IdPv4アップデートに関する情報

Shibboleth SPのアップデートに関してはこちらをご覧ください。⇒SPv3アップデートに関する情報

Shibboleth IdPバージョン5がすでにリリースされております。V5へのアップデートに関してはこちらをご覧ください。⇒IdPv5アップデートに関する情報

- バージョン共通
  - アップデートの手順
- IdP 4.2.x から IdP 4.2.x へアップデートする場合の注意点
- IdP 4.1.x から IdP 4.2.x へアップデートする場合の注意点
- IdP 4.1.x から IdP 4.1.x へアップデートする場合の注意点
- IdP 4.0.x から IdP 4.1.x へアップデートする場合の注意点
- IdP 3.x.x から IdP 4.x.x へアップグレードする場合の注意点
  - Javaのバージョン
  - Javaコンテナ
  - 設定ファイルの移行について

※ 4.0.xから4.2.xにアップデートするような場合、途中のバージョンの注意点にも有効なものがありますので確認するようにしてください

バージョン共通

アップデートの手順

アップデート時には設定ファイルの変更点を手動で反映させなければならない場合があります。以下の注意事項をご確認ください。

/opt/shibboleth-idp/以下に存在しないファイル/ディレクトリはアップデート時に自動的に作成されますが、インストール後修正したファイルのほか、修正していないファイルも一切上書きはされませんので、新バージョンの内容を適宜反映してください。各自で修正していないファイルはdist/以下のファイルで上書きする、各自で修正したファイルは新バージョンでの変更点をマージする形になります。

反映しない場合、旧来の機能は変わらず動作することが保証されますが、新バージョン以降の新機能が（基盤となる機能がデフォルトで有効な場合でも、手動で有効化した場合でもいずれも）正しく動作することが保証されません。このため、将来的な新機能利用も見据えて、アップデート後でもかまいませんのでなるべく早く反映するようにしてください。

edit-webapp/ については扱いが特殊で、この配下のファイルについては dist/webapp/ 配下の変更点に従って更新してください。また、*.jar など edit-webapp/ に元々存在しないファイルは反映する必要がありません。基本的に差分のあるもののみ edit-webapp/ に配置されます。逆に、dist/webapp/ 配下のファイルと同一内容の edit-webapp/ のファイルについては、将来的な更新に備えて配置されたものとなります。edit-webapp/ 上のファイルを削除しても問題なく動作しますが、将来のアップデート時に動作しなくなる原因となりますので、削除しないようご注意ください。dist/webapp/ 上で削除されたファイルについては、他のファイルを更新する前提で安全に削除できます。

特に、edit-webapp/WEB-INF/web.xmlは頻繁に更新されていますので、当該ファイルが存在する場合はweb.xmlの変遷を参照の上必要な変更を反映してください。（dist/のほうは dist/webapp/WEB-INF/web.xml にあり、変更不要の場合はedit-webapp/以下には存在しません。存在していない場合は対応不要です。ただしJettyを使っている場合は技術ガイドで変更するよう案内しており、存在します。すなわちインストール後からの変更を適宜反映する必要があります。）

uApproveJPをインストールしている場合はsystem/以下の修正が元に戻ってしまうので、アップデート前に展開したディレクトリの当該ファイルを修正した上でアップデートを行うのがお勧めです。system/以下の修正箇所をパッチ形式にしたものを置いておきますので、展開したディレクトリにて適用してください。
uapprovejp3-system.patch

```
$ patch -p0 < .../uapprovejp3-system.patch
```

ただし、4.1.0以降ではファイルがライブラリの中に入っております。下記の手順で展開したディレクトリの webapp/WEB-INF/lib/idp-conf-impl-4.1.x.jarの中にある当該設定ファイルを更新してJARファイルを上書きしてください。

webapp/WEB-INF/lib/idp-conf-impl-4.1.2.jar
を適当な空のディレクトリでunzipしまして、中の
net/shibboleth/idp/conf/services-system.xml
につきましてid="shibboleth.AttributeFilterService"のbean定義の<constructor-arg name="strategy">を以下のように変更、

```
    <constructor-arg name="strategy">
        <bean class="jp.gakunin.idp.attribute.filter.spring.impl.AttributeFilterServiceStrategy"
            depends-on="shibboleth.AttributeRegistryService"
            p:transcoderRegistry-ref="shibboleth.AttributeRegistryService"
            id="ShibbolethAttributeFilter"/>
    </constructor-arg>
```

同じく中の
net/shibboleth/idp/flows/intercept/attribute-release-beans.xml
につきまして、id="IsConsentRequiredPredicate"のbean定義のclassを以下のように変更

```
<bean id="IsConsentRequiredPredicate"
    class="jp.gakunin.idp.consent.logic.impl.IsConsentRequiredPredicate" />
```

してください。再度全体をzipしてidp-conf-impl-4.1.2.jarというファイル名にして元のファイルを上書きしてください。

まず、プラグインを使用している場合にはそちらの最新版を確認し、あればそちらを先にアップデートしてください。

shibboleth-identity-provider-4.x.x.tar.gzパッケージを展開してそのディレクトリに移動します。

```
# tar zxvf shibboleth-identity-provider-4.?.?.tar.gz
# cd shibboleth-identity-provider-4.?.?
```

配布物の設定ファイルについて旧バージョンからの変更点を確認し、/opt/shibboleth-idp/ に既に存在するファイルについて適宜変更点を反映してください。もしくは、出力を任意のファイルに保存しておき、後日変更点を反映してください。

```
# diff -rb -x LICENSE.txt -x bin -x credentials -x doc -x idp_ant¥¥.log -x logs -x metadata -x system /opt/shibboleth-idp/dist/ .
```

4.1以降モジュール化された一部の設定ファイルはこのコマンドでは変更点が出力されません。後述のようにアップデート後、*.idpnewもしくは*.idpsaveというファイル名のファイルから差分を確認してください。

Jettyを停止後、インストールスクリプトを実行することでアップデートを行います。

```
# systemctl stop jetty
# bin/install.sh -Didp.conf.credentials.filemode=640 -Didp.conf.credentials.group=jetty
```

```
Buildfile: /root/shibboleth-identity-provider-4.1.2/bin/build.xml

install:
Source (Distribution) Directory (press <enter> to accept default): [/root/shibboleth-identity-provider-4.1.2] ?
[Enter]  ←入力なし
Installation Directory: [/opt/shibboleth-idp] ?
[Enter]  ←入力なし
INFO [net.shibboleth.idp.installer.V4Install:162] - Update from version 4.0.1 to version 4.1.2
INFO [net.shibboleth.idp.installer.BuildWar:103] - Rebuilding /opt/shibboleth-idp/war/idp.war, Version 4.1.2
INFO [net.shibboleth.idp.installer.BuildWar:113] - Initial populate from /opt/shibboleth-idp/dist/webapp to /opt/shibboleth-idp
/webpapp.tmp
INFO [net.shibboleth.idp.installer.BuildWar:92] - Overlay from /opt/shibboleth-idp/edit-webapp to /opt/shibboleth-idp/webpapp.tmp
INFO [net.shibboleth.idp.installer.BuildWar:125] - Creating war file /opt/shibboleth-idp/war/idp.war

BUILD SUCCESSFUL
Total time: 8 seconds
#
```

インストールスクリプトが正常に終了したらJettyを起動します。

```
# systemctl start jetty
```

アップデート後、以下のコマンドでバージョンが更新されていることを確認してください。

```
$ /opt/shibboleth-idp/bin/status.sh | grep idp_version
idp_version: 4.1.2
```



アップデートが完了しましたら、後日でもかまいませんので各種設定ファイルを最新版の書式に更新することをご検討ください。

- モジュール化されていない設定ファイルについては、上記diffコマンドの出力を元に差分を反映していない場合は、反映してください。
- モジュール化された設定ファイルについては、末尾に .idpnew が付くファイルを探し、元のファイルのカスタマイズ部分を確認した上で *.idpnew のファイルで置き換え、適切にカスタマイズ部分を反映させてください。
    - 末尾に .idpsave が付くファイルについては、ファイル名から .idpsave を除いたファイル（元ファイル）が存在しない場合は削除できます。ただしカスタマイズが失われている可能性がありますので削除前によく内容をご確認ください。元ファイルが存在する場合も同様です。

※ *.idpnewはモジュール化された設定ファイルに関するものしか生成されませんので、非モジュールの設定ファイルの差分は上述の従来通りの方法で dist/ 配下のものとの差分をご確認ください。

IdP 4.2.x から IdP 4.2.x へアップデートする場合の注意点

このバージョンに限ったことではありませんが、Jetty起動時のタイムアウトが60秒に設定されているようですので、環境により起動に時間がかかるようであれば設定変更をご検討ください。下記の例は起動スクリプトのタイムアウトを120秒、systemdのタイムアウトを150秒にする例です。

/etc/sysconfig/jetty に以下の行を挿入します。

```
 （省略）
JETTY_RUN=/opt/jetty-base/tmp
JETTY_STATE=/opt/jetty-base/tmp/jetty.state
JETTY_START_TIMEOUT=120
```

/etc/systemd/system/jetty.service に以下の行を挿入します。

```
 （省略）
User=jetty
Group=jetty
TimeoutStartSec=150

[Install]
 （省略）
```

こちらもこのバージョンに限ったことではありませんが、2022年6月10日以前の学認の技術ガイドを参照して構築した場合、ソースIPアドレスの詐称により情報漏洩の問題が発生する場合があります。該当する場合は以下のアナウンスを参照して設定を修正してください。
Back-Channelの設定を行っている場合はこちらを参照して設定を修正してください:
https://www.gakunin.jp/ml-archives/upki-fed/msg01494.html
2021年9月以前に構築したものの場合（この場合Back-Channel設定の有無によりません）はこちらもご参照ください:
https://www.gakunin.jp/ml-archives/upki-fed/msg01425.html

古いJetty 10には高負荷時のメモリリークの問題があるようですので、該当する場合には10.0.11以降を使用することをお勧めします。https://shibboleth.atlassian.net/browse/IDP-1969

---

logbackおよびSLF4Jライブラリの問題ですが、Jettyをバージョンアップすると

```
java.security.PrivilegedActionException: java.lang.ClassNotFoundException: ch.qos.logback.access.jetty.RequestLogImpl
```

のエラーで起動しなくなる場合があります。Jettyが要求するバージョンのlogback/slf4j各ライブラリが配置されていないことによる問題ですが、Shibboleth開発元では各ライブラリのバージョン指定を上書きするという対処をとっているようでございますのでここでもそちらの方法で案内します。
まず前提として、Jettyは同梱していないライブラリに対してもバージョン指定されているものがあり、こちらでライブラリを用意していても指定されたバージョンのものでなければ使用されません。
/opt/jetty-base/で提供しているバージョンのライブラリの使用を強制するために、/opt/jetty-base/start.d/idp-logging.ini の末尾に以下のような記述を追加します。以下はjetty-baseの9.4.1-20211028版の例ですが、/opt/jetty-base/lib/{logback,slf4j}/ 以下に配置されている実際のライブラリのバージョンを記述してください。

```
@@ -6,3 +6,8 @@

 # This seems to be needed in some cases to get early logging output.
 -Dlogback.configurationFile=resources/logback.xml
+
+# Override logback version pinned in jetty-home/modules/logback-impl.mod
+logback.version=1.2.6
+# Override SLF4J version pinned in jetty-home/modules/slf4j-api.mod
+slf4j.version=1.7.32
```

どのバージョンから提供された機能か不明ですが、アカウントロックアウト機能が提供するAPIにアクセスできるようにするにはbeanの移動が必要です。（アカウントロックアウトを有効化するだけでAPIは不要の場合は移動は不要と思われます）
移動が必要な場合は、conf/authn/password-authn-config.xml に定義されている id="shibboleth.authn.Password.AccountLockoutManager" のbeanを conf/global.xml に移動させてください。
詳細: https://shibboleth.atlassian.net/wiki/spaces/IDP4/pages/1294074654/AccountLockoutManagement

---

このバージョンに限ったことではありませんが、以前の3.xから運用していて4.xにバージョンアップした場合、secrets.propertiesファイルが存在しない状態となります。起動ごとにwarningが出ますが動作に支障はありません。もしV4の新書式へ移行しsecrets.propertiesファイルの利用を希望する場合は以下のページをご参照ください。
⇒credentials/secrets.propertiesの生成

IdP 4.1.x から IdP 4.2.x へアップデートする場合の注意点

4.1.xおよびそれ以前のAttribute-Based Subject C14N機能において、実装ミスによりePPN等のスコープが取り除かれていることの注意喚起が行われております。多くの運用には影響を与えないと思われますが、当該機能を利用している場合かつこの実装ミスに依存している場合は4.2以降へのアップデートによって思わぬ挙動変更に遭遇する可能性がありますので、下記対処を行ってください。以下リリースノートより：

> *For example, an eduPersonPrincipalName of “foo@example.org” would be returned as “foo”. This was easily fixed, but MAY impact existing behavior if the “broken” behavior were relied on. This can be remedied by adjusting the configuration to transform the scoped value back into an unscoped one but is something that could alter behavior following an upgrade until it’s addressed.*

4.1までに以下のIdPプラグインをご利用の場合は、4.2以降では古いバージョンのプラグインは動作しませんので、継続してご利用の場合は記載のバージョンへのアップデートが必要です。

- OIDC OP: 3.1.1
- DuoOIDC: 1.2.0
- OIDC Commons: 2.0.0
- TOTP: 1.0.1

4.1のプロパティ読み込み処理（以前お知らせしたとおり4.1からプロパティが複数のファイルに分割記述され、conf/以下のすべてのファイルが読み込み対象となることが期待されます）は、idp.properties と同名のファイルをスキップするというバグがありました。4.2以降はそのようなことはなくすべてのファイルが読み込まれますので、バックアップファイルを同名でconf/内に保存しているような場合に影響を受けると思われます。バックアップファイルなど読み込まれるべきでないファイルはconf/外に退避させるか拡張子を変更してください。

上記に関連して、4.2以降では同じキーのプロパティが複数宣言されている場合に警告をログに出力するようになりました。これも活用して4.1以降標準となったプロパティによる各種設定に問題がないことをご確認ください。

4.2でのその他の特筆すべき変更点ならびに新機能は以下のとおりです：

- inbound SAML profile interceptor flows関連の互換性に影響する修正
- ログアウトの挙動変更オプション
- ログイン画面の変更（新規インストール時のみ）

詳細はリリースノート原文および開発者からの月例お知らせをご参照ください。

遅れての情報提供ですが、アサーション内の属性の格納方法が正規化されました。これまで同一属性の複数属性値についてばらばらに格納される場合がありましたが、4.2.0以降同一XML要素内に記載されます。これによる影響はないはずですが、実装に問題のあるSPによっては挙動に違いが発生することが考えられます。

IdP 4.1.x から IdP 4.1.x へアップデートする場合の注意点

このバージョンに限ったことではありませんが、Windows MSI installerを使っている場合アップデート時に最大メモリ設定が変更されるという情報があります。Windowsをお使いの場合はアップデート後にチェックするようにしてください。
詳細: (Shibboleth Wiki) WindowsInstallation の"Always Check"

こちらもこのバージョンに限ったことではありませんが、技術ガイドでご案内しているようにsystemdでJettyの起動や停止を行っている場合、かなり低い確率で起動時に以下のエラーが出て起動できない場合があるという報告があります。（これはJetty 9.4の問題でJetty 10にはこの問題はありません）

```
Job for jetty.service failed because the control process exited with error code. See ”systemctl status jetty.service” and ”journalctl -xe” for details.
```

ログには特にエラー等出ず起動中に突然停止した状態になります。
このようになりましたら、/opt/java-base/tmp/jetty.state が存在するかどうか確認し、存在する場合は一旦削除してからJettyの起動をお試しください。
長期間稼働しているとこうなる確率が増えるようですがそれに限りません。
本問題はJetty停止時に上記jetty.stateが削除されないことが原因で、何故削除されないかは不明ながら、/etc/init.d/jettyに以下の行を加えて削除ののちに起動するようにすれば本問題は回避できます。

```
@@ -447,6 +447,7 @@
 ###################################################
 case ”$ACTION” in
   start)
+    rm -f $JETTY_STATE
     echo -n ”Starting Jetty: ”

     if (( NO_START )); then
```

IdP 4.0.x から IdP 4.1.x へアップデートする場合の注意点

/opt/shibboleth-idp/system/ディレクトリから多くのファイルが削除されました。
3.1およびそれ以前の配布物のweb.xmlを edit-webapp/WEB-INF/ に配置しており、その後の必要な更新を行っていない場合にはsystem/以下への参照が含まれているためエラーになるという情報があります。共通手順にも含めておりますがweb.xmlの変遷を参照し必要な変更を取り込んでください。特に `DeferPlaceholderFileSystemXmlWebApplicationContext` という文字列が含まれる場合はエラーになる可能性があります。

また、uApproveJPのインストール・アップデート方法に変更が生じております。詳しくは上記共通手順の注意書きをご参照ください。

4.1からモジュールおよびプラグインの機能が導入されております。以前のバージョンからアップデートした場合は自動的に有効化等されるため問題ありませんが、4.1.xおよびそれ以降を新規インストールした場合、モジュールを有効化しないと使えない機能がございますのでご注意ください。現在有効化されているモジュールは以下のコマンドで確認できます：

```
# /opt/shibboleth-idp/bin/module.sh -l
```

モジュールの例: 属性送信同意画面の設定(IdPv4)

4.1からはidp.propertiesの記述が別の複数のファイルに分割記述されるようになりました（例: authn/authn.properties）。ただし4.0.x以前からのアップデートの場合はそれらの記述は適切に無視されます。4.1流にプロパティを分散記述する場合は、idp.propertiesにある重複を排除した上でidp.searchForPropertiesをtrueに設定してください。詳しくは dist/conf/idp.properties の冒頭のコメントを参照してください。
4.1の書式に書き換える例: https://shibboleth.atlassian.net/wiki/spaces/KB/pages/1469908146/Example+4.1+Upgrade

---

4.1から新たにプロパティで指定できるようになった設定が多数ありますが、古い設定ファイルが邪魔をしてプロパティで指定しても適切に反映されない場合があります。適切にプロパティの値を反映させたい場合は、アップデート手順にあるように設定ファイルの更新を適切に反映させてください。

4.0.1までには属性値ハッシュ計算の問題で属性値が複数ある場合変更されていないのに再同意を求める問題（idp.consent.compareValues=trueの場合の問題）があるようです。本運用環境を移行する場合にはご注意ください。4.1.0で修正されました。
https://issues.shibboleth.net/jira/browse/IDP-1660

LDAPを使った属性取得の例が conf/examples/ の下に移動しました。（conf/examples/attribute-resolver-ldap.xml）

4.1以降では conf/session-manager.xml ファイルは不要になりました。もし当該ファイルの `shibboleth.ClientStorageServices` beanをカスタマイズして使っている場合は、当該beanを conf/global.xml に移したのち、conf/session-manager.xml を削除してください。

IdP 3.x.x から IdP 4.x.x へアップグレードする場合の注意点

下記ページの通りバージョン4.0.0が2020-03-11にリリースされました。下記の注意事項および手順をご確認の上、バージョンアップを行ってください。
https://wiki.shibboleth.net/confluence/display/NEWS/2020/03/11/Shibboleth+Identity+Provider+V4.0.0+Released

なお、バージョン3のEoLは2020年末です。未だバージョン3を運用している機関様におかれましては至急V4へのバージョンアップのスケジューリングをお願いします。
http://shibboleth.net/pipermail/announce/2020-March/000213.html

学認が提供している技術ガイドも順次更新していきます。uApproveJPはバージョン4対応版を公開中です。TiqrShib等のNIIが提供しているIdPプラグインは現在バージョン4対応版の公開準備中です。

また、アップデート時のノウハウなど情報をお持ちの方がいらっしゃいましたら、情報交換ML等で共有いただけましたら幸いです。

Javaのバージョン

- Java 11以上が必須です。
- 公式サポートはJavaのLTSのみなので、2020年9月現在では Java 11 のみとなります。

Javaコンテナ

- Shibboleth IdP V3で利用されていたTomcat 7はサポート対象外となりました。
- Shibboleth開発元がサポートしているのはJetty 9.4、またはTomcat 9以上（参照: SystemRequirements）
    - 開発元はJettyを推奨しています
    - 少なくともServlet 3.1をサポートしたJavaコンテナでなければ動作しません

設定ファイルの移行について

- 名前空間のフラット化が強制され、プレフィックスありの（v2由来の名前空間を使用した）設定ファイルが使用できなくなります
- IdP v3.4ではフラット化していない場合、DEPRECATEDのwarningとしてログに出力されます
    - 3.4.0以降も細かい改善が続けられていますので、v3.4系の最新版で確認してください
    - 静的に解析しているものと動的に解析しているものがあり、3.4系の最新版でしばらく動かして一通り認証フローの動作確認を行ってみることが必要です
    - フラット化の対応手順はこちら
      ⇒https://meatwiki.nii.ac.jp/confluence/x/F4Z7AQ
- その他にも、DEPRECATEDのwarningとなる対象が複数あります
    - 影響の大きいものとして、<Dependency>要素はDEPRECATEDです
        - 内容によって<InputAttributeDefinition>もしくは<InputDataConnector>で置き換えてください。例えば

```
<resolver:Dependency ref="myLDAP" />
```

は（myLDAPがDataConnectorとして定義されていると仮定して）以下で置き換えます。attributeNamesには使用する属性名を（複数ある場合はスペース区切りで）列挙してください:

```
<InputDataConnector ref="myLDAP" attributeNames="..." />
```

また、

```
<resolver:Dependency ref="eduPersonAffiliation" />
```

は（eduPersonAffiliationがAttributeDefinitionで定義されていると仮定して）以下で置き換えます:

```
<InputAttributeDefinition ref="eduPersonAffiliation" />
```

- LegacyPrincipalConnector
  以下のwarningが出る場合は

```
WARN [DEPRECATED:118] - Spring bean 'c14n/LegacyPrincipalConnector', (c14n/subject-c14n.xml): This will be removed in the next major version of this software; replacement is <remove>
```

conf/c14n/subject-c14n.xml の以下の部分を削除してください。

```
<!--
This is installed to support the old mechanism of using PrincipalConnectors in the attribute resolver to map SAML Subjects back into principals. If you don't use those (or this is a new install) you can remove this.
-->
<ref bean="c14n/LegacyPrincipalConnector" />
```

フラット化への対応により<PrincipalConnector>は削除されているはずですので、当該部分を削除しても問題ありません。
- 詳細はShibbolethの本家の情報を参照してください。
  - 変更のある要素への置き換え方法へのリンクを含めて記載されています
- **Shibboleth IdP V4への準備として、V3.4系最新版（3.4.8）にてwarningが出なくなるまで設定ファイルを修正することを推奨します**

- すでにフラット化とDEPRECATED対応済みの学認テンプレートを配布中ですのでこちらも参考にしていただけます。ただし最新版attribute-resolver.xmlテンプレートにはV4向けのAttribute Registry対応も入っておりますので、本ページ（V3からのアップグレード）の文脈では参考にしないでください。具体的に言うと、<AttributeEncoder>の行は消さないでください。あくまでフラット化とDEPRECATEDな要素・属性の置き換えの参考として参照してください。
- LDAP周りで、使用するライブラリがJNDIからUnboundIDに変更になることにより以下の通り設定によってはV4への移行後にエラーになる可能性が若干ございます。
  - LDAPのURL（ldap.propertiesの idp.authn.LDAP.ldapURL ）がスラッシュ(/)で終わる場合はうまく動作しませんのでスラッシュを除去してください。
  - 検索フィルタ（ldap.propertiesの idp.attribute.resolver.LDAP.searchFilter ）に空白が含まれるとActive Directory等との連携に問題が発生する場合がございますので、空白を除去してください。
  - LDAPConnectorもしくはJAASAuthnConfigurationにてJNDI特有のプロパティを使っている場合問題が発生します。プロパティ名に"jndi"を含むものもしくは下記"binary"にご注意ください。代替のものに置き換えてください。
    - 特にバイナリ属性（objectGUID等）については3.4.5よりLDAPConnectorにて<BinaryAttributes>要素がサポートされておりますのでこれで代替してください。プロパティ名は"java.naming.ldap.attributes.binary"となっております。
    - JNDI特有のプロパティとは、例えば、attribute-resolver.xmlの<DataConnector>に以下のような指定がある場合該当します。

```
<LDAPProperty name="com.sun.jndi.ldap.connect.timeout" value="500"/>
```

その他、IdPのディレクトリの中やJavaのディレクトリの中に jndi.properties というファイルが存在しその中で指定しているという場合があるようですのでご注意ください。
  - **上記問題の対象の場合は、V3.4系最新版で以下に記載されている手順でUnboundIDを使うようにして動作確認することを推奨します**
    - V3.4.4以降で以下の行をldap.propertiesに追加すればJNDIでなくUnboundIDを使うようになります。

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

    - V3系でUnboundIDが使われていることの確認は、idp.propertiesに

```
idp.loglevel.ldap=INFO
```

を追加してログレベルを変更の上再起動し、下記のように"Setting ldap provider to"が UnboundIDProvider になっていることを確認してください。

```
2020-03-02 09:46:50,446 -  - INFO [org.ldaptive.DefaultConnectionFactory:192] - Setting ldap provider to org.
ldaptive.provider.unboundid.UnboundIDProvider
2020-03-02 09:46:50,453 -  - INFO [org.ldaptive.DefaultConnectionFactory:192] - Setting ldap provider to org.
ldaptive.provider.unboundid.UnboundIDProvider
2020-03-02 09:46:50,453 -  - INFO [org.ldaptive.DefaultConnectionFactory:192] - Setting ldap provider to org.
ldaptive.provider.unboundid.UnboundIDProvider
```

確認後、idp.propertiesに追加した行を削除してログレベルを元に戻してください。

- 詳細はこちら: LDAPonJava (v4) および LDAPonJava>8 (v3)