# Workday

This information was last reviewed in June, 2018, by Scott Cantor.

Change Log:

> *Jun 11, 2018 – Removed superfluous signed request material, this isn't good advice at this point.*
>
> *Nov 3, 2016 – Added more information about Logout.*
>
> *Sep 19, 2016 – Added explicit advice to separate tenants by entityID due to lack of ACS in SAML request.*

This is not a replacement for the actual documentation and you cannot cut and paste your way to a working system. The examples are **not** usable without taking into consideration your local needs and requirements.

This is a page for documenting Shibboleth integration with Workday.

The official SAML documentation provided by Workday is here at the time of authoring. It requires an account in Workday's authentication system.

In the version being documented, the settings being documented in Workday are under "Workbench -> Security & Audit -> Edit Tenant Setup - Security".

> *Unknown: Whether a role exists for a SSO administrator to only have access to the authentication settings.*

Be aware that Workday can function as both an SP and an IdP, and it can be confusing sometimes to follow their documentation and understand to which aspect they're referring. This page is only discussing its SP functionality.

## Identity Provider Metadata

Workday does not consume SAML metadata, and provides a self-service interface for manually managing the IdP integration settings required. The settings apply "on the fly" any time they're changed by an administrator.

There are a large number of settings to look at and there's a lot of fast and loose use of terms, but most of the IdP information is entered by "adding" an Identity Provider (you can register more than one) and filling in these details:

- Identity Provider Name (descriptive label)
- Issuer (your entityID)
- x509 Certificate (your signing key certificate, typically in *credentials/idp-signing.crt)*

> *The SAML Logout support is so far not proving functional for me, so I haven't documented it.*

Down below that section, there are details related to how the Workday SP issues its requests, and that includes a setting for the IdP's SSO Service URL, which you should set to your SAML 2 POST binding endpoint (https://hostname/idp/profile/SAML2/POST/SSO). The reason for this is that they do not support the HTTP-Redirect binding.

You should also check the Do Not Deflate... option. Sending a POST request while deflating is not SAML compliant (their system is apparently providing options to send non-compliant messages to accomodate broken IdPs).

## Service Provider Metadata

Workday does not produce SAML metadata(*), and provides a self-service interface for manually managing the SP settings used that ultimately need to be reflected in metadata for the IdP. The settings apply "on the fly" any time they're changed by an administrator, so you must be cautious to change settings in most cases only after altering the metadata. For example, you can't change the key the SP will use until you've made the key known to the IdP, and the same goes for other settings.

(*) There are references in the documentation to the ability to generate metadata, but the documentation specifically describes this as generating metadata for Workday acting as an IdP to some other SP.

There are a few different settings that influence what needs to go into the metadata you give the IdP:

- Service Provider ID (entityID, you should use different values for different tenants)
- x509 Private Key Pair (misnamed, this is a private key and public key certificate you generate in Workday if you want it to sign requests)

They do not appear to support XML Encryption, so if you do not require signed requests, you don't need to generate a keypair, and the metadata need not contain a `<KeyDescriptor>`. More around signed requests later.

> *My experience is that their interface only allows a keypair to be generated with a certificate lasting up to 3 years. It is unknown whether it will begin to fail to operate if the certificate is allowed to lapse.*

The main content of the rest of the metadata consists of the various endpoints, and those seem to be based on the tenant and seem to be shown in various places throughout their documentation. There are endpoints for both standard and accessible interfaces, for some reason, and there are endpoints for logout.

An example put together from a working tenant with the details removed follows:

**Example SP metadata for Workday**

```
  <EntityDescriptor entityID="http://www.workday.com/tenant1"
                xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
                xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
    <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <Extensions>
        <mdui:UIInfo>
          <mdui:DisplayName xml:lang="en">Workday</mdui:DisplayName>
        </mdui:UIInfo>
      </Extensions>
      <KeyDescriptor use="signing">
          <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:X509Data>
              <ds:X509Certificate>
<!-- Certificate from the x509 Private Key Pair setting, if used, if not, remove entire KeyDescriptor element --
>
              </ds:X509Certificate>
            </ds:X509Data>
          </ds:KeyInfo>
      </KeyDescriptor>
<!-- Discussed later, this will trigger the IdP to use a particular NameID Format to link users -->
      <NameIDFormat>urn:oid:2.16.840.1.113730.3.1.3</NameIDFormat>
<!-- Replace "tenant1" in the service locations with your tenant name -->
      <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
        Location="https://impl.workday.com/tenant1/logout-saml.htmld"/>
      <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Location="https://impl.workday.com/tenant1/login-saml.htmld" index="1"/>
      <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Location="https://impl.workday.com/tenant1/login-saml.html" index="1"/>
    </SPSSODescriptor>
  </EntityDescriptor>
```

## Profile Requirements

Making SSO work in general required filling in several redirection URLs in a generic section at the top under Single Sign-On labeled Redirection URLs:

- Login Redirect URL
- Mobile App Redirect Login URL
- Mobile Browser Login Redirect URL

Set them to the ACS location placed into the SP's metadata (https://impl.workday.com/tenant1/login-saml2.htmld).

Workday does not always include the desired ACS location to use in its requests; as a consequence, you will need to ensure that you use separate metadata with distinct entityIDs for each tenant you must support. Otherwise a request from one tenant will result in a response to whichever endpoint is the default in the metadata you create.

Workday does not appear to support XML Encryption.

Workday's SAML Setup includes a number of options that influence how it interacts with the IdP. Most of them relate to signing of requests or including the `ForceAuthn` option (the latter would bypass SSO for all users if the IdP is properly configured).

A checkbox labeled "Enable Signature KeyInfo Validation" is unclear, but from its description it should probably be checked.

In most cases you'll want to let Workday push users to your IdP and back, by checking the "Enable SP Initiated SAML Authentication" checkbox.

Finally, the whole thing has to be enabled with an "Enable SAML Authentication" checkbox when you're ready to test it.

### Signed Requests

Signing requests is not advisable and turning this off avoides the need to bother managing a keypair on behalf of the IdP.

### Logout

There is a way to configure simple redirect-based logout by setting the Logout Redirect URL to the IdP's simple logout endpoint (https://hostname/idp/profile/Logout).

The SAML Single Logout support within Workday is broken, at least on the initiating side. Regardless of the contents of the assertion used, Workday creates a `<LogoutRequest>` message containing one of two hard-coded `<NameID>` format values, either the X.509 or unspecified constants. This is a violation of the standard, which requires a strongly matching identifier (meaning the formats have to match) for the IdP to proceed with a logout.

It's possible that Workday may be able to process incoming logout requests from the IdP by virtue of violating the standard in the opposite direction (ignoring the format itself), but it's not safe to enable the feature since you can't prevent it from initiating broken requests itself.

The bug has been reported to Workday, but they have so far been unwilling to accept that what they have implemented is simply incorrect and are treating it as an enhancement request rather than a bug.

### Example Shibboleth Configuration

✓ Refer to the RelyingPartyConfiguration topic and be cognizant that creating overrides for every service is generally an inefficient use of the software. Consider identifying common requirements across services and create overrides tied to multiple services that share those requirements, or that reference profile configuration beans containing common settings.

**Required Profile Configurations**

SAML2.SSO

**Optional Profile Configurations**

SAML2.Logout

Refer to the SecurityConfiguration topic for examples on disabling encryption in different ways.

## Account Provisioning

Most sites will create identities in Workday directly as part of business processes, or possibly provision accounts into Workday using integration APIs or batch feeds if using only, e.g., the Financials portion. The only field relevant for SSO is the primary ID field in the record. There are fields available for storing alternate identifiers, but all information obtained indicates that SSO can **only** be based on the primary ID.

> *Unknown whether any form of just-in-time provisioning is supported.*

## NameID Requirements

Workday requires that the value in the user's primary ID field in Workday be communicated in the `<NameID>` element in the assertion. The `Format` of the element is ignored (in fact, the logout bug mentioned above pertains to a mistake in the `Format` value communicated back to the IdP).

Because the format is ignored, you should use a constant value that is defined in the SAML specification that corresponds to the type of identifier you're sending (not common, but possible), or more likely, use a value that matches that name of the SAML Attribute that would ordinarily carry the data you are passing in the element, and that you may already be using with other SPs.

For example, if you pass an employee ID in the element, use the standard name of the "employeeNumber" attribute, "urn:oid:2.16.840.1.113730.3.1.3".

### Example Shibboleth Configuration

✓ Refer to the NameIDGenerationConfiguration topic for a full treatment of NameID features.

Continuing with the example above, if you have an attribute definition named "employeeNumber" produced by your AttributeResolverConfiguration, release it to the Workday SP in your AttributeFilterConfiguration (example below).

Since Workday metadata must be manually supplied to the IdP, the usual way of producing the right `<NameID>` format is by including a `<NameIDFormat>` element in the metadata, which is illustrated in the example metadata shown earlier.

Finally, to actually produce the necessary `<NameID>`, modify *saml-nameid.xml* as shown:

**Example saml-nameid.xml changes**

```
        <!-- SAML 2 NameID Generation -->
        <util:list id="shibboleth.SAML2NameIDGenerators">

                <ref bean="shibboleth.SAML2TransientGenerator" />

                <!--
                <ref bean="shibboleth.SAML2PersistentGenerator" />
                -->

                <!--
                Add custom support for employeeNumber-based NameID, assumes you've released
                the source attribute (employeeNumber) to any SPs expecting to get it.
                -->
                <bean parent="shibboleth.SAML2AttributeSourcedGenerator"
                        p:format="urn:oid:2.16.840.1.113730.3.1.3"
                        p:attributeSourceIds="#{ {'employeeNumber'} }" />

        </util:list>
```

**Example attribute-filter.xml changes**

```
        <AttributeFilterPolicy id="Workday">
                <PolicyRequirementRule xsi:type="Requester" value="http://www.workday.com/tenant1" />

                <AttributeRule attributeID="employeeNumber" permitAny="true" />
        </AttributeFilterPolicy>
```

## Attribute Requirements

Workday does not appear to support SAML Attributes.