REMOVED_Upgrading from the Previous Release

This section walks you through the steps you need to follow to upgrade from WSO2 Identity Server 5.6.0 to WSO2 Identity Server 5.7.0. In this section, <OLD_IS_HOME> is the directory that Identity Server 5.6.0 resides in, and <NEW_IS_HOME> is the directory that Identity Server 5.7.0 resides in.

Before you begin

This release is a WUM-only release. This means that there are no manual patches. You can use WSO2 Update Manager (WUM) to get any fixes or latest updates for this release.

- **If you are upgrading to use this version in your production environment**, use WSO2 Update Manager to get the latest updates available for WSO2 IS 5.7.0. For more information on how to use WSO2 Update Manager, see Updating WSO2 Products.

Migrating the embedded LDAP user store

WSO2 does not recommend using the embedded LDAP userstore that is shipped with WSO2 Identity Server in a production environment. However, if you want to upgrade your existing IS 5.6.0 LDAP userstore to IS 5.7.0 embedded LDAP, follow the instructions below:

- Copy the <OLD_IS_HOME>/repository/data directory to <NEW_IS_HOME>/repository/data directory.
- Restart the server to save the changes.

To upgrade to the latest version of WSO2 Identity Server, you need to upgrade the userstore database. Note that there are no registry schema changes between versions.

Follow the steps below to perform the upgrade.

It is recommended that you run the cleanup scripts before migration to clean the expired, inactive, and revoked tokens/codes. This reduces the time taken for migration.

1. Download Identity Server 5.7.0 and unzip it in the <NEW_IS_HOME> directory.
2. Take a backup of the existing database used by Identity Server 5.6.0. This backup is necessary in case the migration causes issues in the existing database.
3. Do the following database updates:
    a. Download the wso2is-migration-x.x.x.zip file from the latest release tag of the migration resources. Unzip it to a local directory.
    This folder is referred to as <IS_MIGRATION_TOOL_HOME>.
    b. Copy the org.wso2.carbon.is.migration-x.x.x.jar file in the <IS_MIGRATION_TOOL_HOME>/dropins folder into the <NEW_IS_HOME>/repository/components/dropins directory.
    c. Copy migration-resources directory to the <NEW_IS_HOME> root directory.
    d. Ensure that the following property values are as follows in the migration-config.yaml file found in the <NEW_IS_HOME>/migration-resources directory.

```
migrationEnable: "true"

currentVersion: "5.6.0"

migrateVersion: "5.7.0"
```

4. If you manually added any custom OSGI bundles to the `<OLD_IS_HOME>/repository/components/dropins` directory, copy those to the `<NEW_IS_HOME>/repository/components/dropins` directory.
5. If you manually added any JAR files to the `<OLD_IS_HOME>/repository/components/lib` directory, copy those and paste in the `<NEW_IS_HOME>/repository/components/lib` directory.
6. Copy the `.jks` files from the `<OLD_IS_HOME>/repository/resources/security` directory and paste in the `<NEW_IS_HOME>/repository/resources/security` directory.
7. If you have created tenants in the previous WSO2 Identity Server version and if there are any resources in the `<OLD_IS_HOME>/repository/tenants` directory, copy the content to the `<NEW_IS_HOME>/repository/tenants` directory.
8. If you have created secondary user stores in the previous WSO2 IS version, copy the content in the `<OLD_IS_HOME>/repository/deployment/server/userstores` directory to the `<NEW_IS_HOME>/repository/deployment/server/userstores` directory.
9. You can use one of the following approaches to migrate depending on your production evironment.

    - **Migrate by applying custom configurations to 5.7.0**

      This approach is recommended if:

      - You have done very few configuration changes in your previous version of WSO2 IS. These configuration changes have been tracked and are easy to redo.

      Steps:

      a. If you have made configuration changes to the config files in your previous version of WSO2 IS, update the files in the `<NEW_IS_HOME>/repository/conf` directory with your own configurations.
      b. Proceed to step 10 to run the migration client.

    - **Migrate by updating existing configurations with what's new in 5.7.0**

      This approach is recommended if:

      - You have done many configuration changes in your previous version of WSO2 IS.
      - These configurations have not been tracked completely and/or are difficult to redo.

      Steps:

      a. Make a copy of the `<OLD_IS_HOME>/repository/conf` directory. (Do not change the original configs. You may use it as a backup in case there are any issues)
      b. Copy the `health-check-config.xml` file from the `<NEW_IS_HOME>/repository/conf` directory and paste it in the copy of the `<OLD_IS_HOME>/repository/conf` directory.
      c. Copy the `wso2-log-masking.properties` file from the `<NEW_IS_HOME>/repository/conf` directory and paste it in the copy of the `<OLD_IS_HOME>/repository/conf` directory.
      d. The following table lists all the configuration changes from IS 5.6.0 to IS 5.7.0. You can scroll through the table and change the relevant configurations according to the features you are using.

Tip: Scroll left/right to view the entire table below.

Note: The configuration changes listed below will not affect the existing system because these configurations are applied only at first start up and new tenant creation.
If you wish to change the configurations for the existing tenants, configure it through the management console user interface.

Configuration File	Changes
`carbon.xml` file stored in the `<IS_HOME>/repository/conf` directory.	Change the version property value to 5.7.0. `<Version>5.7.0</Version>`

`axis2.xml` file stored in the `<IS_HOME>/repository/conf/axis2` directory.	Change the following property values to 5.7.0.
	`<parameter name="userAgent" locked="true">`
	`WSO2 Identity Server-5.7.0`
	`</parameter>`
	`<parameter name="server" locked="true">`
	`WSO2 Identity Server-5.7.0`
	`</parameter>`

`application-authentication.xml` file stored in the `<IS_HOME>/repository/conf/identity` directory.	Under `<Extensions>`, do the following changes to enable adaptive authentication: - Change the value of `<StepBasedSequenceHandler>` from `org.wso2.carbon.identity.application.authentication.framework.handler.sequence.impl.DefaultStepBasedSequenceHandler` to `org.wso2.carbon.identity.application.authentication.framework.handler.sequence.impl.GraphBasedSequenceHandler`. - Change the value of `<StepHandler>` from `org.wso2.carbon.identity.application.authentication.framework.handler.step.impl.DefaultStepHandler` to `org.wso2.carbon.identity.application.authentication.framework.handler.step.impl.GraphBasedStepHandler`.

	Add the following configuration under <AuthenticatorConfigs>: `<AuthenticatorConfig name="IdentifierExecutor" enabled="true">   <Parameter name="ValidateUsername">false</Parameter> </AuthenticatorConfig>`
`identity.xml` file stored in the `<IS_HOME>/repository/conf/identity` directory.	Add the following configuration under `<OAuth>`: `<!-- Token cleanup feature config to clean IDN_OAUTH2_ACCESS_TOKEN table--> <TokenCleanup> <!--If true old access token cleaning feature is enabled --> <EnableTokenCleanup>true</EnableTokenCleanup> <!--If true old access token retained in audit table --> <RetainOldAccessToken>true</RetainOldAccessToken> </TokenCleanup>` This configuration is required to clean the `IDN_OAUTH2_ACCESS_TOKEN` table.
	Under `<OAuth>`, change the value of `<OAuth2DCREPUrl>` from `${carbon.protocol}://${carbon.host}:${carbon.management.port}/api/identity/oauth2/dcr/v1.0/register` to `${carbon.protocol}://${carbon.host}:${carbon.management.port}/api/identity/oauth2/dcr/v1.1/register`. This reflects the DCR version update.

Do the following changes under <SupportedResponseTypes> to replace the deprecated `TokenResponseTypeHandler` class: - Change `<SupportedResponseType><ResponseTypeHandlerImplClass>org.wso2.carbon.identity.oauth2.authz.handlers.TokenResponseTypeHandler</ResponseTypeHandlerImplClass>` to `<SupportedResponseType><ResponseTypeHandlerImplClass>org.wso2.carbon.identity.oauth2.authz.handlers.AccessTokenResponseTypeHandler</ResponseTypeHandlerImplClass>`. - Change `<SupportedResponseType><ResponseTypeHandlerImplClass>org.wso2.carbon.identity.oauth2.authz.handlers.TokenResponseTypeHandler</ResponseTypeHandlerImplClass>` to `<SupportedResponseType><ResponseTypeHandlerImplClass>org.wso2.carbon.identity.oauth2.authz.handlers.IDTokenResponseTypeHandler</ResponseTypeHandlerImplClass>`. - Change `<SupportedResponseType><ResponseTypeHandlerImplClass>org.wso2.carbon.identity.oauth2.authz.handlers.TokenResponseTypeHandler</ResponseTypeHandlerImplClass>` to `<SupportedResponseType><ResponseTypeHandlerImplClass>org.wso2.carbon.identity.oauth2.authz.handlers.IDTokenTokenResponseTypeHandler</ResponseTypeHandlerImplClass>`.

Under `<SSOService>`, add the following SAML2 artifact validity period configuration:

```
<SAML2ArtifactValidityPeriodInMinutes>4</SAML2ArtifactValidityPeriodInMinutes>
```

Under `<SCIM>`, add the following configuration that allows you to get all the details of a user from SCIM endpoint if necessary:

```
<ShowAllUserDetails>false</ShowAllUserDetails>
```

Add the following configuration that is introduced to support filtering roles when you have configured a service provider role mapping:

```
<!--
         This configuration is used to filter
the SP configured role mappings. If the
property value is,

         true : If SP role mappings are
configured, returns only the mapped SP roles.
If SP role mappings are not
         configured returns all the mapped
local roles.

         false : If SP role mappings are
configured, returns mapped SP roles for the
mapped roles and the local mapped
         roles for others. If SP role mappings
are not configured returns all the mapped
local roles.

         Default - false.
     -->

    <!--SPRoleManagement>
        <ReturnOnlyMappedLocalRoles>false<
/ReturnOnlyMappedLocalRoles>
    </SPRoleManagement-->
```

Add the following configuration that allows you to customize the default user interfaces displayed at the time of just-in-time provisioning:

```
<JITProvisioning>
     <UserNameProvisioningUI>
/accountrecoveryendpoint/register.do<
/UserNameProvisioningUI>
     <PasswordProvisioningUI>
/accountrecoveryendpoint/signup.do<
/PasswordProvisioningUI>
</JITProvisioning>
```

Add the following configuration to include post authentication handlers introduced via JIT provisioning improvements:

```
<!-- Post Authentication handlers for JIT
provisioning, association and for handling
subject identifier -->
<EventListener type="org.wso2.carbon.identity.
core.handler.AbstractIdentityHandler"
                        name="org.wso2.carbon.
identity.application.authentication.framework.
handler.request.impl.
JITProvisioningPostAuthenticationHandler"
                        orderId="20" enable="
true"/>
<EventListener type="org.wso2.carbon.identity.
core.handler.AbstractIdentityHandler"
                        name="org.wso2.carbon.
identity.application.authentication.framework.
handler.request.impl.
PostAuthAssociationHandler"
                        orderId="25" enable="
true"/>
<EventListener type="org.wso2.carbon.identity.
core.handler.AbstractIdentityHandler"
                        name="org.wso2.carbon.
identity.application.authentication.framework.
handler.request.impl.
PostAuthenticatedSubjectIdentifierHandler"
                        orderId="30" enable="
true"/>
```

Do the following changes under `<ResourceAccessControl>`: - To reflect the changes introduced via security advisory WSO2-2018-0462, replace the following set of resources

```
<Resource context="(.*)/api/identity/user/v1.0
/validate-code" secured="true" http-method="
all"/>
<Resource context="(.*)/api/identity/user/v1.0
/resend-code" secured="true" http-method="all"
/>
<Resource context="(.*)/api/identity/user/v1.0
/me" secured="true" http-method="POST"/>
<Resource context="(.*)/api/identity/user/v1.0
/me" secured="true" http-method="GET"/>


<Resource context="(.*)/api/identity/consent-
mgt/v1.0/consents/purposes" secured="true"
http-method="POST">
<Resource context="(.*)/api/identity/consent-
mgt/v1.0/consents/pii-categories" secured="
true" http-method="POST">
<Resource context="(.*)/api/identity/consent-
mgt/v1.0/consents/purpose-categories" secured="
true" http-method="POST">


<Resource context="(.*)/scim2/Users" secured="
true" http-method="POST">
<Resource context="(.*)/scim2/Groups" secured="
true" http-method="POST">
<Resource context="/scim2/Bulk" secured="
true"  http-method="all">
```

with the following:

```
<Resource context="(.*)/api/identity/user/v1.0
/validate-code(.*)" secured="true" http-
method="all">
            <Permissions>/permission/admin
/manage/identity/identitymgt</Permissions><
/Resource>
<Resource context="(.*)/api/identity/user/v1.0
/resend-code(.*)" secured="true" http-method="
all"/>
<Resource context="(.*)/api/identity/user/v1.0
/me(.*)" secured="true" http-method="POST"/>
<Resource context="(.*)/api/identity/user/v1.0
/me(.*)" secured="true" http-method="GET"/>


<Resource context="(.*)/api/identity/consent-
mgt/v1.0/consents/purposes(.*)" secured="true"
http-method="POST"/>
<Resource context="(.*)/api/identity/consent-
mgt/v1.0/consents/pii-categories(.*)" secured="
true" http-method="POST"/>
<Resource context="(.*)/api/identity/consent-
mgt/v1.0/consents/purpose-categories(.*)"
secured="true" http-method="POST"/>


<Resource context="(.*)/scim2/Users(.*)"
secured="true" http-method="POST"/>
<Resource context="(.*)/scim2/Groups(.*)"
secured="true" http-method="POST"/>
<Resource context="(.*)/scim2/Bulk(.*)"
secured="true"  http-method="all"/>
```

- Replace `<Resource context="(.*)/api/identity/recovery/(.*)" secured="true" http-method="all" />` with the following resource:

```
<Resource context="(.*)/api/identity/recovery/
(.*)" secured="true" http-method="all">
    <Permissions>/permission/admin/manage
/identity/identitymgt</Permissions>
</Resource>
```

This introduces changes done with regard to access permission for account recovery endpoint.

- Add the following resource that allows using /api/identity/auth/ to retrieve data from authentication endpoint itself instead of obtaining via the URL:

```
<Resource context="(.*)/api/identity/auth/(.
*)" secured="true" http-method="all"/>
```

- To reflect the DCR version upgrade, replace the following set of resources

```
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.0/register(.*)" secured="true" http-
method="POST">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/create</Permissions>
</Resource>
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.0/register(.*)" secured="true" http-
method="DELETE">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/delete</Permissions>
</Resource>
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.0/register(.*)" secured="true" http-
method="PUT">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/update</Permissions>
</Resource>
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.0/register(.*)" secured="true" http-
method="GET">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/view</Permissions>
</Resource>
```

with the following:

```
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.1/register(.*)" secured="true" http-
method="POST">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/create</Permissions>
</Resource>
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.1/register(.*)" secured="true" http-
method="DELETE">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/delete</Permissions>
</Resource>
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.1/register(.*)" secured="true" http-
method="PUT">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/update</Permissions>
</Resource>
<Resource context="(.*)/api/identity/oauth2/dcr
/v1.1/register(.*)" secured="true" http-
method="GET">
    <Permissions>/permission/admin/manage
/identity/applicationmgt/view</Permissions>
</Resource>
```

Add the following property that was introduced to restrict federated user association done via the `UserProfileAdmin` admin service:

```
<!--
This property restricts federated user
association done through UserProfileAdmin
admin service.
Would not affect associations done through
provisioning
-->
<EnableFederatedUserAssociation>false<
/EnableFederatedUserAssociation>
```

Under `<TenantContextsToRewrite>` `<WebApp>`, replace `<Context>/api/identity/oauth2/dcr/v1.0/</Context>` with `<Context>/api/identity/oauth2/dcr/v1.1/</Context>` to reflect the DCR version upgrade.

Under `<AdaptiveAuth><EventPublisher>`, replace `<receiverURL>http://localhost:8280/>` with the following configuration:

```
<ReceiverURL>https://localhost:8280/<
/ReceiverURL>
<BasicAuthentication>
         <Enable>true</Enable>
          <Username>admin</Username>
          <Password>admin</Password>
</BasicAuthentication>
```

This introduces the default configurations for event publisher.

Under `<AdaptiveAuth>`, add the following configurations introduced to support external analytics calls in adaptive authentication:

```
<MaxTotalConnections>20</MaxTotalConnections>
<MaxTotalConnectionsPerRoute>20<
/MaxTotalConnectionsPerRoute>

<!--Timeouts in milliseconds-->
<!--Default configs for timeouts-->
<!--<HTTPConnectionTimeout>5000<
/HTTPConnectionTimeout>-->
<!--<HTTPReadTimeout>5000</HTTPReadTimeout>-->
<!--<HTTPConnectionRequestTimeout>5000<
/HTTPConnectionRequestTimeout>-->
<!--End of default configs for timeouts-->

<!--<RefreshInterval>500</RefreshInterval>-->
<!--End of timeouts in milliseconds-->

<!--<PromptOnLongWait>false<
/PromptOnLongWait>-->
<!--Timeout in milliseconds for the waiting
external calls-->
<LongWaitTimeout>10000</LongWaitTimeout>
```

`carbon.xml` file stored in the `<IS_HOME>/repository/conf` directory.	Add the following configuration that introduces parameters related to Carbon Crypto Service, which is a crypto framework used inside Carbon products:
	
	`<!--`
	` Configurations related to Carbon Crypto Service which is a crypto framework used inside Carbon products.`
	`-->`
	`<CryptoService>`
	
	` <Enabled>true</Enabled>`
	
	` <!-- The crypto provider which is used for internal data encryption and decryption -->`
	` <InternalCryptoProviderClassName>org.wso2.carbon.crypto.provider.KeyStoreBasedInternalCryptoProvider</InternalCryptoProviderClassName>`
	
	` <!--`
	`  The crypto provider which is used for the crypto needs which come when communicating with external parties.`
	`  e.g. Signing, Decrypting.`
	` -->`
	` <ExternalCryptoProviderClassName>org.wso2.carbon.core.encryption.KeyStoreBasedExternalCryptoProvider</ExternalCryptoProviderClassName>`
	` <!--`
	`  The list of key resolvers which will be used based on the context when handling crypto with external parties.`
	`  e.g. Resolving the public key of an external entity.`
	` -->`
	` <KeyResolvers>`
	`  <KeyResolver className="org.wso2.carbon.crypto.defaultProvider.resolver.ContextIndependentKeyResolver" priority="-1"/>`
	` </KeyResolvers>`
	
	`</CryptoService>`

Under <Security>, add the following keystore parameters introduced to encrypting/decrypting internal data:

```
<!--
  The KeyStore which is used for encrypting
/decrypting internal data.
  This block is read by Carbon Crypto Service.
-->
<InternalKeyStore>
  <!-- Keystore file location-->
  <Location>${carbon.home}/repository/resources
/security/wso2carbon.jks</Location>
  <!-- Keystore type (JKS/PKCS12 etc.)-->
  <Type>JKS</Type>
  <!-- Keystore password-->
  <Password>wso2carbon</Password>
  <!-- Private Key alias-->
  <KeyAlias>wso2carbon</KeyAlias>
  <!-- Private Key password-->
  <KeyPassword>wso2carbon</KeyPassword>
</InternalKeyStore>
```

`log4j.properties` file stored in the `<IS_HOME>/repository/conf` directory.	Add the following lines that include the properties introduced to support masking sensitive information in your logs:
	`# Log masking configuration. Please uncomment the following log4j property, if you need to mask any`
	`# information in your carbon logs.`
	`# When enabled, the logs will be matched with the provided patterns and masked .`
	`# The 'path-to-masking-patterns' path should be an absolute file path to a properties file. This file should contain`
	`# the patterns that should be checked for masking as key value pairs. (mask-name=masking-regex-pattern)`
	`# If this file cannot be found, wso2-log-masking.properties file will be used as default. If the following`
	`# configuration is not enabled, no masking process will be applied.`
	`#log4j.appender.CARBON_CONSOLE.maskingPatternFile=path-to-masking-patterns`
	
	`# Log masking configuration. Please uncomment the following log4j property, if you need to mask any`
	`# information in your carbon logs.`
	`# When enabled, the logs will be matched with the provided patterns and masked .`
	`# The 'path-to-masking-patterns' path should be an absolute file path to a properties file. This file should contain`
	`# the patterns that should be checked for masking as key value pairs. (mask-name=masking-regex-pattern)`
	`# If this file cannot be found, wso2-log-masking.properties file will be used as default. If the following`
	`# configuration is not enabled, no masking process will be applied.`
	`#log4j.appender.CARBON_LOGFILE.maskingPatternFile=path-to-masking-patterns`

e. Replace the `<NEW_IS_HOME>/repository/conf` directory with the modified copy of the `<OLD_IS_HOME>/repository/conf` directory.
f. Proceed to step 10 to run the migration client.

10. Start the Identity Server 5.7.0 with the following command to perform the data migration for all components.

a. Linux/Unix:

```
sh wso2server.sh -Dmigrate -Dcomponent=identity
```

b. Windows:

```
wso2server.bat -Dmigrate -Dcomponent=identity
```

11. Once the migration is successful, stop the server and start using the appropriate command.
    a. Linux/Unix:

```
sh wso2server.sh
```

    b. Windows:

```
wso2server.bat
```