer and Participant permissions, the environment library fields for the SmartForms runtime and design time URLs, the version of the K2 for SharePoint application, and a few other details for the K2 for SharePoint app integration.
When a list is integrated with K2 for SharePoint, various pieces of information are recorded depending on what you select for your K2 solution. The Create K2 Application page allows you to select what you want to create.



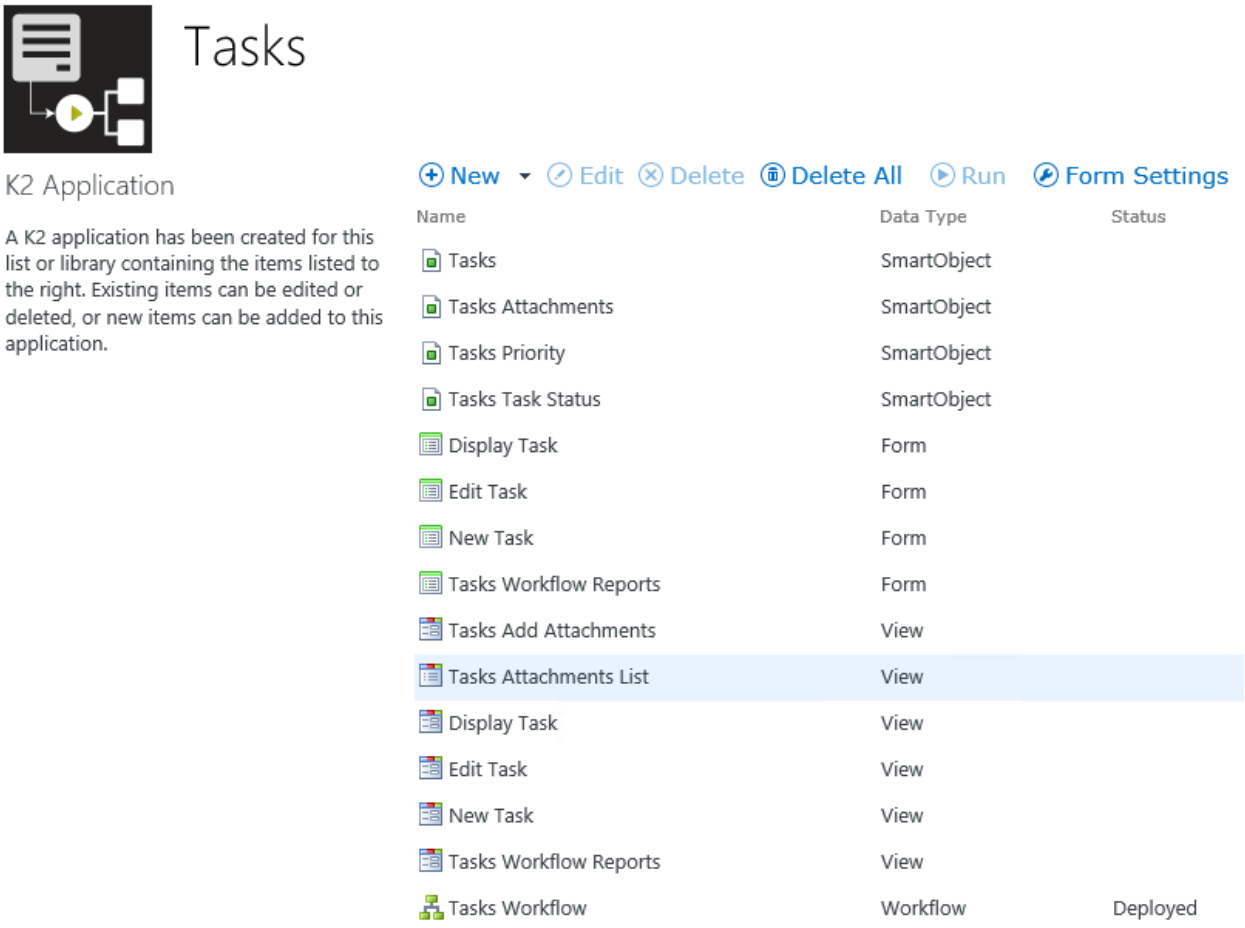
- Data: Creates SmartObjects for the list or library. If the list supports attachments, a separate but associated SmartObject is created for the attachments. If the list or library includes a lookup column, a SmartObject is created for that lookup column. This integration is mandatory for any K2 solution and cannot be unchecked.
- Forms: Forms are created based on the columns in the list or library. You can also choose to replace the existing SharePoint forms with SmartForms. You can switch this back at a later time by clicking Forms Settings from the artifacts page, which is the default page you'll navigate to after your solution is created and you click the K2 Application button in the Ribbon again.

- Workflow: A workflow is created for the list or library. When selecting this option you are taken directly to the K2 Designer to start designing your workflow. If you do not select this option you are taken to the K2 artifact page.
- Reports: Forms and views are created for reporting on the workflow.

When selecting a workflow, you can choose how the workflow starts. For each event that you select, an event receiver is created for the list or library. This is important because if your K2 server is not available over a secure socket layer (SSL, typically port 443) and you want to automatically start workflows based on list or library events, the event receiver will not be able to find the K2 service to start the workflow. The name of the K2 service is RemoteEventService.svc and is part of the SP15EventService site on the K2 server. For more information, see XXX.



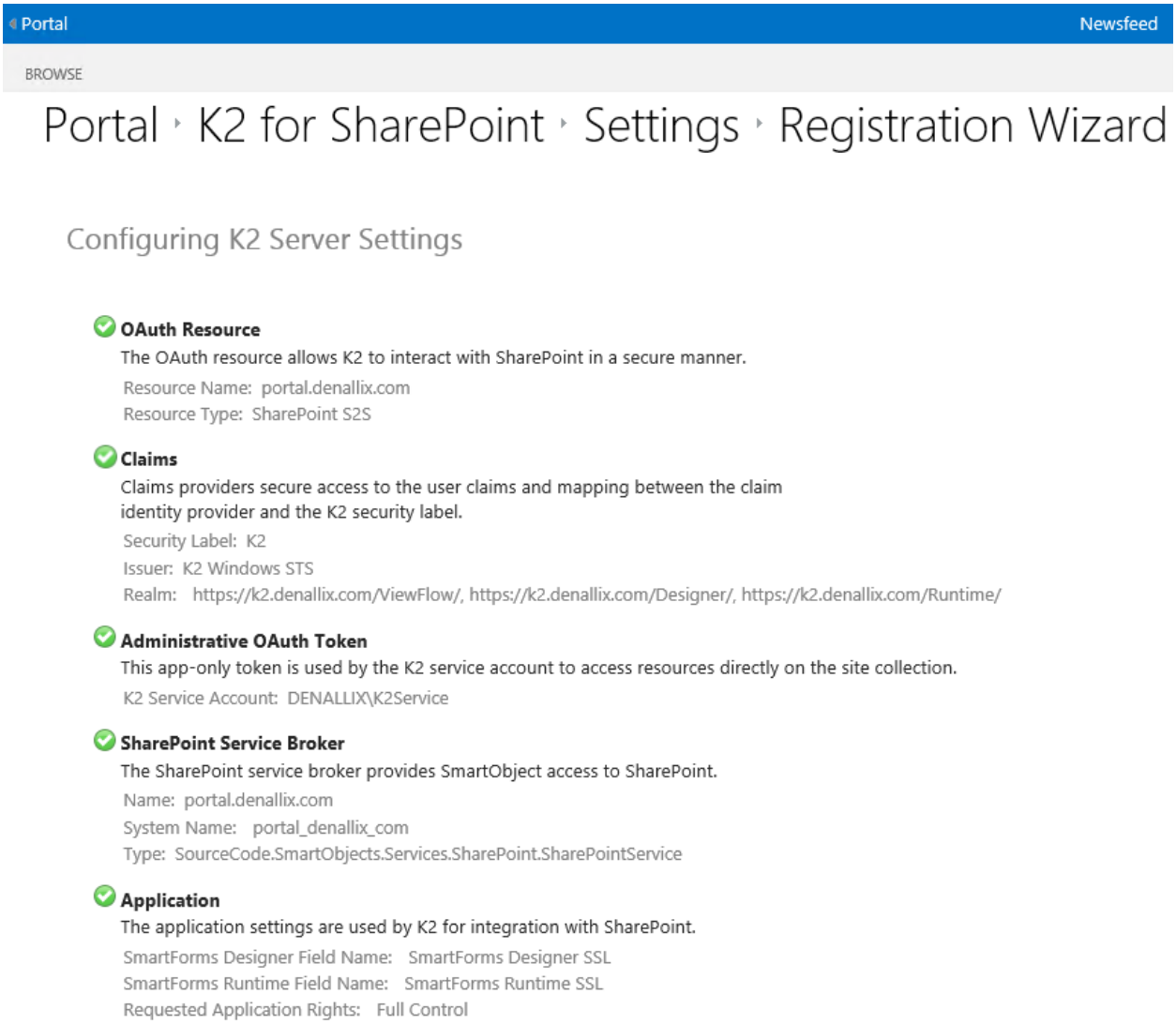
From the artifacts page you can create, edit and delete parts of the K2 solution. As pictured below, this Tasks list is integrated with K2 and all aspects have been generated. This page is a K2-owned page that is surfaced in SharePoint via an iFrame. If you look at the URL you'll see something like https://app-6210f94d-b9d1e7.denallixapps.com/....
This URL indicates that SharePoint has given control over to the app registered with that domain. This is inherent to the SharePoint 2013 app model and is how the context switching occurs between SharePoint and a provider-hosted app such as K2 for SharePoint.
Clicking on any of the K2 artifacts on this page and clicking Edit will allow you to edit that item using the K2 Designer. This is surfaced through SharePoint from the K2 Designer, which is installed with K2 smartforms. If you do not have a license for K2 smartforms you cannot edit your forms and views, but can edit your SmartObjects and workflows.



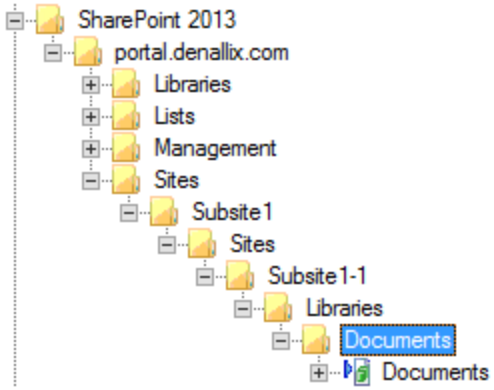
The SharePoint 2013 Service Brokers

The SharePoint 2013 service brokers, SharePoint 2013 and SharePoint 2013 Integration, play a key role in the integration of K2 with SharePoint 2013. When instances of these brokers are generated via the Registration Wizard, the authentication uses either a high-trust, server to server (S2S) token for on-premises SharePoint 2013 servers, or a standard token along with an Azure AD token for SharePoint Online (Office 365) servers.

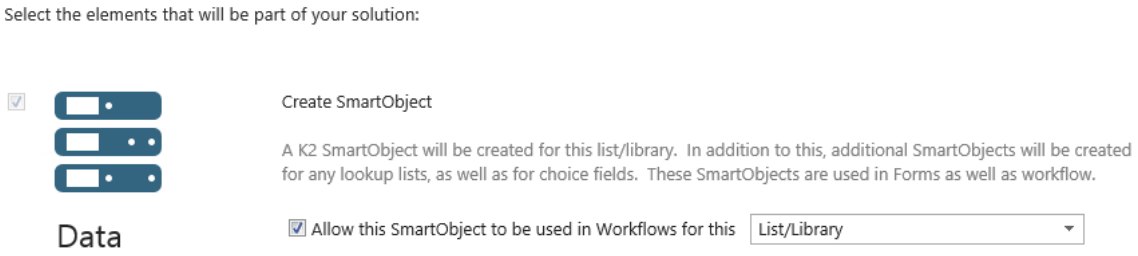
These brokers are OAuth-capable and are setup to use OAuth during the Registration Wizard. In the following screen you see that an administrative OAuth token is generated for this site collection. It is important to note that while the K2 for SharePoint App must be added to each and every site in the site collection, the Administrative OAuth token and the actual instance of the SharePoint 2013 broker is used per site collection.



If the K2 for SharePoint App is added to a subsite of the site collection, it is added to the service instance that already exists for the site collection. The structure of the artifacts generated, such as SmartObjects, View, Forms, Workflows and Reports, follows the structure of the site collection. For example, if you have a subsite called Subsite1 that was a child of your root site, and created a subsite under Subsite1 called Subsite1-1, when you add the K2 for SharePoint app to Subsite1-1 and generate at least a data app for the Documents library on Subsite1-1, the following structure is created automatically.



To generate just a data app for the Documents library on Subsite1-1 you would go to the library, click the K2 Application button, and then makes sure Data is selected. This refreshes the service instance and generates the SmartObject in the corresponding category.



You can also use the SmartObjects link from the K2 for SharePoint Settings page to generate multiple SmartObjects at a time. This is especially helpful if you need to use data from other lists and libraries on your site but do not need to create additional artifacts, such as Views, Forms, Workflows and Reports, for those lists and libraries. You can also add line of business SmartObjects using this page.

Notes:

- The scope of the SmartObjects on this page and the Create Solution page allows you to control if the methods of the SmartObjects appear in the K2 Workflow Designer. If you do not need these SmartObjects to show up in the designer you can select None. Note that selecting This site and all of its subsites allows the SmartObject to appear in when designing workflows on sites that are direct descendants of the site.
- It is highly recommended to allow the K2 for SharePoint app maintain the structure of the service instance and SmartObjects. You should not need to refresh the service instance manually or move artifacts from their existing categories as they are generated.
- The SharePoint 2013 Integration broker is not meant to be used directly for K2 solutions that original inside SharePoint or that originate from outside of SharePoint but use SharePoint data. Use the SharePoint 2013 broker for accessing SharePoint as a line of business app.

SharePoint 2013 Integration Requirements

Overview

The K2 for SharePoint 2013 integration components use the new SharePoint 2013 apps architecture. This architecture allows 3rd party applications, like the K2 for SharePoint app, to be used with SharePoint on-premises, SharePoint Online (Office 365), as well as a mixture of online and on-premises. In each of these scenarios there are certain prerequisites that need to be met in order for the integration to function. This document outlines the required prerequisites for each SharePoint infrastructure scenario.

SharePoint on-premises

This section outlines the K2 and SharePoint requirements for using the K2 for SharePoint app with an on-premises SharePoint environment.

Requirements

SSL for K2 Site

If SharePoint is configured for SSL then K2 also needs to be configured for SSL. The K2 for SharePoint registration wizard will prevent you from proceeding if a mismatch is detected between the SSL settings for SharePoint and K2.

Enable SharePoint for Apps

After you install SharePoint on-premises there are a number of manual configuration steps that must be completed in order to enable your SharePoint environment for apps. These steps are not specific to the K2 for SharePoint app. These steps must be completed in order to install any 3rd party app into your SharePoint environment.
The following TechNet article contains a collection of resources to guide you in the configuration and management of apps in your SharePoint environment:
http://technet.microsoft.com/en-us/library/fp161232.aspx

App Upload and Installation Permissions

There are a minimum set of permissions that are required of the user that will upload the K2 for SharePoint app into the SharePoint app catalog and of the user that will install the K2 for SharePoint app onto a SharePoint site.

- Permissions required to upload the K2 for SharePoint app to the SharePoint app catalog
    - Local Administrator on the SharePoint Server
    - K2 Administrator
    - Site Collection Administrator of the App Catalog Site Collection
    - db_owner Access to the SharePoint_Config Database
    - SharePoint Shell Access role (http://technet.microsoft.com/en-us/library/ff607596(v=office.15).aspx)

- Permissions required to install the K2 for SharePoint app onto a SharePoint site
    - Contributor Rights on the SharePoint site

SharePoint online (Office 365)

This section outlines the K2 and SharePoint requirements for using the K2 for SharePoint app with an online SharePoint environment.

Requirements

SSL for K2 Site

SharePoint Online is always SSL enabled and thus it is mandatory for the K2 site to also be enabled for SSL. The K2 for SharePoint app registration wizard will prevent you from proceeding if this configuration has not been completed.

Internet Accessible K2 Sites

When using the K2 for SharePoint app with SharePoint Online there is communication that needs to take place between your on-premises K2 Server and your online SharePoint environment. This may require opening ports and sites through your corporate firewall. There are two scenarios listed below, each with their own requirements. Follow the steps for the scenario that best fits your needs.

- All users of your SharePoint Online environment will be behind your corporate firewall

- The SP15EventService on the K2 Server that SharePoint will call when SharePoint events occur will need to be publically available on the web. This is the service that is used for initiating workflows from SharePoint events (Example: File Uploaded, Item Added, etc.)

- Some or all of your users will access your SharePoint Online environment from outside your corporate firewall
  - The entire K2 site and your K2 smartforms sites, will need to be made publically available on the web.

Azure Active Directory

SharePoint Online requires the use of Azure Active Directory (AAD). When you register the K2 for SharePoint app K2 will automatically register the appropriate resources against your AAD tenant for the purpose of Authentication and Authorization.

App Upload and Installation Permissions

There are a minimum set of permissions that are required of the user that will upload the K2 for SharePoint app into the SharePoint app catalog and of the user that will install the K2 for SharePoint app onto a SharePoint site.

- Permissions required to upload the K2 for SharePoint app to the SharePoint app catalog
  - Tenant Admin of the SharePoint environment

- Permissions required to install the K2 for SharePoint app onto a SharePoint site
  - Contributor Rights on the SharePoint site

Mixture of SharePoint on-premises and SharePoint Online

It is increasingly common for organizations to have a mixture of SharePoint on-premises and SharePoint online. A single K2 server can support both of these environments at the same time however there are some slight differences to the configuration if your on-premises SharePoint environment is setup in Hybrid mode. You can find more details on enabling Hybrid mode for SharePoint 2013 in the Hybrid for SharePoint 2013 TechNet article. Follow the steps in the SharePoint Server 2013 Hybrid section below if this applies to your environment. If you are not configured for Hybrid authentication then the requirements are simply a combination of the two sections listed above for SharePoint on-premised and SharePoint Online.

SharePoint Server 2013 Hybrid

In organizations with both SharePoint on-premises and SharePoint Online there is a configuration mode available to allow these two environments to work more closely together known as Hybrid mode. The excerpt below from the Hybrid for SharePoint Server 2013 TechNet article describes the high level features of enabling Hybrid mode.

."A SharePoint Server 2013 hybrid environment enables identity management and trusted communications between SharePoint Online and SharePoint Server 2013. When you have established this trust framework, you can configure integrated functionality between services and features such as Search, Microsoft Business Connectivity Services, and Duet Enterprise Online for Microsoft SharePoint and SAP."

When K2 for SharePoint is installed in a SharePoint Server 2013 hybrid environment, the installation experience will be a combination of on-premises and online experiences . The K2 for SharePoint app will be installed to the on-premises and online app catalogs. However, since both the on-premises and online SharePoint environments will be configured for Azure Active Directory, the K2 application will treat the users and authentication the same as in the SharePoint Online scenario.
For more information on upload the K2 for SharePoint App into a SharePoint Server 2013 hybrid environment please review the following KB article: http://help.k2.com/en/kb001443.aspx