# REMOVED_Upgrading from the Previous Release

We recommend migrating directly to one of the latest stable releases of WSO2 Identity Server (i.e., version 5.6.0 or a later version). For instructions on migrating directly to a later version, see REMOVED_Upgrading From an Older Version of WSO2 IS.

The following instructions guide you through upgrading from WSO2 Identity Server 5.2.0 to WSO2 Identity Server 5.3.0. In this topic, `<OLD_IS_HOME>` is the directory that Identity Server 5.2.0 resides in and `<NEW_IS_HOME>` is the directory that Identity Server 5.3.0 resides in.

## Before you begin

1. This release is a WUM-only release. This means that there are no manual patches and any further fixes or latest updates for this release can be updated through the WSO2 Update Manager (WUM).
   - **If you are upgrading to this version to use this version in your production environment**, use the WSO2 Update Manager and get the latest available updates for WSO2 IS 5.3.0. For more information on how to do this, see Updating WSO2 Products.
   - **If you are upgrading to this version only to do an incremental upgrade to the next available version** (e.g., if you are upgrading from WSO2 IS 5.2.0 - 5.4.0), you can skip this step and migrate to 5.3.0 by following the steps given in this document. You do not need to use WUM in this instance because the WUM updates available for this version will be included in the WSO2 IS pack of the next version.
2. If you have added any custom claims, expand the section below and follow the steps before migrating to WSO2 IS 5.3.0.

   This is required because all claims external to the WSO2 dialect in WSO2 IS 5.3.0 are mapped to the relevant claim in the WSO2 dialect and not to the underlying attribute in the user store. When there are custom claims, there is no claim in the WSO2 dialect that is mapped to that attribute. Therefore, follow the steps below to create a new claim in the WSO2 dialect and map your custom claim to the local claim (i.e., the new claim created in the WSO2 dialect).

   a. Start the WSO2 IS server of IS 5.2.0 and login to the management console.
   b. Click on **Add** under **Claims** on the **Main** tab of the management console.
   c. Click **Add New Claim** and select the http://wso2.org/claims dialect.
   d. Enter the required information of the custom claim. For more information, see Adding Claim Mapping in IS 5.2.0.
   e. Click **Add**. The claim you created will be listed.
   f. Click on **List** under **Claims** on the **Main** tab of the management console again.
   g. Click on the claim dialect where you have your custom claim, and click on the **Edit** button of your custom claim.
   h. Map the local claim you just created to the custom claim by editing the **Mapped Attribute(s)** field.
   i. Click **Update**.

   **Note:** Repeat the steps above for every custom claim you have created.

## Migrating the embedded LDAP user store

It is not generally recommended to use the embedded LDAP user store that is shipped with WSO2 Identity Server in production setups. However, if migration of the embedded LDAP is required, follow the instructions below to migrate the existing IS 5.2.0 LDAP user store to IS 5.3.0.

1. Copy the `<OLD_IS_HOME>/repository/data` folder to `<NEW_IS_HOME>/repository/data` folder.
2. Restart the server to save the changes.

To upgrade the version of WSO2 Identity Server, the user store database should be upgraded. Note that there are no registry schema changes between versions.

Follow the steps below as needed to complete the migration process.

⚠ 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 5.3.0 and unzip it in the `<NEW_IS_HOME>` directory.
2. Take a backup of the existing database used by Identity Server 5.2.0. This backup is necessary in case the migration causes issues in the existing database.
3. Make the database script updates as indicated below.
    i. Download the migration resources and unzip it to a local directory. This folder is referred to as `<IS5.3.0_MIGRATION_TOOL_HOME>`.
    ii. Copy the DB script files in the `<IS5.3.0_MIGRATION_TOOL_HOME>/dbscripts/identity` directory to the `<NEW_IS_HOME>/dbscripts/identity/migration-5.2.0_to_5.3.0/` directory.
    iii. Copy the `org.wso2.carbon.is.migrate.client-5.3.0.jar` file in the `<IS5.3.0_MIGRATION_TOOL_HOME>/dropins` directory to the `<NEW_IS_HOME>/repository/components/dropins` directory.
    iv. Alternatively, if you are using Oracle database, you can either provide the database owner credentials in the datasource configurations (identity and user management databases) or pass the identity database owner name with `-DidentityOracleUser` and user management database owner name with `-DumOracleUser`.
4. Copy any custom OSGI bundles that were added manually from the `<OLD_IS_HOME>/repository/components/dropins` folder and paste it in the `<NEW_IS_HOME>/repository/components/dropins` folder.
5. Copy any added JAR files from the `<OLD_IS_HOME>/repository/components/lib` folder and paste it in the `<NEW_IS_HOME>/repository/components/lib` folder.
6. Copy the `.jks` files from the `<OLD_IS_HOME>/repository/resources/security` folder and paste them in `<NEW_IS_HOME>/repository/resources/security` folder.
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. The ClaimManagementService API is not recommended for use with WSO2 IS 5.3.0. If you are using the ClaimManagementService API and have written any clients using the service, convert the clients to the new and improved ClaimMetadataManagementService API that is packaged with WSO2 IS 5.3.0.
10. You can use one of the following approaches to migrate depending on your production environment.

    - **Migrate by applying custom configurations to 5.3.0**

**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 configurations in the config files of your previous version of WSO2 IS, reconfigure the files in the `<NEW_IS_HOME>/repository/conf` folder with your configurations.
b. Proceed to step 11 to run the migration client.

- **Migrate by updating the existing configurations with what's new in 5.3.0**

**This approach is recommended if:**

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

**Steps:**

a. Make a copy of the `<OLD_IS_HOME>/repository/conf` folder. (Do not change the original configs. You may use it as a backup in case there are any issues)
b. Copy the following files from the `<NEW_IS_HOME>/repository/conf/identity` folder and paste it into the copy of the `<OLD_IS_HOME>/repository/conf/identity` folder:
    - `captcha-config.properties`
    - `identity-event.properties`

c. Open the `output-event-adapters.xml` file found in the `<NEW_IS_HOME>/repository/conf` folder and configure the relevant email configurations.

```
<adapterConfig type="email">
    <!-- Comment mail.smtp.user and mail.smtp.password
properties to support connecting SMTP servers which use trust
        based authentication rather username/password
authentication -->
    <property key="mail.smtp.from">abcd@gmail.com</property>
    <property key="mail.smtp.user">abcd</property>
    <property key="mail.smtp.password">xxxx</property>
    <property key="mail.smtp.host">smtp.gmail.com</property>
    <property key="mail.smtp.port">587</property>
    <property key="mail.smtp.starttls.enable">true</property>
    <property key="mail.smtp.auth">true</property>
    <!-- Thread Pool Related Properties -->
    <property key="minThread">8</property>
    <property key="maxThread">100</property>
    <property key="keepAliveTimeInMillis">20000</property>
    <property key="jobQueueSize">10000</property>
</adapterConfig>
```

**Tip:** This email configuration is similar to the email configuration shown in the code block below, which is found in the `<IS_HOME>/repository/conf/axis2/axis2.xml` file. This configuration is used for email-enabled features. You can configure the same values in the `output-event-adapters.xml` file for email-enabled features using REST APIs in IS 5.3.0.

**Email configuration in axis2.xml file**

```
<transportSender name="mailto"class="org.apache.axis2.transport.
mail.MailTransportSender">    <parameter name="mail.smtp.from"
>sampleemail@gmail.com</parameter>
    <parameter name="mail.smtp.user">sampleemail</parameter>
    <parameter name="mail.smtp.password">password</parameter>
    <parameter name="mail.smtp.host">smtp.gmail.com</parameter>
    <parameter name="mail.smtp.port">587</parameter>
    <parameter name="mail.smtp.starttls.enable">true</parameter>
    <parameter name="mail.smtp.auth">true</parameter>
</transportSender>
```

d. The table below lists out all the configuration changes from IS 5.2.0 to IS 5.3.0. Scroll through the table and change the relevant configurations according to the features you are using. Any step which is not explicitly mentioned as “optional” is mandatory for the migration.

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

e. Proceed to step 11 to run the migration client.

> **Note:** Note that if you followed this approach for migration, the migration client will map all claims to a local claim in the wso2 claims dialect. This is done by matching the attribute IDs. If there is a claim with no matching attribute ID, the migration client will create a new local claim to create the association.
>
> For example:
>
> If the following two claims were mapped in WSO2 IS 5.2.0, the migration client may not identify this because the attribute IDs are different.
>
> ```
> <Dialect dialectURI="http://wso2.org/claims">
>     <ClaimURI>http://wso2.org/claims/streetaddress</ClaimURI>
>     <DisplayName>Address</DisplayName>
>     <AttributeID>streetAddress</AttributeID>
>
> <Dialect dialectURI="http://wso2.org/oidc/claim">
>      <ClaimURI>street_address</ClaimURI>
>      <DisplayName>Street Address</DisplayName>
>      <AttributeID>street</AttributeID>
> ```
>
> As a result, the migration client will create a new local claim like "http://wso2.org/claims/migration__street__73622 " and map the OIDC claim to the new local claim.

11. Start the Identity Server 5.3.0 with the following command to perform the data migration for all components.

> See the notes below to perform migration for individual components or for active tenants only.

a. Linux/Unix:

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

b. Windows:

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

### Migrate individual components

**Optional:** To migrate certain components only, use the relevant commands in the table below.

> **Warning!** Unless specifically required, it is recommended to perform the full data migration by executing the command given above. Component migration is intended for certain special cases only, and may cause errors due to incomplete migration, if done incorrectly.

| Component | Linux/Unix | Windows |
|---|---|---|
| **Identity Database Schema** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigrateIdentityDB` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigrateIdentityDB` |
| **Claim Data** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigrateClaimData` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigrateClaimData` |
| **Email Template Data** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigrateEmailTemplateData` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigrateEmailTemplateData` |
| **Permission Data** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigratePermissionData` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigratePermissionData` |
| **Challenge Question Data** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigrateChallengeQuestionData` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigrateChallengeQuestionData` |
| **Resident IdP MetaData** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigrateResidentIdpMetaData` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigrateResidentIdpMetaData` |
| **OIDC Scope Data** | `sh wso2server.sh -Dmigrate -Dcomponent=identity -DmigrateOIDCScopeData` | `wso2server.bat -Dmigrate -Dcomponent=identity -DmigrateOIDCScopeData` |

## Migrate active tenants only

**Optional:**If you have any disabled/inactive tenants in your previous version of WSO2 IS that you do not want to bring forward to the next version, do a complete migration for all components with active tenants only.

Start the server against the migration client jar located in the `<IS_HOME>/repository/components/dropins` directory using the `-DmigrateActiveTenantsOnly` flag, as shown below.

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

Once the migration is successful, stop the server and start using the appropriate command.

a. Linux/Unix:

```
sh wso2server.sh
```

b. Windows:

```
wso2server.bat
```

**Troubleshooting tip**

If the database script has not run properly, there might be an error when trying to login to the management console. If you are faced with the following error, follow the steps given below to fix it.

**Error log**

```
ERROR {org.wso2.carbon.core.services.authentication.AuthenticationAdmin} -
System error while Authenticating/Authorizing User : Error when handling event :
PRE_AUTHENTICATION
```

1. Stop the WSO2 IS server.
2. Set the following IdentityMgtEventListener with `orderId=95` to **enable=false** in the `<IS_HOME>/repository/conf/identity/identity.xml` file.

```
<EventListener type="org.wso2.carbon.user.core.listener.
UserOperationEventListener" name="org.wso2.carbon.identity.governance.
listener.IdentityMgtEventListener" orderId="95" enable="false" />
```

3. Start the WSO2 IS server and login to the management console.
4. Click **Add** under **Claims** and click **Add Local Claim**.
5. Enter the following values and click **Add**.
    a. **Claim URL:** `http://wso2.org/claims/identity/accountDisabled`
    b. **Display Name:** Account Disabled
    c. **Description:** Account Disabled
    d. **User Store Domain Name:** PRIMARY
    e. **Mapped Attribute:** Ref
6. Stop the WSO2 IS server.
7. Set the following IdentityMgtEventListener with `orderId=95` to **enable=true** in the `<IS_HOME>/repository/conf/identity/identity.xml` file.

```
<EventListener type="org.wso2.carbon.user.core.listener.
UserOperationEventListener" name="org.wso2.carbon.identity.governance.
listener.IdentityMgtEventListener" orderId="95" enable="true" />
```

8. Start the WSO2 IS server. You should now be able to login to the management console.