Exercise 1: Setup Development Environment

Getting Help

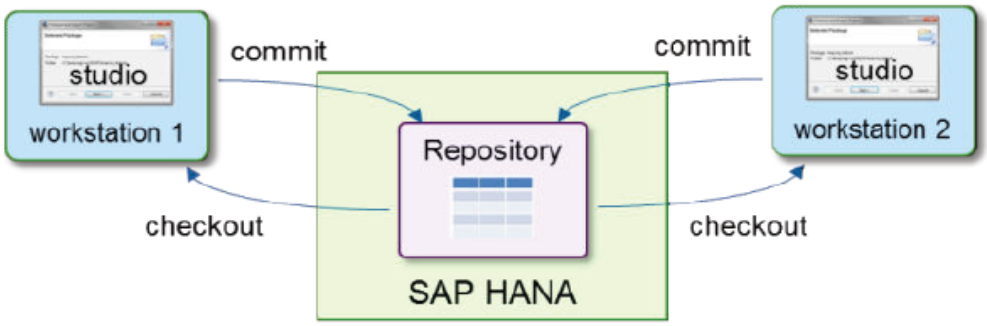
If you need addition help resources beyond this document, we would suggest the following content:

- Learning Portal at http://training.startups.saphana.com/
- HANA Academy at http://www.saphana.com/community/implement/hana-academy
- The internal SAP HANA XS Community https://community.wdf.sap.corp/sbs/community/xs
- The Online Help at http://help.sap.com/hana/hana_dev_en.pdf

About the HANA Development Perspective

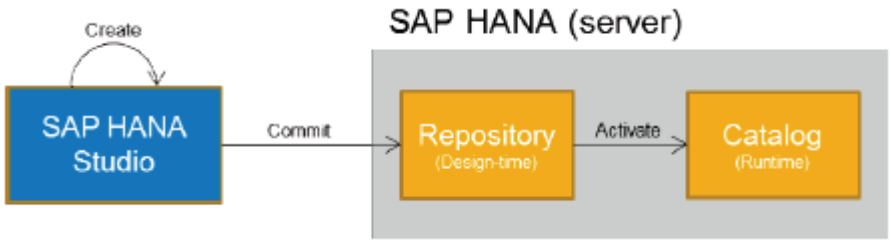
HANA Development Perspective of HANA Studio supports integrated and collaborative development, debugging and deployment of HANA contents and applications. The SAP HANA repository is the design-time storage system for development objects. You can add objects to the repository, update the objects and commit them and activate them to compile design-time objects into runtime objects. The repository supports the following:

- Sharing of objects between multiple developers
- Version control
- Transport
- Type sensitive editor and debugger



To enable multiple developers to work on the same project, here are the steps to go:

- **Creating a Repository workspace:** A local directory like other eclipse workspace, map to the HANA repository.
- **Creating a Project:** Projects group together all the artifacts you need for the development.
- **Sharing a Project:** Enable developers to track and synchronize local changes with the repository.



And have a quick look at the status descriptions below:

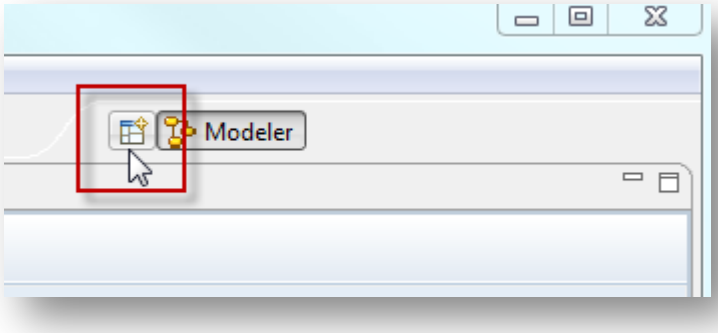
Table 7: Status of Development Objects

Icon	Description
	The file is not committed. You made a change to the file and saved it to your workstation, but have not yet committed the changes to the repository.
	The file is committed.
	The file is activated.

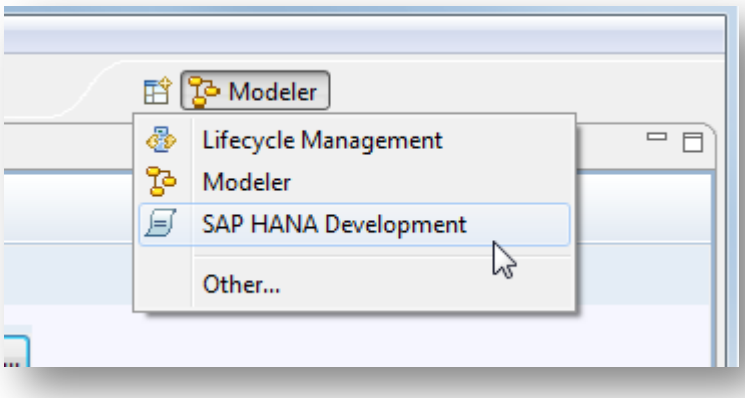
Adding the HANA Studio Perspectives

To support the new developer centric workflow, there are some need Eclipse Perspectives which have been added to SAP HANA Studio. These are not displayed by default.

1. In the upper right corner of your SAP HANA Studio, there is an add Perspectives button. Press this.

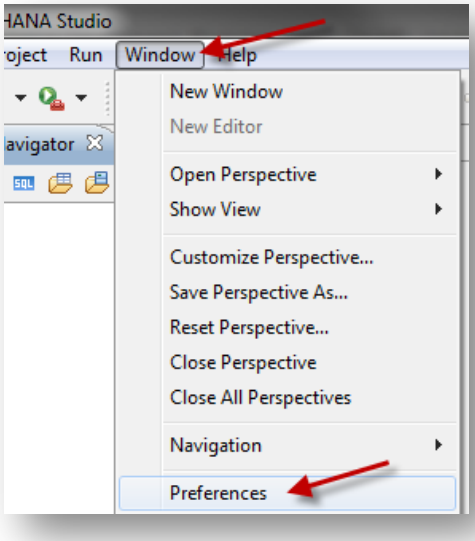


2. Add the SAP HANA Development perspective. This is the perspective you should be using for almost all of this workshop.

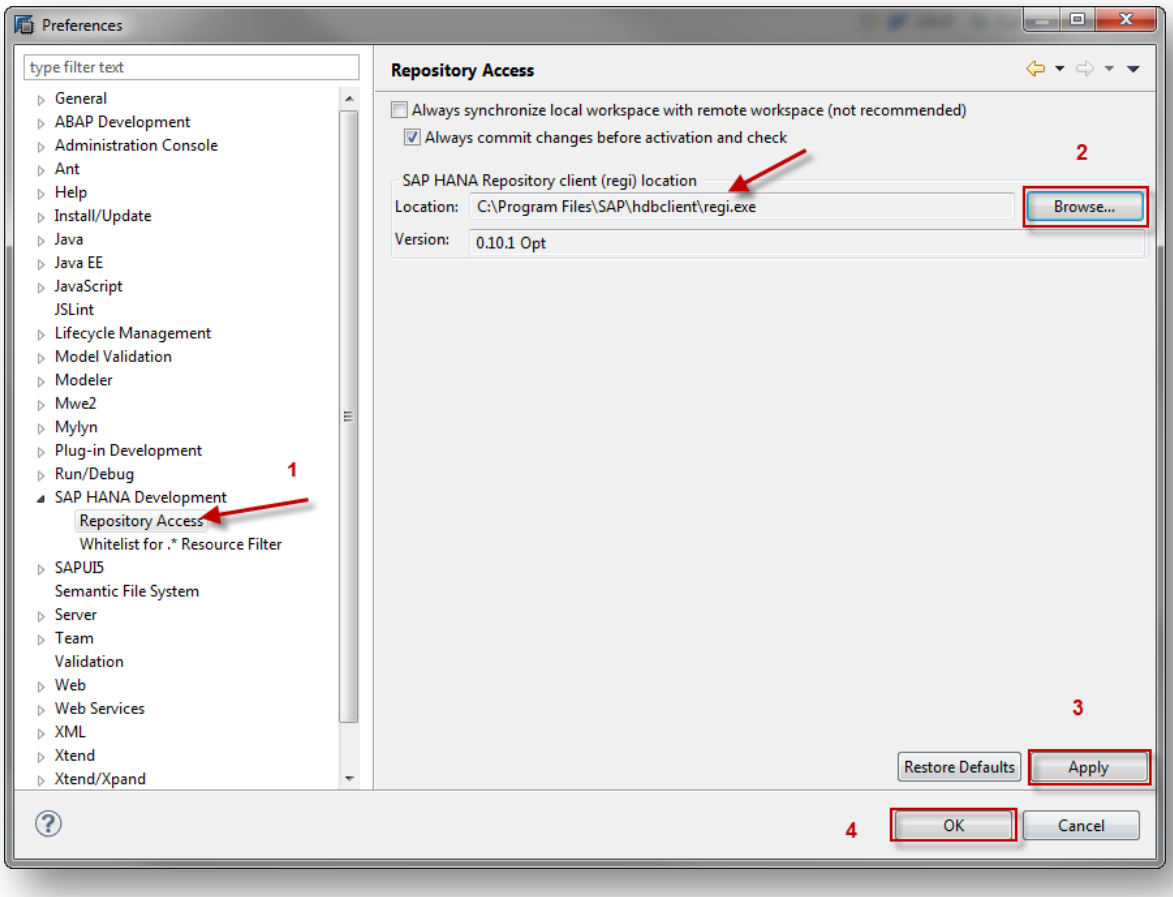


Configure Regi for SAP HANA Studio

1. From the Window menu, choose “Preferences”.

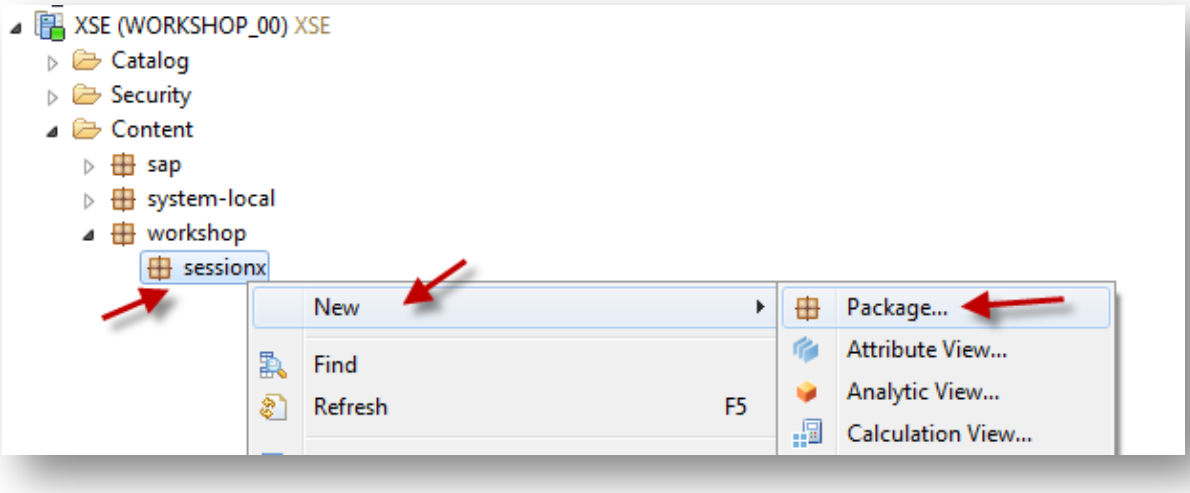


2. From the left side of the screen, expand “SAP HANA Development”, then click “Repository Access”. In the right side of the screen, click “Browse”. Browse to the client installation directory(C:\Program Files\SAP\hdbclient\) and choose “regi.exe”. Next, click “Apply”, then “OK”.



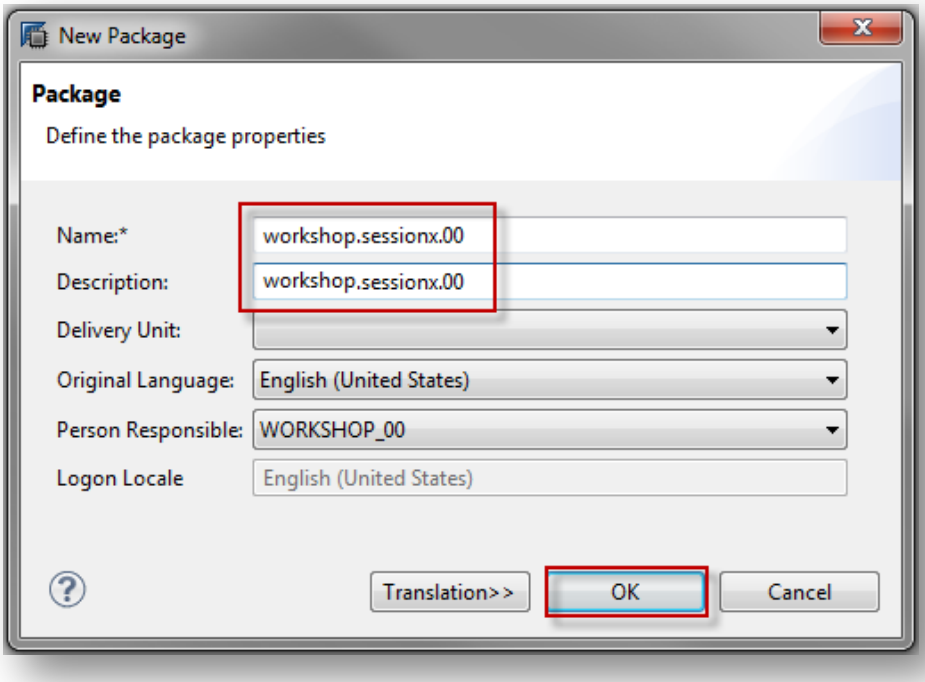
Creating a workspace and project

1. Expand the “workshop” package under the Content folder. There should be session specific packages already created for you. If you are using your own HANA instance, you will be asked to create the “workshop” package by yourselves. Because this hands-on session will be given multiple times, we will use different session letters to separate each classes’ work. The session instructor will tell you what session package to use for your session. Right-click on the session package and choose “New”, then “Package”.

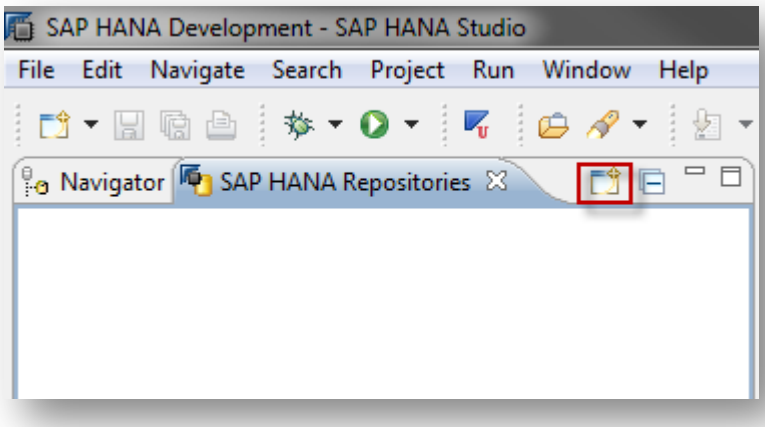


2. In the package dialog, you will create a package for your developer group. This ensures that every developer has a unique namespace for his or her work. All the content you create today will then be placed within this unique session/group package hierarchy. Enter the name of the package as your group number.

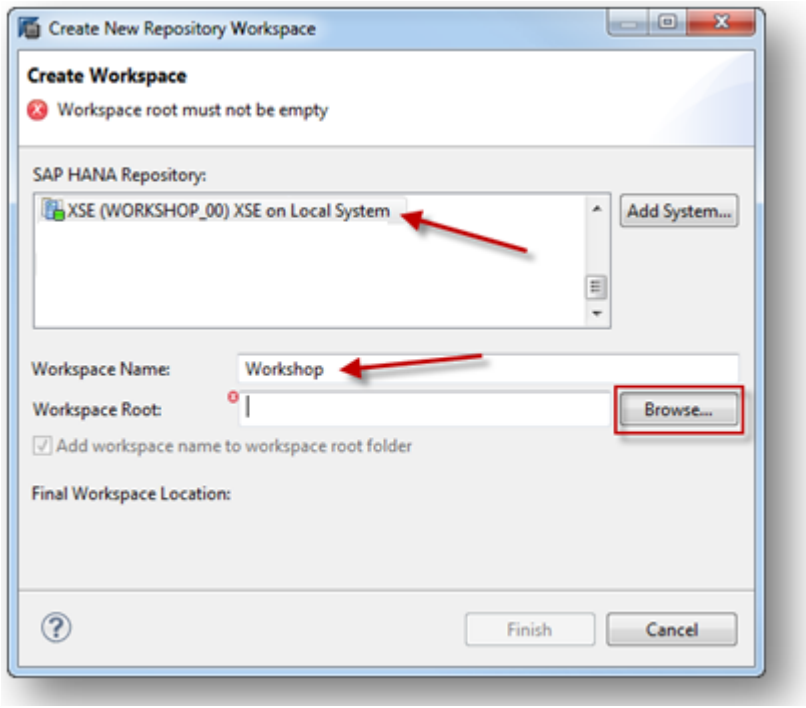
   For example, if the assigned group number is “01” and your session package is sessionx, then the complete package name would be workshop.sessionx.**01**. That is workshop**<dot>**sessionx**<dot>**01. You will use this same naming convention throughout the rest of the exercise document. Click “OK”.



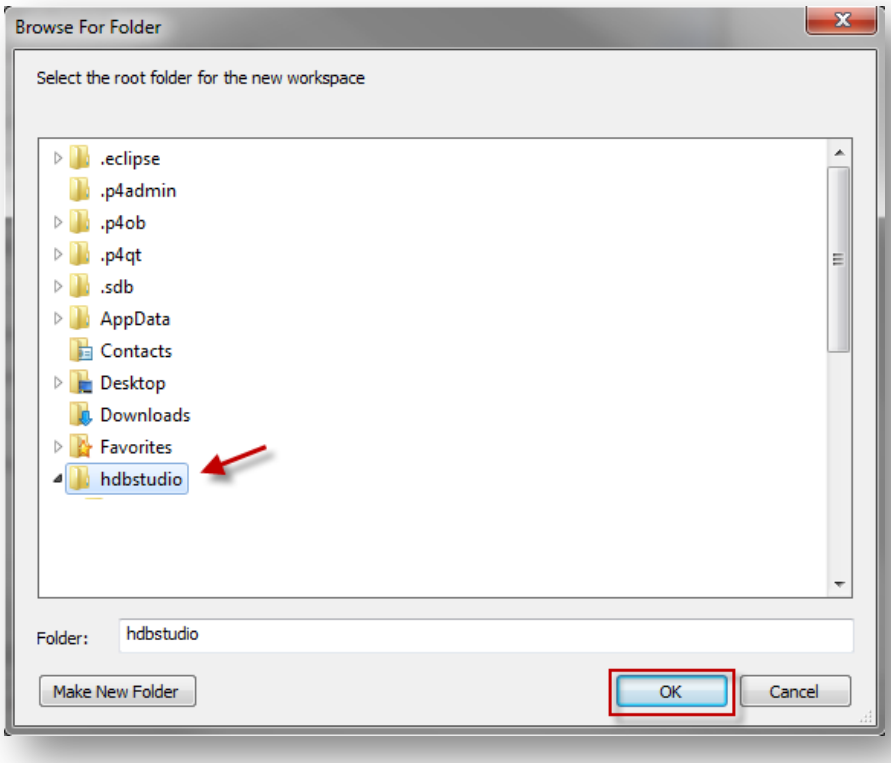
3. Switch to the “SAP HANA Repositories” tab, and click the “Create Repository Workspace” button or right click and select the “Create Repository Workspace” in the context menu.



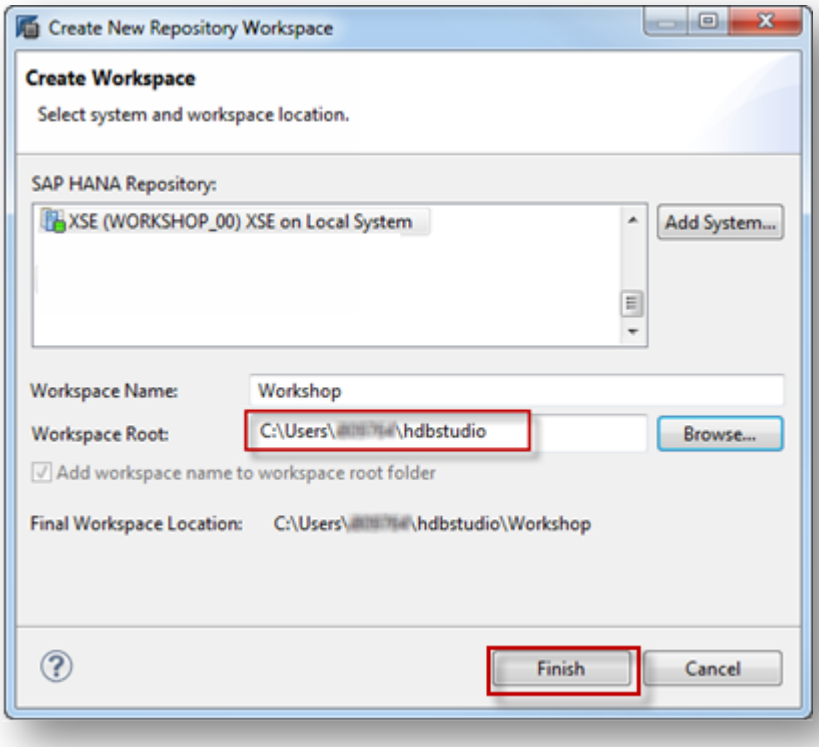
4. Select the system you created in the previous steps. Then enter the name of the workspace as “Workshop”, and then click “Browse”.



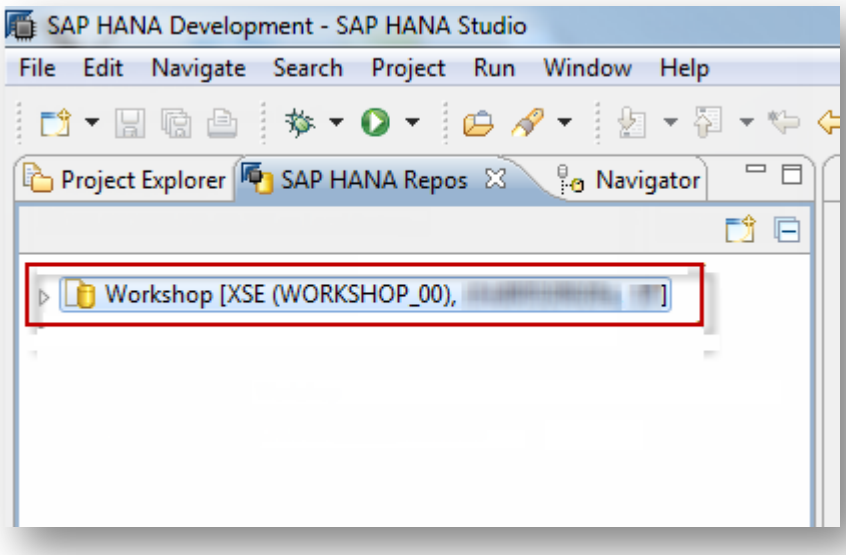
5. Choose any directory on your local drive, for example, you could use the hdbstudio folder under your documents folder. Click “Ok”.



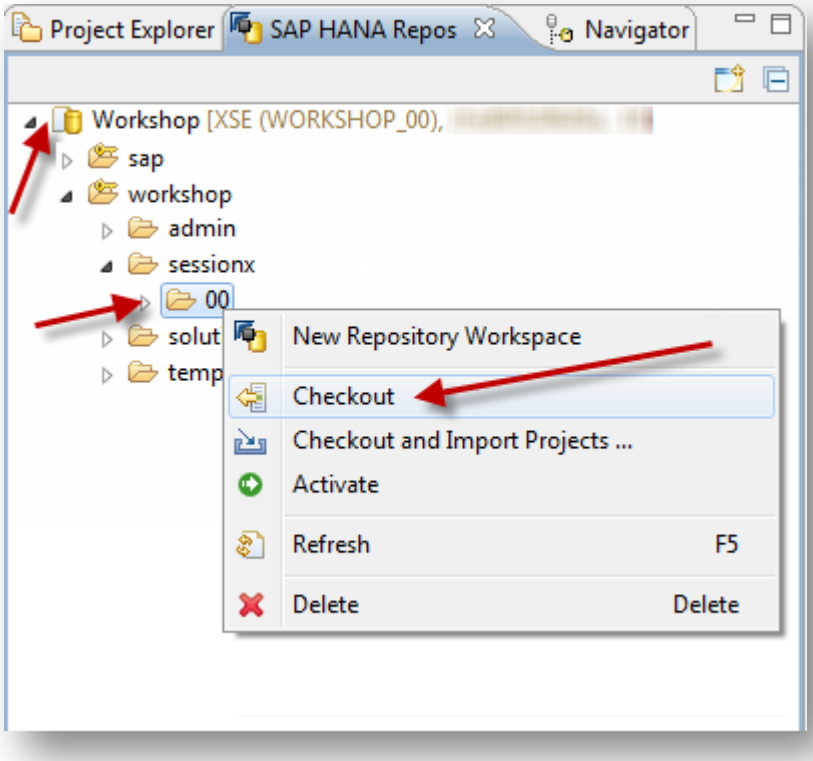
6. Click “Finish”.



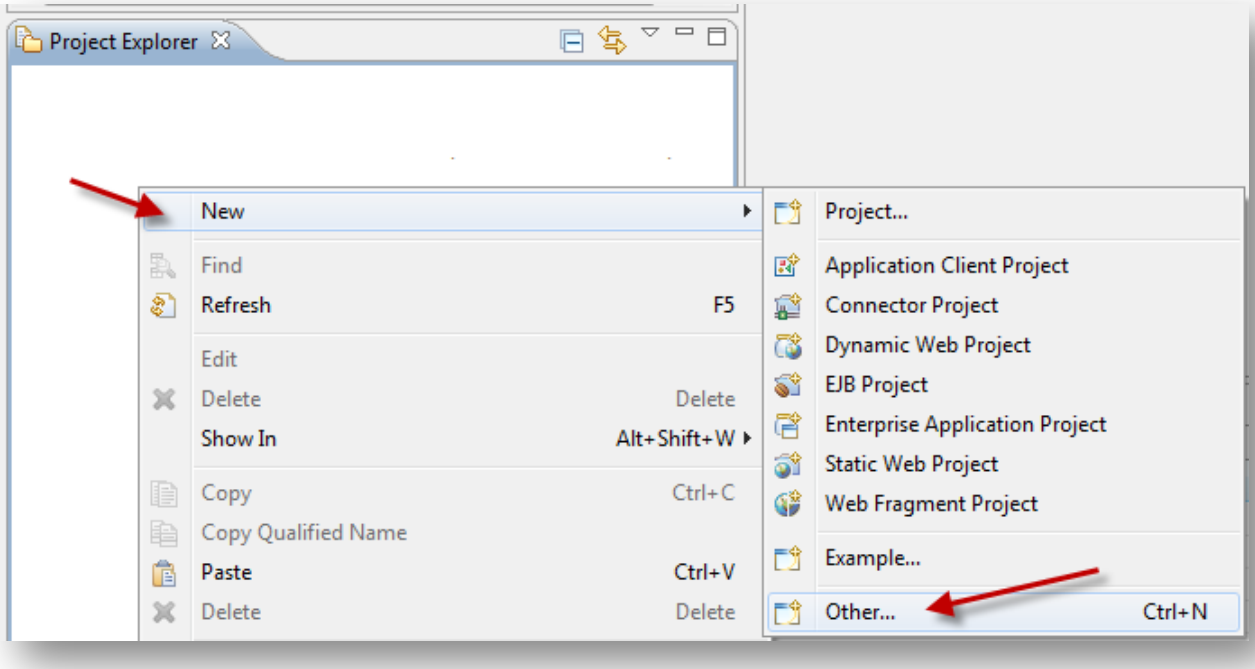
7. You should now see your new workspace in the “SAP HANA Repositories” tab.



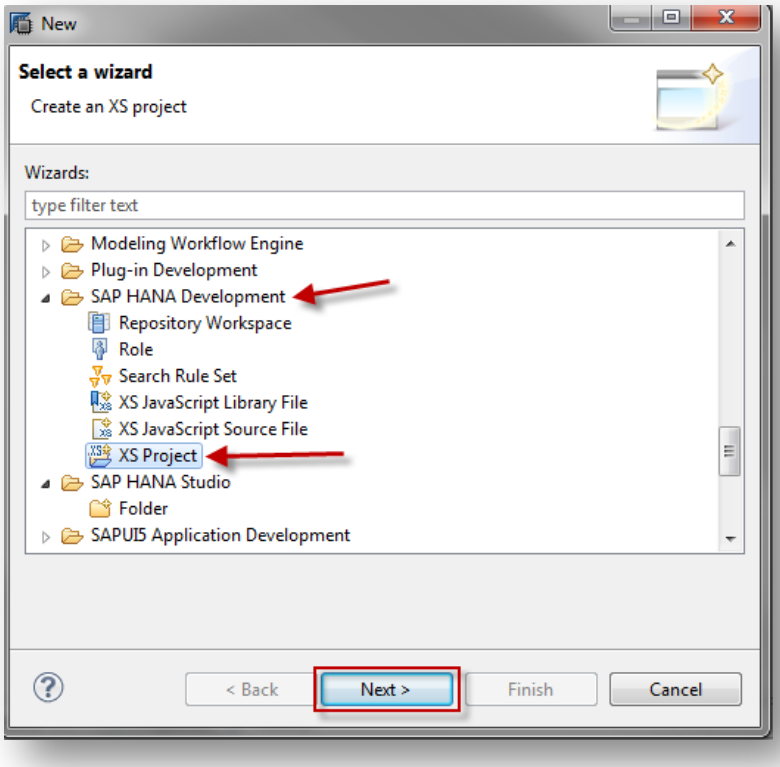
8. Open the workspace by clicking on the arrow to the left. Open your session folder and right click on your group package. Choose “Checkout”. A “Checking-Out” dialog will appear showing the progress, and simply go away when complete. Continue to the next step.



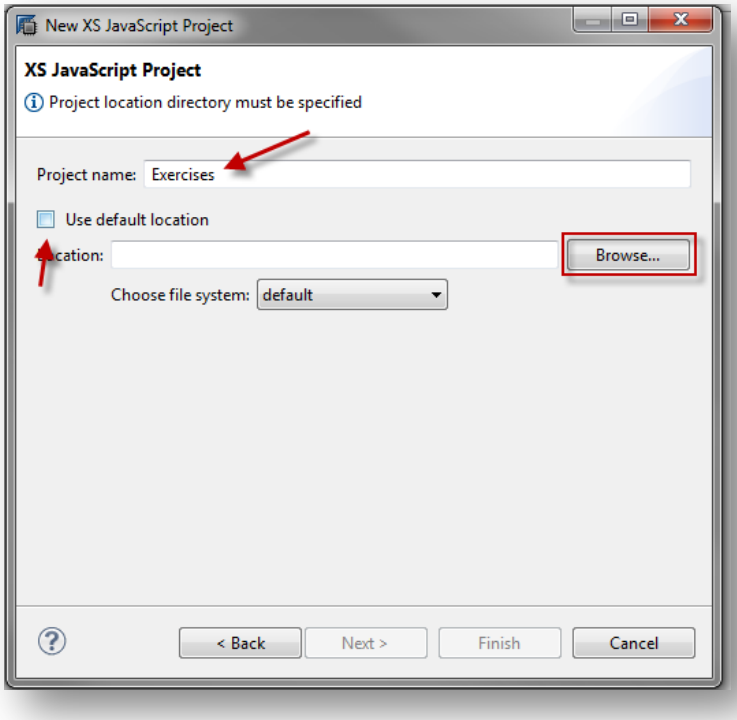
9. Move to the “Project Explorer” view, and right-click in the white space. Choose “New”, then “Other”.



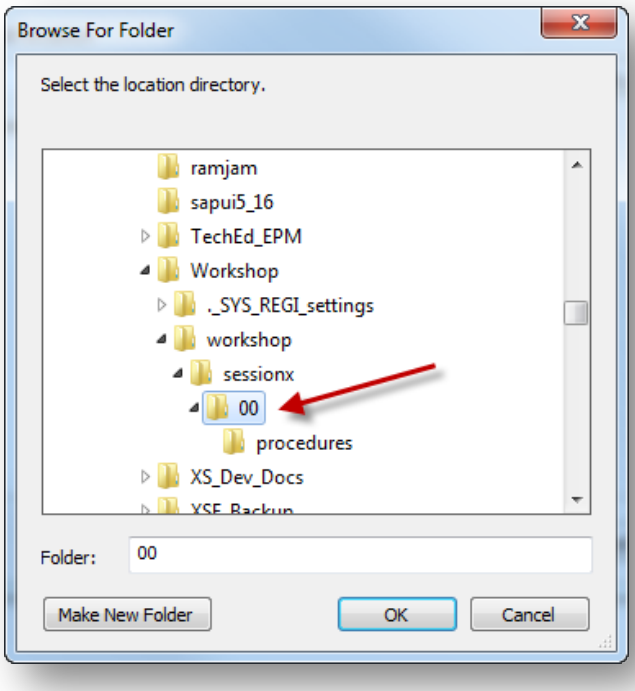
10. Browse to “SAP HANA Development” and choose “XS Project”. Then click “Next”.



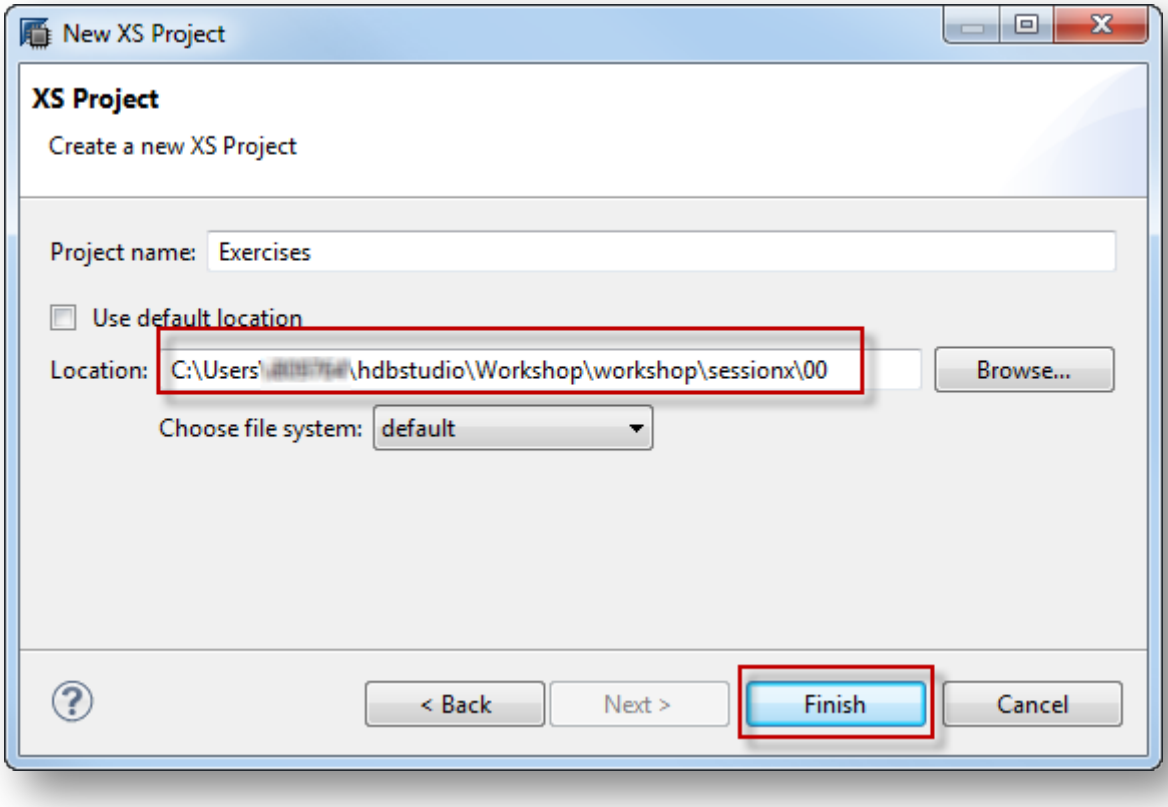
11. Enter the name of the project as “Exercises”, uncheck the checkbox for “Use Default Location”, and then click “Browse”.



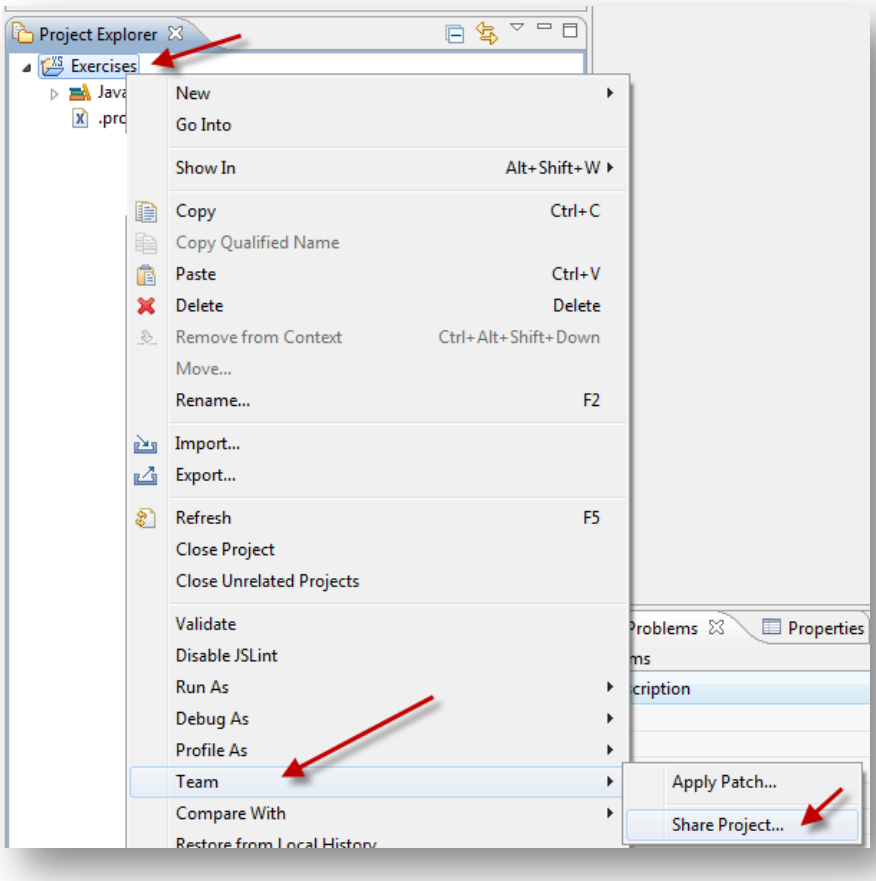
12. Browse to your group package folder on your local machine which was created when you checked it out from the repository earlier. Click “OK”.



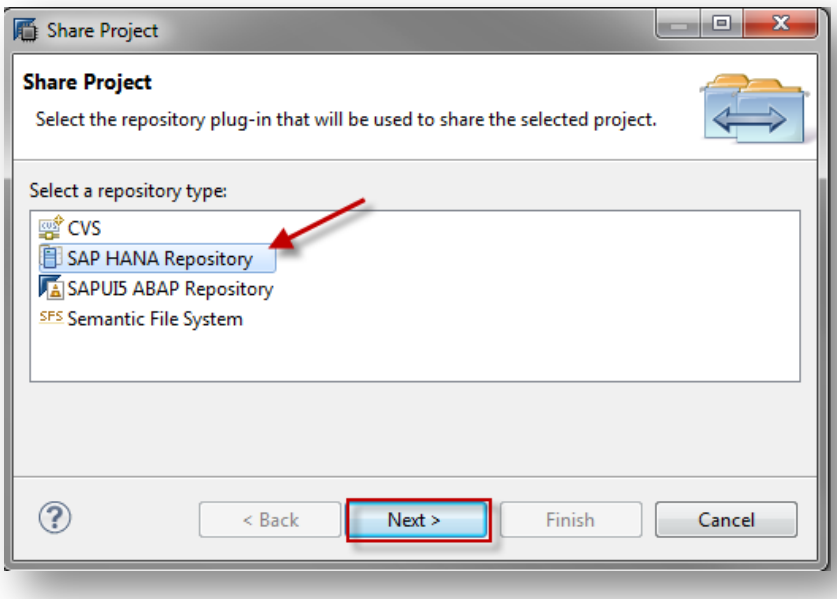
13. Click “Finish”.



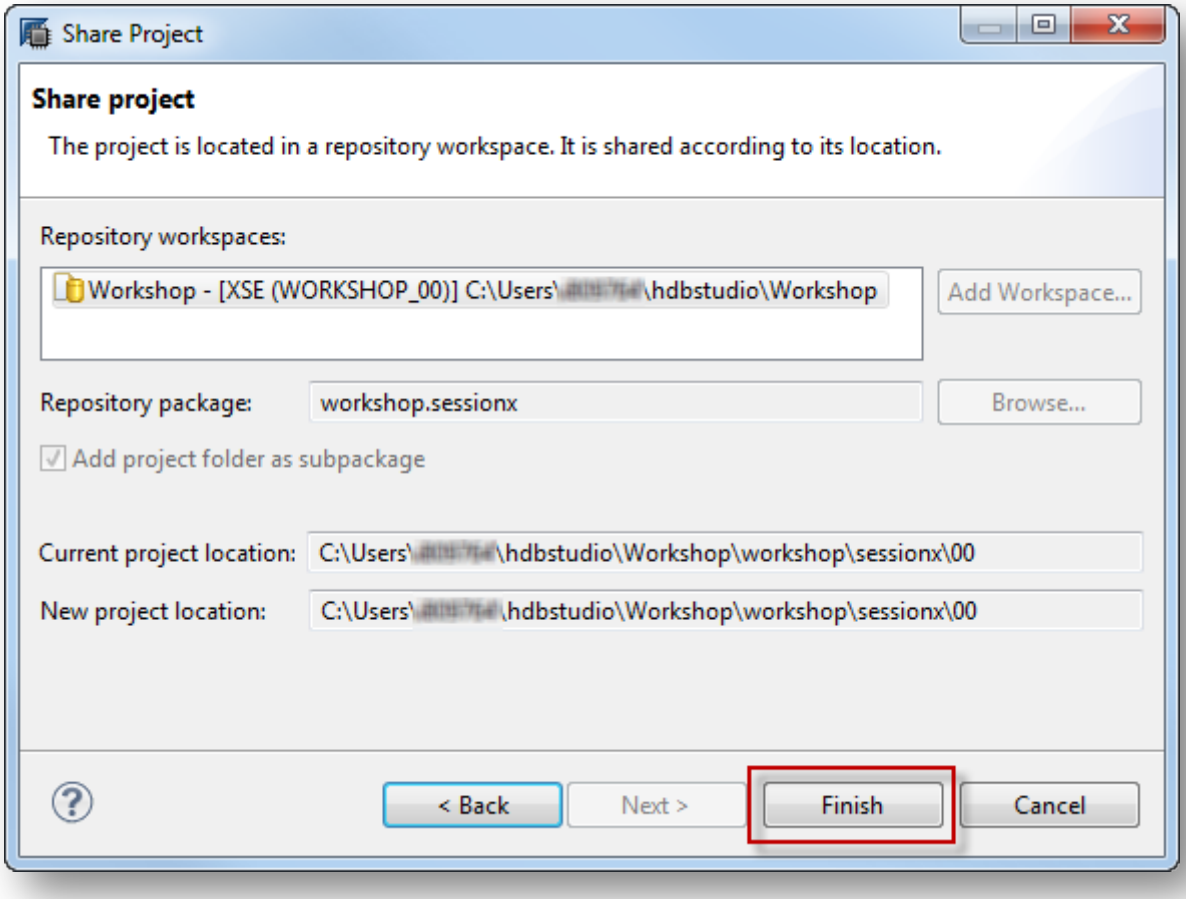
14. Your project should now be visible in the “Project Explorer”. Right click on the project and choose “Team”, then “Share Project”.



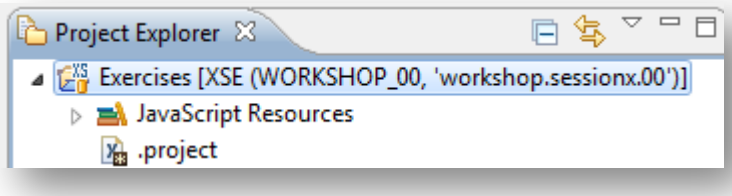
15. Based on your installed plug-ins, you may not get the following dialog. If you do receive the following dialog, choose “SAP HANA Repository” and click “Next”.



16. Keep the defaults, and click “Finish”.



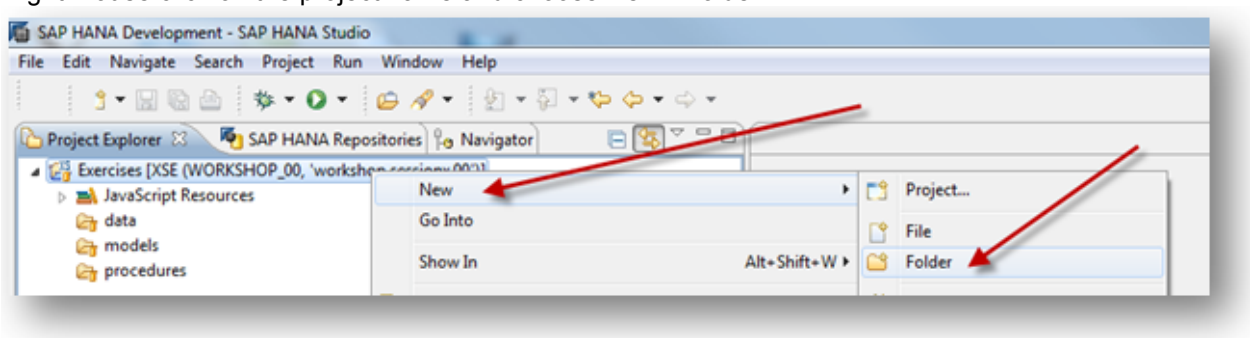
17. Your project is now shared with the repository and you are ready to begin developing your artifacts.



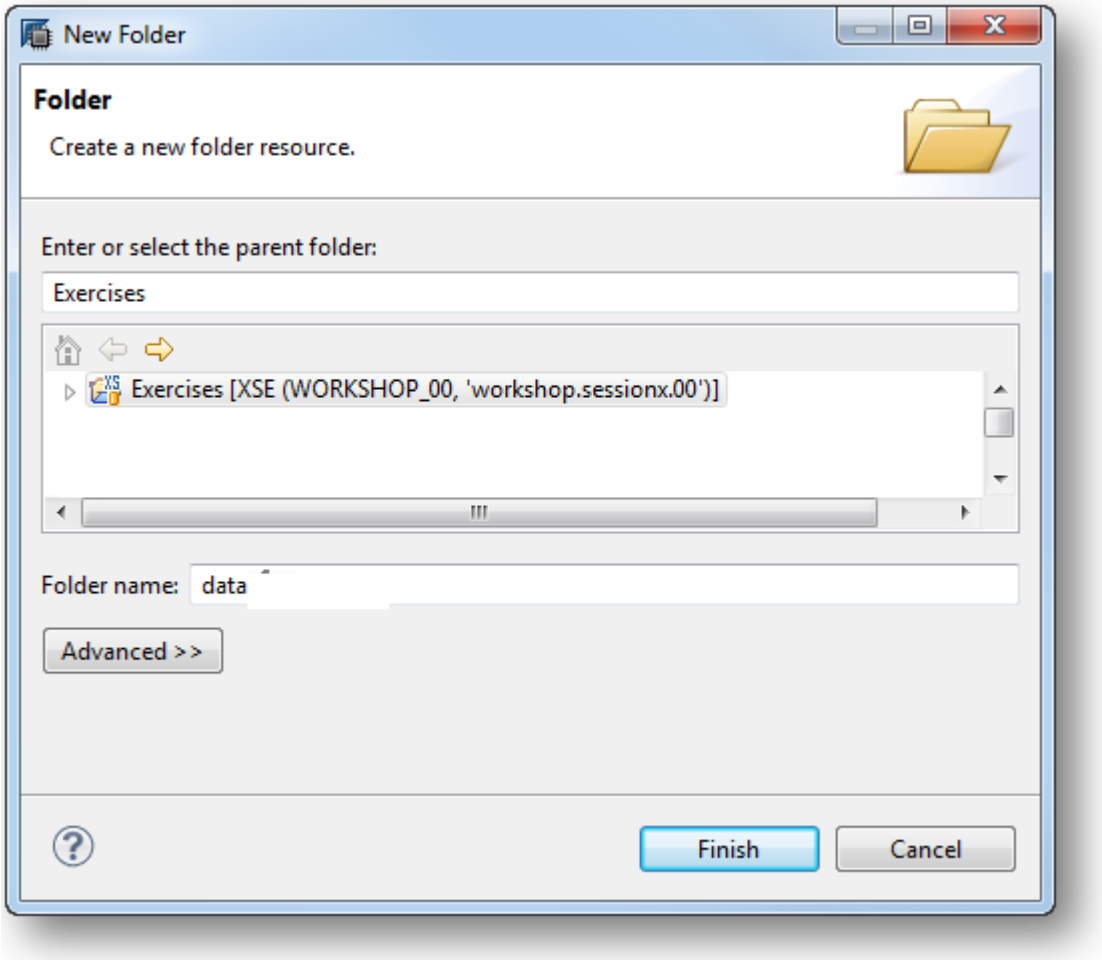
Creating Sub-Packages

In order to prepare for the development we will be doing, we should create sub-packages under your project package. You can create packages from the project explorer simply by creating folders.

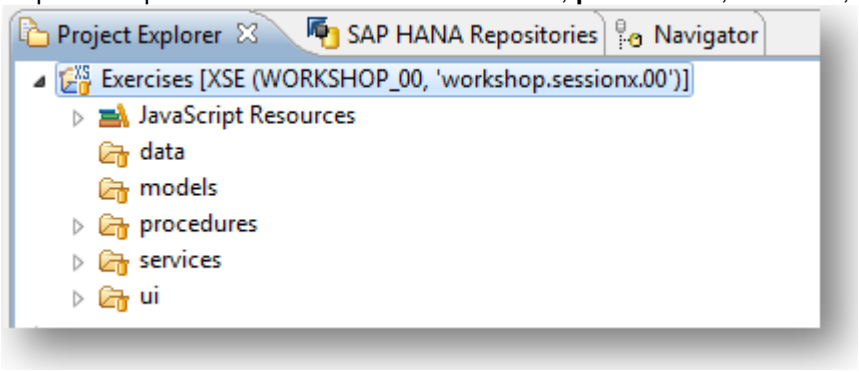
1. Right mouse click on the project name and choose New->Folder.



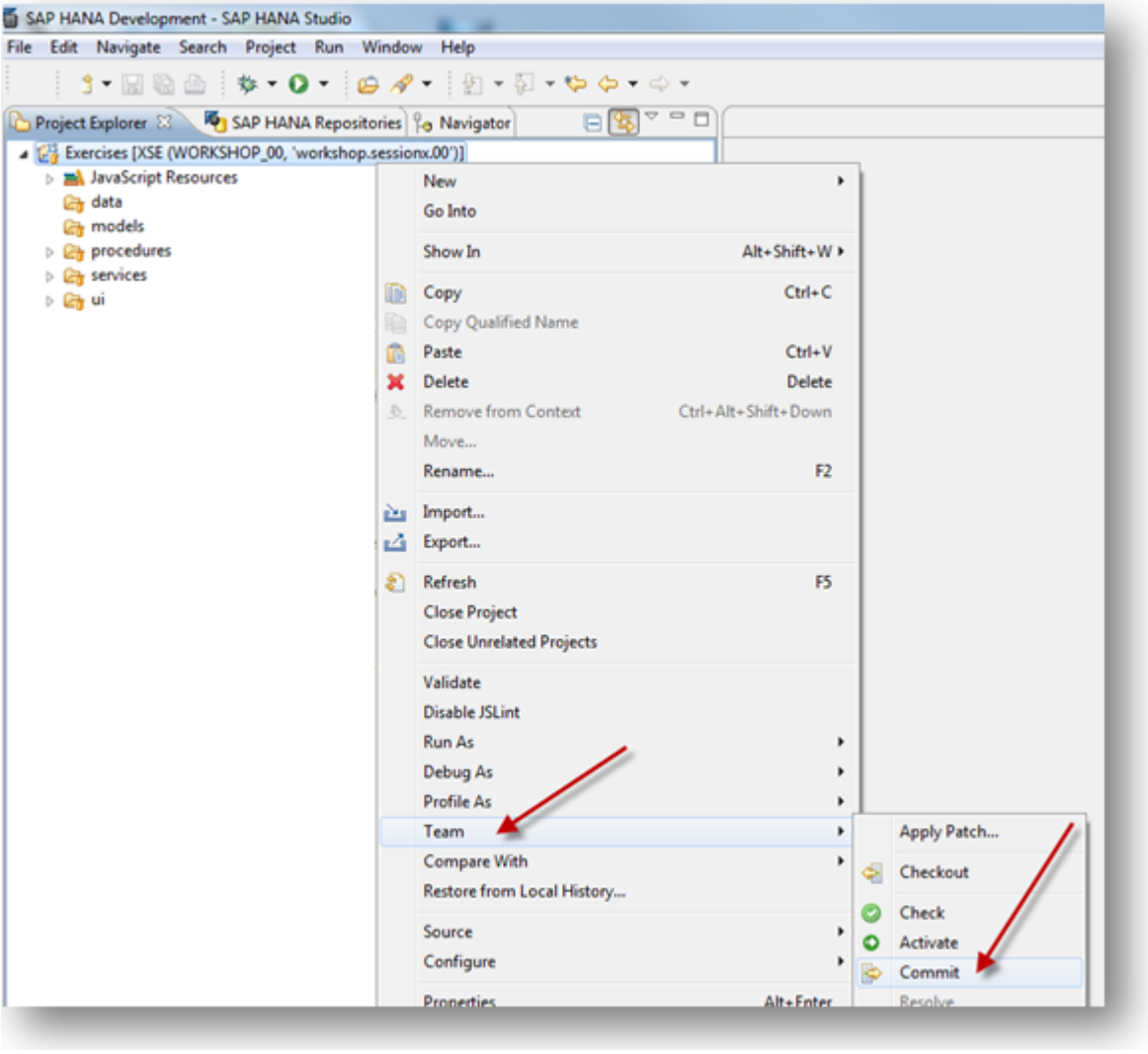
2. Create a folder named **data**.



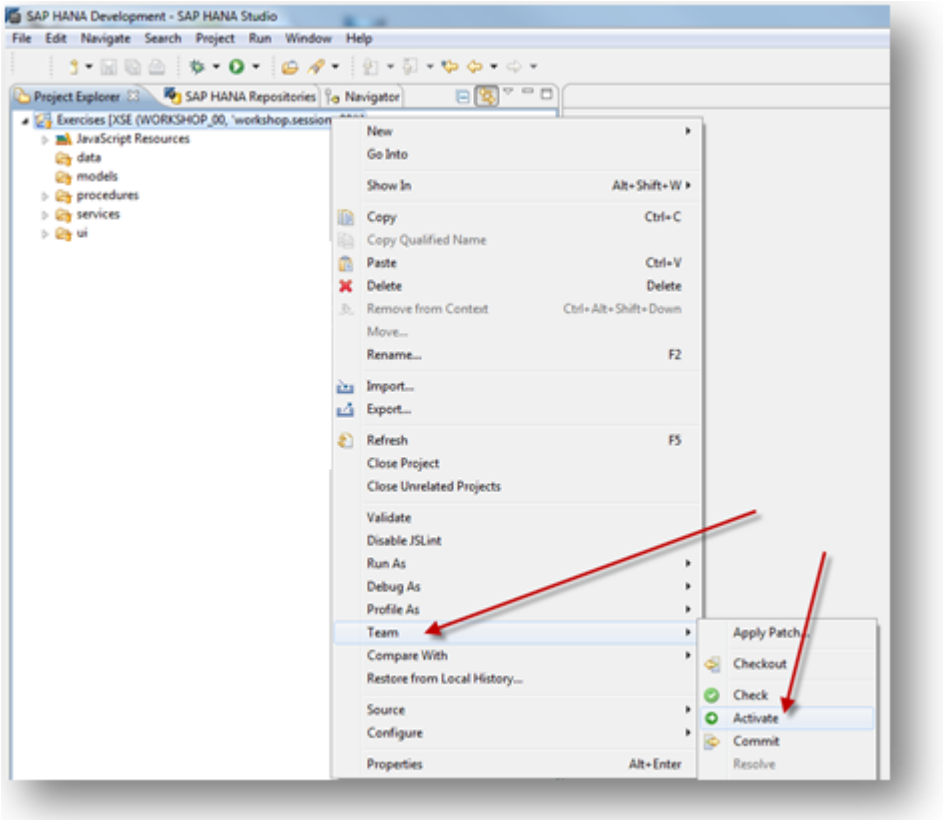
3. Repeat this process for folders named: **models**, **procedures**, **services**, and **ui**.



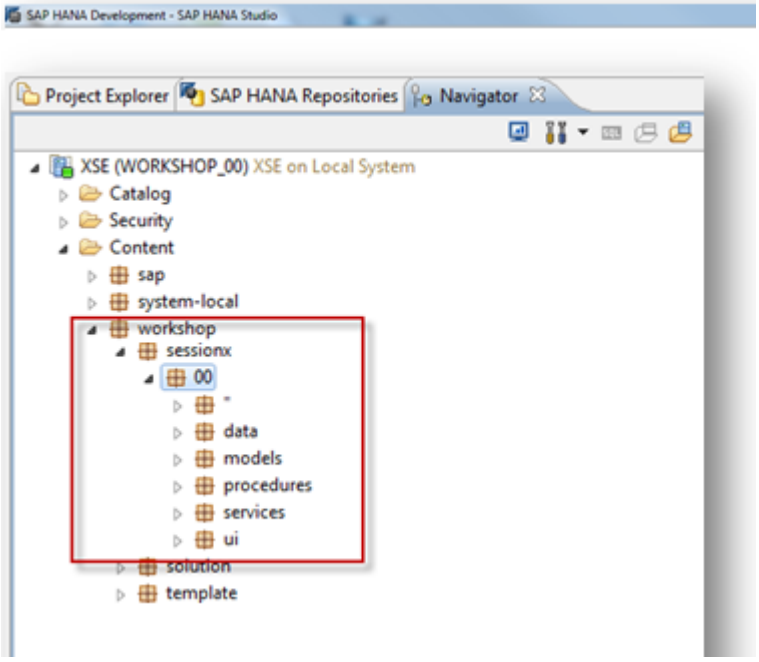
4. You now need to send your project and new folders back to the HANA Repository. This will cause the folders to generate packages. Right mouse click on the project root and choose Team->Commit.



5. Then choose Team->Activate.



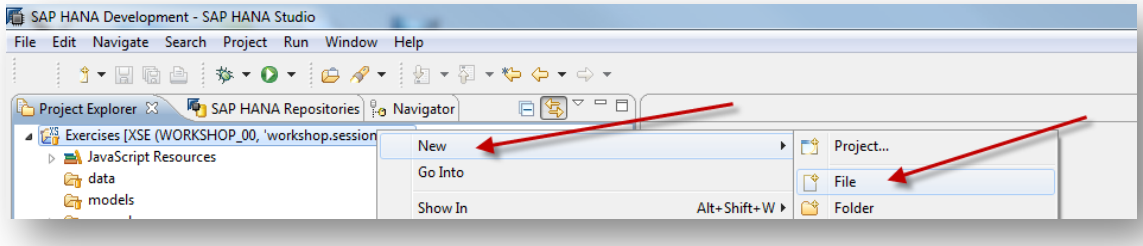
6. If you switch back to the Navigator view, you will see the full package hierarchy which you have created.



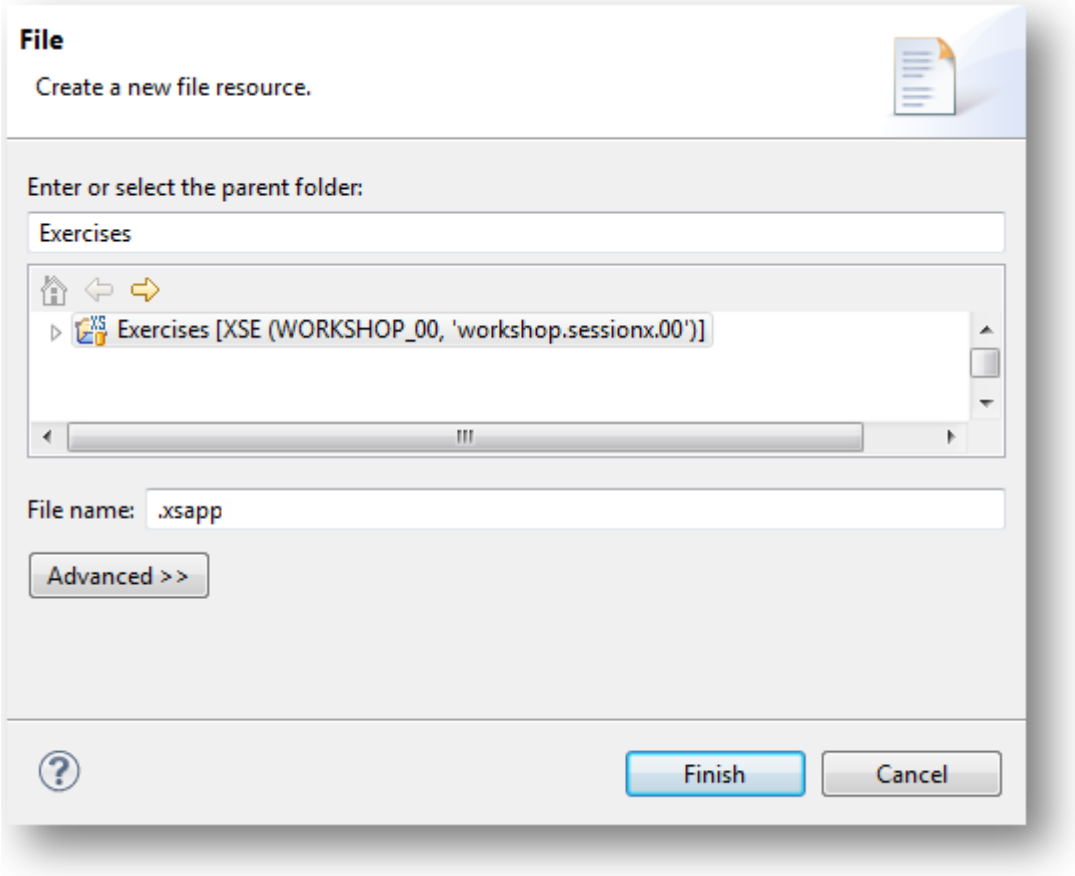
Create an XSAPP and XSACCESS

1. The first artifact we must create is the **XSAPP**. The XSAPP file is necessary for exposing any content via the XSEngine web server.

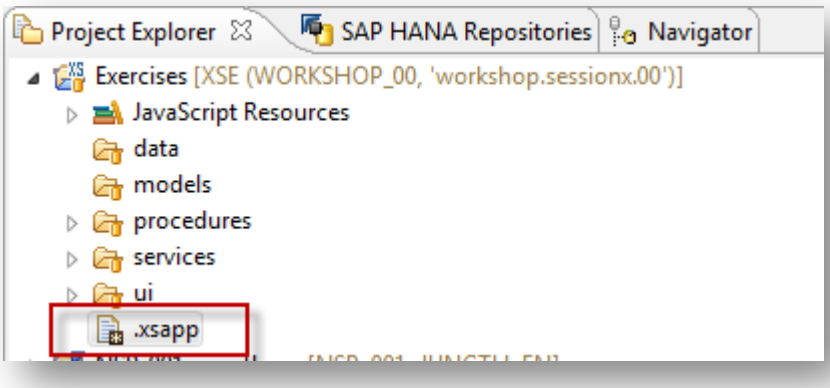
2. Begin by right mouse clicking on the root of your project and choosing **New->File**



3. Create a file named .**xsapp**. This is file with only the extension. This way you can only have one file per folder. This same naming concept is repeated with other artifacts which must only exist once per folder. The names of the individual development artifacts don't need to be unique because the package hierarchy also functions as the namespace and web server path.

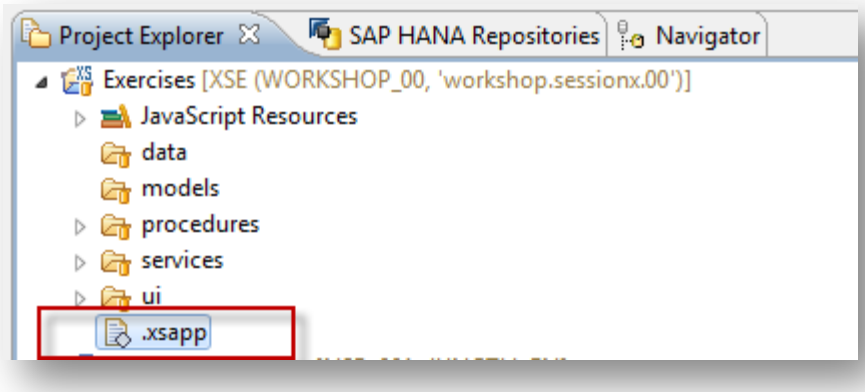


4. The newly created file is visible in your Project Explorer. The icon (white star on a black background) for the file signifies that this object exists on the client side but has not been committed to the HANA Repository yet.



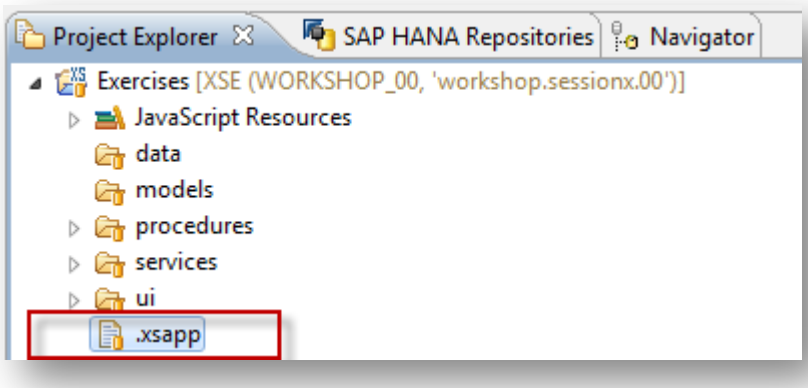
5. The file is automatically opened in an editor on the right side of the screen. This particular development artifact doesn't actually need any content in it. You can close the editor.

6. Commit the file to the repository using **Team->Commit** from the root of the package or on the file itself. This icon (grey diamond) means the file is in the repository, but there is an inactive version.



7. All objects are stored in an inactive state when first committed. Therefore before this artifact can be used on the server it must be activated. Choose **Team->Activate** from the parent folder. This will activate all inactive objects. Individual object activation is also possible.

8. Once successfully activated, the icon next to the object will change.

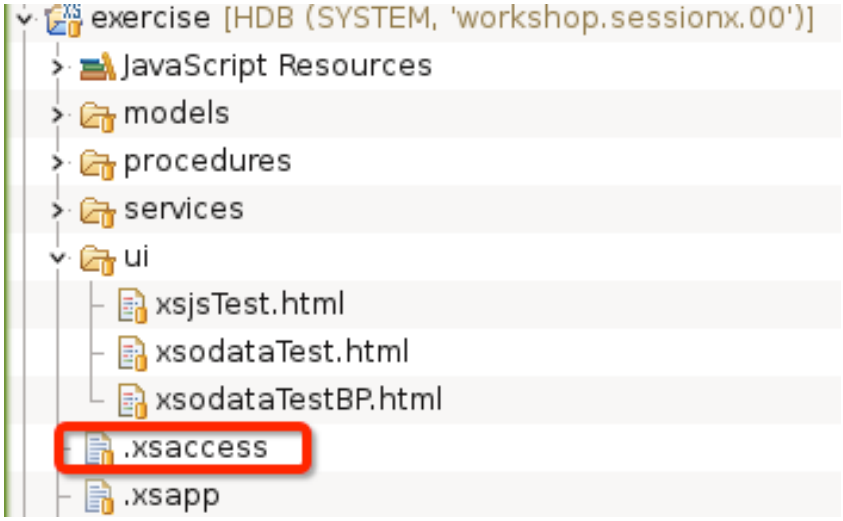


9. Repeat the same steps to create the other file named **.xsaccess**. This is file with only the extension. Copy the source code as below to the file, save and activate it.

Source Code:

```
{
    "exposed" : true
}
```

10. After that, you will be able to see the file.



The Best-Run Businesses Run SAP™