# ReleaseNotes

Please review these release notes before upgrading your system. You should review all the versions subsequent to the one you're running prior to upgrade.



## 3.4.5 (Unreleased)

11 issues

This is a patch update.

Among other fixes, it extends support for binary LDAP attributes when using the UnboundID LDAP provider in place of JNDI. Both the older JNDI property and a new (portable) `<BinaryAttributes>` element are supported.

## 3.4.4 (May 1, 2019)

15 issues

This is a patch update containing a number of bug fixes, primarily motivated by two issues in particular:

- Changes to the HTTP Client code to correct an issue with client certificate authentication during TLS renegotiation.
- A fix to the JPA StorageService to fail properly in the event that a case-insensitive database is used by mistake.

It also includes additional Java libraries that allow use of the UnboundID LDAP provider in place of JNDI, to work around a bug that occurs in newer Java versions. This change is not automatic, but switching providers can be accomplished with the use of a single property:

```
idp.ldaptive.provider = org.ldaptive.provider.unboundid.UnboundIDProvider
```

A more complete description of this issue can be found here. Switching the property alone will adjust the provider used, but may not work without additional modifications to LDAP settings used in relative few cases.

This setting is likely to be the default in future versions of the software.

Finally, this patch also adds deprecation warnings when features for compatibility with V2 attribute resolver scripting are used.

The following new properties have been added in this release (defaults in parentheses):

- idp.ldaptive.provider (%{org.ldaptive.provider:org.ldaptive.provider.jndi.JndiProvider})

This property defaults to any pre-existing Ldaptive system property setting the provider class, or failing that, defaults to the JNDI provider for compatibility with current behavior.

## 3.4.3 (January 9, 2019)

3 issues

This is a patch update containing a fix for a regression in the TemplateAttributeDefinition caused by a late decision to deprecate the `<SourceAttribute>` construct and a regression in the "renew" feature for CAS.

## 3.4.2 (December 19, 2018)

7 issues

This is a patch update containing bug fixes and addressing a security advisory.

The fix for this advisory introduced an incompatibility for those already using the impacted feature, involving CAS proxy trust configuration. If you have an existing configuration that defines a bean named **shibboleth.CASProxyTrustedCertificates**, usually defined in *cas-protocol.xml,* you will need to adjust the definition of this bean to directly expose the filenames containing the certificates as the values of the list bean instead of indirecting them through the factory bean used in older examples. The correct syntax has been updated in the example in CASProxyPKIXTrustSimple (and is shown in the file included with the distribution).

This patch includes additional deprecation warnings and an improvement to the handling of expired metadata at startup.

There is (yet another) spurious warning being produced new in this patch that can be ignored if you've removed all the legacy `<SourceAttribute>` elements from your configuration:

```
08:38:46.114 -  - WARN [DEPRECATED:118] - XML Element 'SourceAttribute', (file [/opt/shibboleth-idp/conf
/attribute-resolver.xml]): This will be removed in the next major version of this software; replacement is by
using <InputAttributeDefinition> and <InputDataConnector>
```

The warning appears incorrectly even after removal of the element(s).

There's also a regression involving this particular deprecated feature that is fixed in V3.4.3. In the meantime, upgraded systems may need to update and replace the deprecated `<Dependency>` elements in any TemplateAttributeDefinition constructs to avoid running into

**IDP-1386** - Getting issue details... STATUS .

## 3.4.1 (November 1, 2018)

8 issues

This is a patch update containing bug fixes.

Several spurious or outright incorrect warnings have been suppressed, and a pair of regressions impacting upgrades have been corrected. Some warning-related issues remain and are documented under the V3.4.0 release.

## 3.4.0 (October 10, 2018)

183 issues

This is a minor upgrade containing new features and bug fixes. New material in the documentation can be identified with a $^{3.4}$ superscript.

### Important Notes for Upgraders

A few issues were identified with the V3.4.0 release and are now corrected in V3.4.1 (see above); refer to the fixed issues list for details if necessary but "just use 3.4.1" and you can ignore them.

If you were improperly relying on the contents of the *idp.home/webapp* directory (which is built by the installer) to operate the software directly rather than using the built warfile, that was not a supported approach and it will not work with this release. As part of some related changes to the war build process, the upgrade will remove that directory (as it is considered to be "owned by the war build process") and if you have your own content in it, you should make sure it's present in the **edit-webapp** directory before upgrading. Do **not** copy any of our files or jars into that directory, it is **only** meant for your own additions or changed files.

A new feature supporting automatic injection of response headers such as Content-Security-Policy and Strict-Transport-Security requires the installation of a servlet filter into *web.xml*, installed by default for new installations and older systems that have not customized the *web.xml* file.

For upgrades, this could result in a change in behavior because the default property values result in denial of framing of most of the IdP's user interface, including the login form. This is consistent with the offiicial project stance not to support use of frames due to the impact on users that choose to disable third-party cookies. If you wish to, you can support frames by explicitly setting the relevant new properties (idp.frameoptions, idp.csp) to empty values.

If you have local customizations to *web.xml*, be sure to review the changes and if desired, add the DynamicResponseHeaderFilter declaration and mapping to your version.

A regression exists in V3.3.0 that impacts deployers who are relying on the "condition" flow error/warning feature of the Password login flow to handle an expiring password condition. To ensure correct behavior, the patch that fixes IDP-1101 should be applied. One of the corrections involves a user configuration file so will not be fully corrected during the next upgrade (and this note will be maintained for all subsequent releases).

If you use Windows and you have manually configured the **idp.home** property in an environment variable or Java servlet configuration system variable because of a non-standard installation location, make sure the path in the variable uses only forward-slashes and not backslashes. Use of backslashes in this path, or any path in any configuration file, is not supported and the IdP may not function properly. This is not a change, but since the IdP may have worked before despite this, and will not work now, it's being highlighted.

There's an unintentionally confusing warning in the log ("*Use of secure property is strongly advised*"); it's referring to the setting controlled by the **idp.cookie.secure** property, which defaults to false for backward compatibility but should be set to true in most cases.

In the very unlikely event that you are using an alternative JAXP XML Parser implementation, you may have relied on a technique for controlling some aspects of the parser's security settings via a property that we were forced to deprecate and turn off to prevent a future Java compatibility issue. The SystemRequirements page describes the alternative means of handling JAXP replacement for V3.4, though it remains unsupported officially.

The new support for the SAML HTTP-Artifact binding for inbound SAML 2 SSO and Logout requests requires a user-space configuration update for full functionality. A new bean has been defined to declare the default client TLS credential, **shibboleth.DefaultClientTLSCredential**. To enable client TLS authentication to an SP, this bean will need to be defined in *conf/credentials.xml.* Typically one would just declare this as an alias to the existing default signing credential (below). No configuration error will result if it is not declared. However if client TLS is indicated to be performed and the bean is not defined, a runtime error will likely result during artifact resolution with the SP. (Note that the default for artifact resolution to an SP over https on port 443 is to perform message signing rather than client TLS.)

```
<!-- Your IdP's default client TLS credential, by default the same as the default signing credential. -->
<alias alias="shibboleth.DefaultClientTLSCredential" name="shibboleth.DefaultSigningCredential" />
```

A CAS API component used for advanced proxy endpoint validation has been deprecated; this only affects deployers that wrote third-party extension components implementing that interface.

### New Objects

In addition to the lists below, a variety of beans and properties have been added in support of these features:

- Duo non-browser profile support
- FunctionAuthnConfiguration
- MetadataDrivenConfiguration
- AttendedRestartConfiguration
- ImpersonateInterceptConfiguration

The following new beans (or at least support/placeholders for them) have been added in this release:

- shibboleth.HTTPResource
- shibboleth.ResponseHeaderFilter
- shibboleth.DefaultResponseHeaderMap
- shibboleth.ResponseHeaderMap
- shibboleth.ResponseHeaderCallbacks
- shibboleth.X509TrustManager
- shibboleth.context-check.Function
- shibboleth.DefaultParserPool
- shibboleth.DefaultClientTLSCredential
- shibboleth.EncryptionConfiguration.CBC
- shibboleth.EncryptionConfiguration.GCM
- shibboleth.SecurityConfiguration.CBC
- shibboleth.SecurityConfiguration.GCM

The following new properties have been added in this release (defaults in parentheses):

- idp.hsts (max-age=0)
- idp.frameoptions (DENY)
- idp.csp (frame-ancestors 'none';)
- idp.entityID.metadataFile (%{idp.home}/metadata/idp-metadata.xml)
- idp.encryption.config (shibboleth.EncryptionConfiguration.CBC)
    - **Don't change this without testing, you likely have lots of SPs that don't support AES-GCM.**
- idp.consent.attribute-release.userStorageKey (shibboleth.consent.PrincipalConsentStorageKey)
- idp.consent.attribute-release.userStorageKeyAttribute (uid)
- idp.consent.terms-of-use.userStorageKey (shibboleth.consent.PrincipalConsentStorageKey)
- idp.consent.terms-of-use.userStorageKeyAttribute (uid)
- idp.replayCache.strict (true)
- idp.xml.parserPool (shibboleth.ParserPool)

- idp.audit.shortenBindings (false)

### New Features

There are a large number but some of the most significant worth reviewing include:

- HTTPConnector for web services integration in the attribute resolver (and all the various features that can be controlled via attributes)
- Extensive customization of behavior through new metadata "tag" conventions
- Non-browser Duo AuthAPI support for ECP clients
- Improvements for embedding inline scriptlets in a variety of configuration scenarios, reducing the need for custom Spring beans
- A new "attended startup" mode that prevents on-disk access to an unlocked or trivially encrypted private key
- The ability to provision CAS services using SAML metadata for consistency and to support the new metadata-driven configuration mechanisms
- Greatly enhanced context-check interceptor that can address multiple authorization scenarios at the same time
- A new impersonation interceptor that supports advanced debugging or testing scenarios
- Support for configurable trust for remotely accessed TLS-protected configuration resources and other HTTP client scenarios
- Support for key pinning of LDAP connections (i.e., verifying only the LDAP server key, not certificate)
- Support for inbound SAML 2 protocol requests for the SSO and logout profiles conveyed via the SAML 2 artifact binding.
- An ldaptive system property, org.ldaptive.response.encodeCntrlChars, can be set to prevent unruly characters from appearing in the IdP's log files due to Active Directory.

### Deprecation Warnings in Preparation for V4

It is expected that V3.4 will be the last minor version of V3 of the IdP.  It is a goal (though not an absolute promise) for V4 of the IdP that any configuration that loads *without warning* on V3.4 will load in V4.  To this end V3.4 has warnings against use of deprecated function. The complete list of deprecated configuration is listed here.

## 3.3.3.1 (June 27, 2018) (Windows Only)

This is a service release of the 3.3.3 Windows Installer that updates Jetty to 9.3.24.v20180605 to address Jetty security issues. If you did not install Jetty via the Shibboleth installer, then this update is not required (but of course you may still be affected by the security issues if you have an affected Jetty version in use).

As noted on the WindowsInstallation page, service releases (represented by the fourth version number) do not indicate an actual update to the Shibboleth software, only to third party components we support.

## 3.3.3 (May 16th, 2018)

4 issues

This is a patch upgrade containing an update to the Spring Framework to correct a Windows-only security issue and addressing this security advisory.

## 3.3.2 (October 4, 2017)

21 issues

This is a patch upgrade containing bug fixes and addressing this security advisory.

The Duo Java integration library in this release has also been updated to V1.3 and V2.6 of the Javascript UI.

### Important Notes for Upgraders

It was brought to our attention that many may be unaware of the fact that the IdP's cookie handling defaults to cookies without the "secure" attribute. We suggest most deployers modify the **idp.cookie.secure** property in *idp.properties* to account for this.

As an exception to our normal versioning policy, this patch update includes a small feature addition, support for Base32 encoding of computed identifiers in the attribute resolver and for "persistent" NameID generation. This feature, while not the default, is strongly recommended for new deployments of these features as outlined on the relevant documentation pages. The prevalence of unsafe application handling of case-sensitive identifiers motivated the project to make this feature available ahead of V3.4.

An additional, related change is the reversal of an unofficial intention to deprecate the two data connectors related to producing pairwise identifiers in support of an expected effort on the part of at least some federations to discourage use of NameIDs in favor of SAML Attributes, necessitating continued support for these connectors within the Attribute Resolver.

## 3.3.1.1 (March 23, 2017) (Windows Only)

This is a service release of the 3.3.1 Windows Installer that fixes an issue with new installs.( **IDP-1149** - Getting issue details... STATUS )

There is no other change, so users running 3.3.1 do ***not*** need to apply this update.

## 3.3.1 (March 15, 2017)

23 issues

This is a patch upgrade containing bug fixes and addressing this security advisory.

### Important Notes for Upgraders

A very small minority of deployers may need to take some simple additional steps in applying this update, as a result of changes required to address the security vulnerability. This only applies to deployers that have built custom login flows, interceptor flows, or subject canonicalization flows and are returning custom Events from those flows for error handling purposes, and the additional change is required to keep those customizations working.

In such cases, it's necessary to modify the corresponding abstract event flow definition files *(conf/authn/authn-events-flow.xml, conf/intercept/intercept-events-flow.xml,* or *conf/c14n/subject-c14n-events-flow.xml* respectively) and add transition rules for your custom event in addition to the end-states illustrated in prior versions. You can see examples of how to do this in the newly-delivered default files in the **dist** subdirectory and it will just take you a minute or two.

If you haven't ever modified the files mentioned above, you need not be concerned and the patch as applied is sufficient.

A regression exists in V3.3.0 that impacts deployers who are relying on the "condition" flow error/warning feature of the Password login flow to handle an expiring password condition. To ensure correct behavior, the patch that fixes IDP-1101 should be applied. One of the corrections involves a user configuration file so will not be fully corrected during the next upgrade (and this note will be maintained for all subsequent releases).

## 3.3.0 (November 10, 2016)

207 issues

This is a minor upgrade containing new features and bug fixes. New material in the documentation can be identified with a $^{3.3}$ superscript.

### Issues Identified Since Release

A regression exists in this release that impacts deployers who are relying on the "condition" flow error/warning feature of the Password login flow to handle an expiring password condition. To ensure correct behavior, the patch that fixes IDP-1101 should be applied. One of the corrections involves a user configuration file so will not be fully corrected during the next upgrade (and this note will be maintained for that release).

The example script included in the MFA flow's configuration contained an error demonstrating an incorrect way to obtain the right username to use for attribute lookup. The example has been corrected for future releases and there are better examples in the documentation to refer to.

### Important Notes for Upgraders

A fix to one of the logout features requires a modification to the views/logout.vm view template. See this patch/diff for details, or refer to the file provided with the new distribution.

A refactoring was done to merge the previously separate message property files in the *messages* directory into a single system-supplied message file with a single placeholder file for overriding individual messages. For compatibility reasons, editing the new *messages.properties* file will not affect the system unless you update *conf/services.xml* to load the message files you want it to load (you can see the new defaults in *dist/conf/services.xml.dist*). Doing so may require that you migrate existing customizations and will simplify future upgrades and support installing translations (plus it's a lot simpler to see your own customizations this way).

Improvements in back-button behavior result in some new exception types being raised. Mapping them to existing back-button behavior requires additions to one of the message property files:

```
NoSuchFlowExecutionException = stale
ExternalAuthenticationException = stale
```

A bug in the ComputedIDDataConnector meant that salts containg leading or trailing spaces, and provided **inline** in the attribute resolver would have them stripped (which was incompatible with V2). If you want to keep generating values based on a trimmed salt, you need to change the salt appropriately and trim it yourself.

A weird issue that caused responses to be issued with missing signatures was corrected. If you encounter sporadic problems in older versions with SPs complaining about missing signatures, the issue will be corrected once you update.

A change to the way relying party user interface information is being populated led to the factoring out of a single global setting to turn this on and off for all authentication and post-authentication interceptor flows. This eliminates support for controlling this on a per-login flow basis using flags set in individual login configuration files like *conf/authn/external-authn-config.xml* and others.

In support of some new, and likely future, REST APIs, the ability to use the less common HTTP methods (e.g. DELETE) with Spring WebFlow has been opened up. While this has no apparent impact on the functionality of existing flows (you can invoke a typical profile flow with DELETE if you really want to), we're aware that some extremely stupid security scanners assume that DELETE should return an error or "something bad is possible". The default `web.xml` descriptor provided with the software includes new security constraints that block most of the known methods except for the /profile/admin tree.

This release includes built-in support for Duo Security, and therefore includes files that could conflict with earlier third-party Duo extensions. If your installation includes its own version of the DuoWeb Java library, you may wish to remove *webapp/WEB-INF/lib/DuoWeb-1.1.jar* from the installation after upgrading and rebuild the warfile, to ensure that your own version (which should be in the **edit-webapp** tree) is used.

By default, the IdP now removes the in-memory copy of the user's password from the internal state of the request after validating it. If you need to retain the UsernamePasswordContext for the life of the request, you will need to add a bean to conf/authn/password-authn-config.xml named shibboleth.authn.Password.RemoveAfterValidation and set it to java.boolean.FALSE.

### New Objects

The following new user-space configuration files have been added in this release. They will be installed in their default form when you upgrade.

- conf/admin/*
- conf/authn/duo-authn-config.xml
- conf/authn/duo.properties
- conf/authn/mfa-authn-config.xml
- conf/intercept/expiring-password-intercept-config.xml
- conf/messages/messages.properties
- views/client-storage/*

The following new beans have been added in this release:

- shibboleth.AuditDateTimeFormat
- shibboleth.AuditDefaultTimeZone
- shibboleth.AuditFieldReplacementMap
- shibboleth.SuppressedEvents
- shibboleth.DefaultSuppressedEvents
- shibboleth.PrincipalSerializers
- shibboleth.DefaultPrincipalSerializers
- shibboleth.PrincipalSymbolics
- shibboleth.DefaultPrincipalSymbolics
- shibboleth.PredicateAccessControl
- shibboleth.ParentFlowRegistry
- shibboleth.FlowMap
- shibboleth.DefaultFlowMap
- shibboleth.FlowPatterns
- shibboleth.DefaultFlowPatterns
- shibboleth.DefaultRESTFlows
- shibboleth.RESTFlows
- shibboleth.StaticExplicitTrustEngine
- shibboleth.StaticPKIXTrustEngine
- shibboleth.AvailableAdminFlows
- shibboleth.AdminFlow
- shibboleth.c14n.attribute.PrincipalNameLookupStrategy
- shibboleth.StorageBackedAccountLockoutManager
- shibboleth.authn.Password.AccountLockoutManager
- shibboleth.authn.Password.RemoveAfterValidation
- shibboleth.authn.RemoteUser.externalAuthnPathStrategy
- shibboleth.authn.External.externalAuthnPathStrategy
- shibboleth.authn.X509.externalAuthnPathStrategy
- shibboleth.authn.SPNEGO.externalAuthnPathStrategy
- shibboleth.authn.MFA.*
- shibboleth.authn.Duo.*
- shibboleth.expiring-password.*
- shibboleth.metrics.*

The following new properties have been added in this release (defaults in parentheses):

- idp.authn.rpui (true)
- idp.persistentId.encodedSalt
- idp.storage.clientSessionStorageName (shib_idp_session_ss)
- idp.storage.clientPersistentStorageName (shib_idp_persistent_ss)
- idp.duo.*

### New Features

- IDP-962 : A new framework for multi-factor authentication workflows is available, see MultiFactorAuthnConfiguration.
- IDP-1013 : Official support for Duo iframe-based authentication, see DuoAuthnConfiguration.
- IDP-156 : Account lockout and a simple REST interface to remove lockout records
- IDP-961 : A new mechanism for enabling authentication to the IdP's own administrative functions.
- IDP-981 : Ability to support multiple protected paths when using External login methods
- IDP-887 : The Scripted Attribute Definition, DataConnector and Attribute Filters all gain a new script variable, "subject", with the Java Subject produced during authentication.
- IDP-926 : The ContextDerivedAttributeAttributeDefinition allows arbitrary user attributes to be derived from the IdP's environment and the Subject DerivedAttributeAttributeDefinition allows user atttributes to be derived from the Principals associated with an authenticated subject.
- IDP-966 : Supported mechanism for overriding built-in webflow locations and registering flows automatically from a plugin library
- IDP-813 : Remove the need for separate 'dc:', 'enc:', and 'ad:' namespaces in the attribute-resolver file. Additionally, the ordering requirements on sub-elements has been removed.
- Significant enhancements to the caching capability of the Dynamic metadata resolver, including caching across restarts
- Support for local file-based dynamic resolution of metadata

## 3.2.1.1 (June 23, 2016) (Windows Only)

This is a service release of the 3.2.1 Windows Installer that updates Jetty to V9.3. There is no impact (or importance) for anyone not using it to install an embedded version of Jetty. The reason for this release is to anticipate the impending sunsetting of Jetty 9.2 in the near future.

This update REQUIRES the use of Java 8. Jetty now requires Java 8 and you MUST be using Java 8 in order for this upgrade to function. The installer will note this but does not include Java and does not take responsibility for making sure your Java environment is going to work.

The upgrade process should maintain any customizations made to the Jetty environment via the supported property files. As always, if you need significant customizations you should be using a container you install and maintain separately.

## 3.2.1 (December 19, 2015)

15 issues

This is a patch update containing bug fixes.

### Important Notes for Upgraders

Because of the expected lag in delivering the next minor upgrade, the fixes for IDP-873 and IDP-880 include additions to the configuration that are not backward compatible with V3.2.0, an exception to our normal version policy regarding patch releases.

### New Objects

The following new beans have been added in this release:

- shibboleth.consent.attribute-release.AttributeDisplayOrder

## 3.2.0 (November 18, 2015)

183 issues

This is a minor upgrade containing new features and bug fixes. New material in the documentation can be identified with a $^{3.2}$ superscript.

### Important Notes for Upgraders

A major bug in the implementation of storage-backed SAML 2 persistent identifiers was addressed by significantly changing the implementation. The configuration is backward-compatible, but there are database definition considerations that need to be addressed as part of the upgrade to correct this issue. The new documentation includes information you should review if you're using this feature.

A new environment variable, IDP_BASE_URL, can be set to globally override the URL used to call the administrative flows from the command line tools. The default value has also been slightly adjusted to include the servlet context path, so it now defaults to "http://localhost/idp". If you have scripts that set the -u parameter to control this URL now, they may need to be adjusted (or may well no longer be needed). Note that using anything but localhost will generally require modifying *conf/access-control.xml*.

The new logout support is dependent on a copy of JQuery, now included in the war tree under a /js directory. An explicit copy is included to ensure clients are dependent only on web content included with the software (because browsers erroneously and dangerously do not verify the authenticity of the scripts they run), but this may necessitate occasional out of band notices that a new version should be inserted if security issues arise.

The SPNEGO feature makes use of a new user-prefs.vm view template for manipulating the auto-login cookie, and use of this feature in turn requires that you add some new message properties to *conf/messages/authn-messages.properties*. Failure to do so will result in errors due to the lack of messages the template depends on.

### New Objects

The following new user-space configuration files have been added in this release. They will be installed in their default form when you upgrade.

- conf/mvc-beans.xml
- conf/authn/spnego-authn-config.xml
- flows/user/prefs/prefs-flow.xml
- views/spnego-unavailable.vm
- views/user-prefs.vm
- views/user-prefs.js

The following new properties have been added in this release (defaults in parentheses):

- idp.service.failFast (false)
- idp.persistentId.useUnfilteredAttributes (true)
- idp.persistentId.dataSource
- idp.authn.resolveAttribute.filterActiveResults (true)
- idp.storage.htmlLocalStorage (false)
- idp.consent.expandedMaxStoredRecords (0)
- idp.consent.expandedStorageThreshold (1048576)
- idp.entityID.metadataFile (%{idp.home}/metadata/idp-metadata.xml)
- idp.webflow.timeout (30)
- idp.webflow.maxConversations (5)
- idp.authn.spnego.externalAuthnPath (/Authn/SPNEGO)
- idp.errors.excludedExceptions
- idp.errors.exceptionMappings

- idp.fticks.federation
- idp.fticks.algorithm (SHA-2)
- idp.fticks.salt

The following new beans have been added in this release:

- shibboleth.CustomViewContext (added in comment to *conf/global.xml*)
- shibboleth.ClientStorageServices (added to *conf/session-manager.xml*)
- shibboleth.IgnoredContexts (added to *conf/authn/authn-comparison.xml*)
- shibboleth.authn.Krb5.ServicePrincipal (added to *conf/authn/krb5-authn-config.xml)*
- shibboleth.authn.Krb5.Keytab (added to *conf/authn/krb5-authn-config.xml*)
- shibboleth.authn.Password.RetainAsPrivateCredential (added to *conf/authn/password-authn-config.xml*)
- shibboleth.authn.Password.ExtendedFlows (added to *conf/authn/password-authn-config.xml*)
- shibboleth.authn.Password.PrincipalOverride (added to *conf/authn/password-authn-config.xml*)
- shibboleth.authn.SPNEGO.EnforceRun (added to *conf/authn/spnego-authn-config.xml*)
- shibboleth.authn.SPNEGO.Krb5.RefreshConfig (added to *conf/authn/spnego-authn-config.xml*)
- shibboleth.authn.SPNEGO.Krb5.Realms (added to *conf/authn/spnego-authn-config.xml*)
- shibboleth.KerberosRealmSettings (added to *system/flows/authn/spnego-authn-beans.xml*)
- shibboleth.authn.X509.ClassifiedMessageMap (added to *conf/authn/spnego-authn-config.xml*)
- shibboleth.JDBCPersistentIdStore (added to *system/conf/saml-nameid-system.xml*)
- shibboleth.LogoutRequestAuditExtractors (added to *conf/audit.xml*)
- shibboleth.DefaultLogoutRequestAuditExtractors (added to *system/conf/audit-system.xml*)

### New Features

IDP-594: A new implementation of client-side data storage has been swapped in that is compatible with the prevous use of cookies, and allows a deployer to optionally enable new support for HTML local storage, which greatly expands the size of data that can be stored. In combination with that feature, it's possible to enable the session tracking properties related to logout while still avoiding the use of server-side state.

IDP-224: Our first true single logout implementation is now available, covering front-channel mixed SAML and CAS logout. Documentation on this feature will be developed subsequent to this release.

IDP-111: A new login flow supporting SPNEGO authentication with Kerberos has been added, thanks to a contribution by SWITCH.

IDP-114: The Kerberos login flow has been enhanced to support KDC verification using a service principal and keytab. New beans must be uncommented and configured to use this feature (see KerberosAuthnConfiguration).

IDP-624: The order in which attributes are displayed to the user during attribute release consent is now configurable.

A new velocity context "`attributeDisplayDescriptionFunction`" is available to the attribute release consent screens. This is the language browser sensitive content of the `<DisplayDescription>` declared for the attribute in *attribute-resolver.xml.* See VelocityVariables for more details.

IDP-661: Two new MDC keys are added in support for logging.  See the documentation.

IDP-808: The Filtering language has been simplified, allowing all parts to be specified in the same namespace thus obviating the need for afp: basic: and saml: (although the old syntax is still supported). In some cases the name for the Matcher of PolicyRule has been simplified. The complete mapping is given in AttributeFilterLegacyNameSpaceMapping.

IDP-774: All Velocity views gain a new context "`custom`" which is whatever is defined as the bean "`shibboleth.CustomViewContext`". Similarly, scripting subsystems gain a new injectable bean named "`customObject`" which is made available to scripts as the variable "`custom`". The custom syntax for the Scripted Attribute Definition, Scripted Data Connector and Scripted Matcher and Policy Rule are all extended to allow a new attribute "`customObjectRef`".

IDP-715: Plugins can add configuration by placing a Spring configuration file at /META-INF/net.shibboleth.idp/config.xml on the classpath for their jar.  All copies of this file which are discovered will be loaded into the root context.

IDP-821: The Password login flow has been enhanced with support for sending the user into other login flows instead of returning its own result, allowing the offer of stronger methods at the same time the password prompt is available. See the documentation on this "Extended Flow" feature.

IDP-840: F-TICKS logging is now explicitly supported.

IDP-852: The default logging configuration has been redesigned to make use of property variables. These changes will not be installed during upgrades but may be reviewed afterwards in case they're of interest.

The LDAP and RDBMS data connectors have been enhanced to avoid repeated attempts to connect to failed data sources for a configurable period of time to improve failover performance. The new `noRetryDelay` setting enables this feature.

### Miscellaneous Fixes

IDP-666: To enable internationalization of messages displayed to users, the charset used when parsing message source property files has been changed to UTF-8.

IDP-685: The `onlyIfRequired` attribute as supplied to the MappedAttributeInMetadata and AttributeInMetadata filters was wrongly defaulting to false. This has been changed and it now defaults to true.

IDP-729: The restriction on sourcing persistent NameID values from a released attribute has been fixed and now defaults to allowing unreleased attribute sources since the value is not exposed directly.

IDP-780: A regression was corrected so that SP requests for the "unspecified" AuthnContext class are ignored, consistent with V2 behavior. A bean was added to allow the set of ignored values to be configured for advanced cases or to override this change.

IDP-782: In attribute filter construction, AND and OR Matchers and PolicyRules can have a single child Matcher (or Rule)

IDP-785: A regression was corrected so that attributes with more than one compatible AttributeEncoder attached appear once for each encoder in the resulting SAML AttributeStatement.

JSE-15: The preferred way of specifying the backing file to the FileBackedHTTPResource is via the backingFile constructor parameter. The `resource` parameter still works but has been deprecated. See the documentation.

## 3.1.2 (July 1, 2015)

12 issues

Notable bugs which have been addressed are:

IDP-703: In previous releases, Failover data connectors did not work. This is fixed.

IDP-666: Allow non Iso-Latin-1 characters in message files

IDP-682: The ProfileRequestContext is now available to scripts as `profileContext`

## 3.1.1.2 (Mar 31, 2015) (Windows Only)

This is a service release of the 3.1.1 Windows Installer that fixes a bug (IDP-668) that was preventing proper upgrades of the installer. It is not a change to any of the supplied software, and is only relevant for new upgrades, or for anybody having problems with the upgrade process.

As part of this fix, it's important to note that any changes made directly to the **webapp** folder's contents after installation do not survive across upgrades. Any such changes must be made to the **edit-webapp** tree designed for that purpose.

## 3.1.1 (Mar 26, 2015)

16 issues

This is a bug fix release.

This release contains a fix for the issue described in the security advisory issued on March 26, 2015. Apart from upgrading, no other actions are required to address the issue.

A bug (IDP-646) was fixed where the maxValidityInterval of the RequiredValidUntil metadata filter was incorrectly interpreted in milliseconds rather than seconds if a duration was specified as a number rather than a duration string.

A bug (IDP-642) was fixed that prevented use of the schema validation metadata filter.

A bug (IDP-635) was fixed that ignored languages preferred by the browser when displaying attributes during consent to attribute release.

A bug (IDP-651) was fixed that prevented the idp.session.consistentAddress property from being turned off.

A bug (IDP-654) was fixed that prevented the use of configuration properties to set return attributes in the LDAP data connector configuration.

Several bugs in CAS protocol support were fixed: IDP-614 (integration), IDP-658 (error handling), IDP-659 (concurrency).

An improvement to the *messages/error-messages.properties* file was made in the way that runtime exception messages are rendered, so updating to the most recent version of this file is suggested, or alternatively just copying over the updated **runtime-error.message** property.

## 3.1.0 (Mar 10, 2015)

72 issues

This is a bug fix release that necessitated some API additions.

### Important Notes for Upgraders

This release corrects a bug in the handling of null or empty attribute values in the attribute resolver, and the fix was done in a way that is incompatible with some scenarios that V2 supported. This change is described in a few places in the documentation, including here. If you have data connectors, attribute defintions, and particularly scripts that rely on the support in V2 for null values embedded in results, you may experience issues and will need to make adjustments to your scripts to account for the new EmptyAttributeValue class that distinguishes these values from the rest.

### New Objects

The following new properties have been added in this release (defaults in parentheses):

- idp.consent.storageRecordLifetime (P1Y)
- idp.replayCache.StorageService (shibboleth.StorageService)

- idp.artifact.StorageService (shibboleth.StorageService)
- idp.attribute.resolver.LDAP.searchFilter ("(uid=$requestContext.principalName)")
- idp.service.attribute.resolver.maskFailures (true)
- idp.service.attribute.filter.maskFailures (true)
- idp.httpclient.useTrustEngineTLSSocketFactory (false)

The following new beans have been added in this release:

- shibboleth.AuthenticationPrincipalWeightMap (added to *conf/authn/general-authn.xml*)

### Miscellaneous Fixes

This is a bug fix release addressing IDP-573 which corrected a serious bug in the attribute resolver required the addition of new public APIs, necessitating a minor version change, but this is not a significant feature upgrade. A few new properties and Spring beans have been added, and these are denoted in the documentation with the superscript $^{3.1}$ to distinguish them. Anything so denoted will be ignored or fail if used with an earlier version. (This convention will be used going forward to denote anything introduced with new releases.)

New properties were added for configuring alternative storage services for the replay cache and artifact store for clustered deployments.

A new "map" bean was added to *conf/authn/general-authn.xml* to address IDP-602 and make it possible to apply more control over which SAML AuthenticationMethod/AuthnContextClassRef is returned from a login flow that supports more than one. A map of Principal objects to numeric weights is used to favor some over others. The default configuration now applies a weight of "1" to the "urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport" context class principal so that it is used in place of "urn:oasis:names:tc:SAML:2.0:ac:classes:Password" when both are potentially valid. You can add this bean from the delivered file into your configuration to incorporate this improvement.

Per IDP-580, a syntax introduced in V3.0 to declare `<security:TrustEngine>` elements inside `<metadata:MetadataProvider>` elements has been deprecated in favor of declaring the trust engine element directly within a `metadata:SignatureValidation` MetadataFilter, which is the only current filter plugin that supports such an object. The deprecated syntax will likely be removed promptly due to its limited usefulness and very recent introduction.

A bug (IDP-585) was fixed that prevented the use of caching in the attribute resolver. In conjunction with this fix, the deprecated `cacheResults` LDAP /RDBMS data connector attribute is no longer honored (and a warning emitted). The `<dc:ResultCache>` and `<dc:ResultCacheBean>` elements are now the only supported mechanism for configuring caching.

Several bugs (IDP-588) were fixed to support using server-side storage such as MySQL or other databases for storage of consent decisions.

Per IDP-560, the default/example view templates include a few improvements, so you may wish to review those changes if you have a previous install, as the original files will not be overwritten.

## 3.0.0.11 (Feb 25, 2015) (Windows Only)

This is a service release of the 3.0.0 Windows Installer that updates Jetty to 9.2.9.v20150224 to address a Jetty security issue. If you did not install Jetty via the Shibboleth installer, then this update is not required (but of course you may still be affected by the issue if you have an affected Jetty version in use).

As noted on the WindowsInstallation page, service releases (represented by the fourth version number) do not indicate an actual update to the Shibboleth software, only to third party components we support.

## 3.0.0 (Dec 22, 2014)

This is the first release of the third-generation Identity Provider software. The key documentation links are located on the IDP30 space Home page, such as SystemRequirements, Installation, and UpgradingFromV2 material.

This release should interoperate with all previous releases of Shibboleth and other software that supports the same standards. As a major upgrade, the list of issues fixed and features added is numerous and you should refer to the documentation itself for information on what's changed or new.

### Significant Behavioral Changes

Significant changes in behavior from previous releases:

- The default signing/digest algorithms in this version have changed from SHA-1 to SHA-256. This can be adjusted per-SP or globally, see Security Configuration.
- This version restores the original V2 defaults as pertains signing behavior, and defaults to signing responses and not assertions. This is now best practice with SAML, but later versions of V2 contain altered defaults that match what was (at the time) the best practice of signing the assertion instead. Signing the response mitigates attacks against XML Encryption, though in practice these mitigations are of little use unless SPs actually require signed responses, and few if any do so. Use what works for the SPs you have to interoperate with.
- A change was made to the process of selecting the format of NameIdentifier included in assertions. A `<NameIDFormat>` element in an SP's metadata containing "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified" is no longer evaluated when selecting the format to use. Selecting that format requires supplying a `nameIDFormatPrecedence` property in the RelyingPartyConfiguration (both the legacy and current formats allow this).