

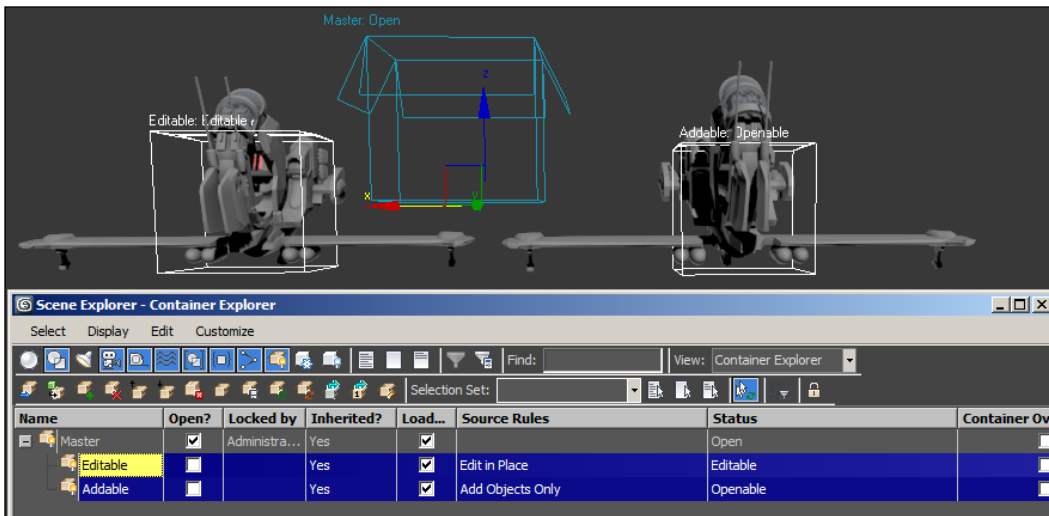
# Containers and XRefs

## Containers

In *Chapter 1, First Launch: Getting to Know 3ds Max*, we mentioned the **Container** as being the younger, richer sibling of the **Group**. It links content to external files for collaborative access or simply for scene organization. Some users contend that XRef Scene and Container functionality competes within 3ds Max and there's debate over which works best. Both allow content updates by various users to feed into other scenes where the assets are also being used. Containers offer rules to lock or protect content inside them while other people are working on related parts of the content. This is done wholesale or by assignment. It's a great feature, but the downside is that containers can be disrupted when working on a network where users swap machines regularly. What can happen is that files which have a local path, can have their Source Definition disrupted. Different users require different work flows, so there are disagreements within the user community (see the Autodesk forum) about the best utility of Containers and XRefs, especially with very large, complex cases. Personally speaking, the best use of Containers seems to be when you are working on your own content, have divided it up, and just need to selectively **Load** parts of a large scene and **Unload** the parts you don't want to show. Collaboration using this tool requires appropriate team communication.

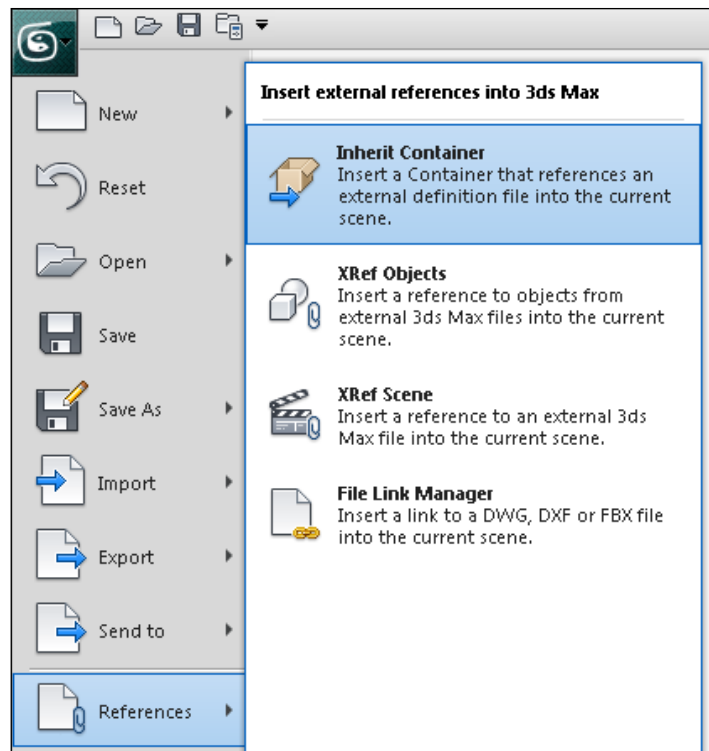
For XRef Scene assets, you can't edit assets in the scene they're cast to. Instead, you have to edit them in the scene they are inherited from. Also, an important benefit of Containers over the older XRef Scenes for collaborators is you can track the current usage of the Container nodes in a scene, using **Tools | Open Container Explorer**. This includes a **Sync Selection** toggle so external changes to the content can be live or deferred.

In the following screenshot, the plane on the left-hand side is sourced from a saved container, **Editable**, and a user in this scene can only **Edit in Place** its contents, so they can edit the polygons, materials, and so on. Meanwhile, objects in **Addable** cannot be edited except that users can **Add Objects Only**. This container can still be added to, but that's all. Presently **Addable** is not open, but is openable. Both **Containers** have been subsumed into another **Container** called **Master**. If we use **Unload Master** option, all the sub-containers are also removed. If we toggle **Open** off, the content remains visible but is not accessible until we toggle it **Open** again.



It is important to close Containers before saving them and adding them into another scene (or a temp file is saved that locks them). This can be subverted by opening Windows Explorer and deleting the **LOCK** file associated with the container, which is a **.maxc** file (they will appear in the same folder).

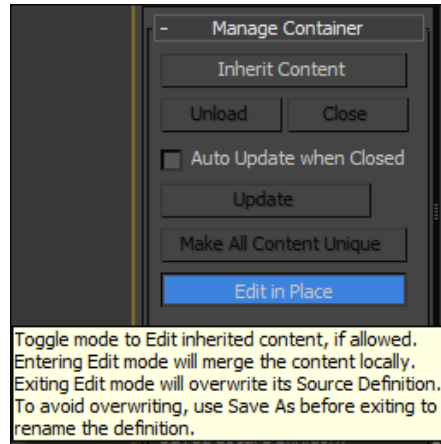
To access a shared container, you can create a new container helper and inherit content to it or go to **Tools | Open Container Explorer (Alt + Ctrl + O)** and click on **Inherit** there. Setting the unassigned hotkey **I** to **Inherit** will save a few clicks: **Customize | Customize User Interface... | Customize | Category: Containers | Inherit Container**. You can also use the method shown in the following screenshot: **File | References | Inherit Container**:



**Inherit** lets you browse folders to locate an existing container and add an instance of it to your scene. You'll have to save the instance to preserve local changes.



If a container fails to display its contents as expected and can't be selected in the view, highlight its entry in the **Container Explorer** (*Alt + Ctrl + O*) and right-click and choose **Containers | Update**.

**Edit in Place** is a mode that you need to toggle and has a button **Edit in Place** within the Command Panel in the **Manage Container** panel of the selected **Container**. Turning it on lets you access and edit the objects collected in the container. If you save the changes, the original container will be updated accordingly. If you choose **Save As...**, you can preserve the original state of the model and keep the changes separate.

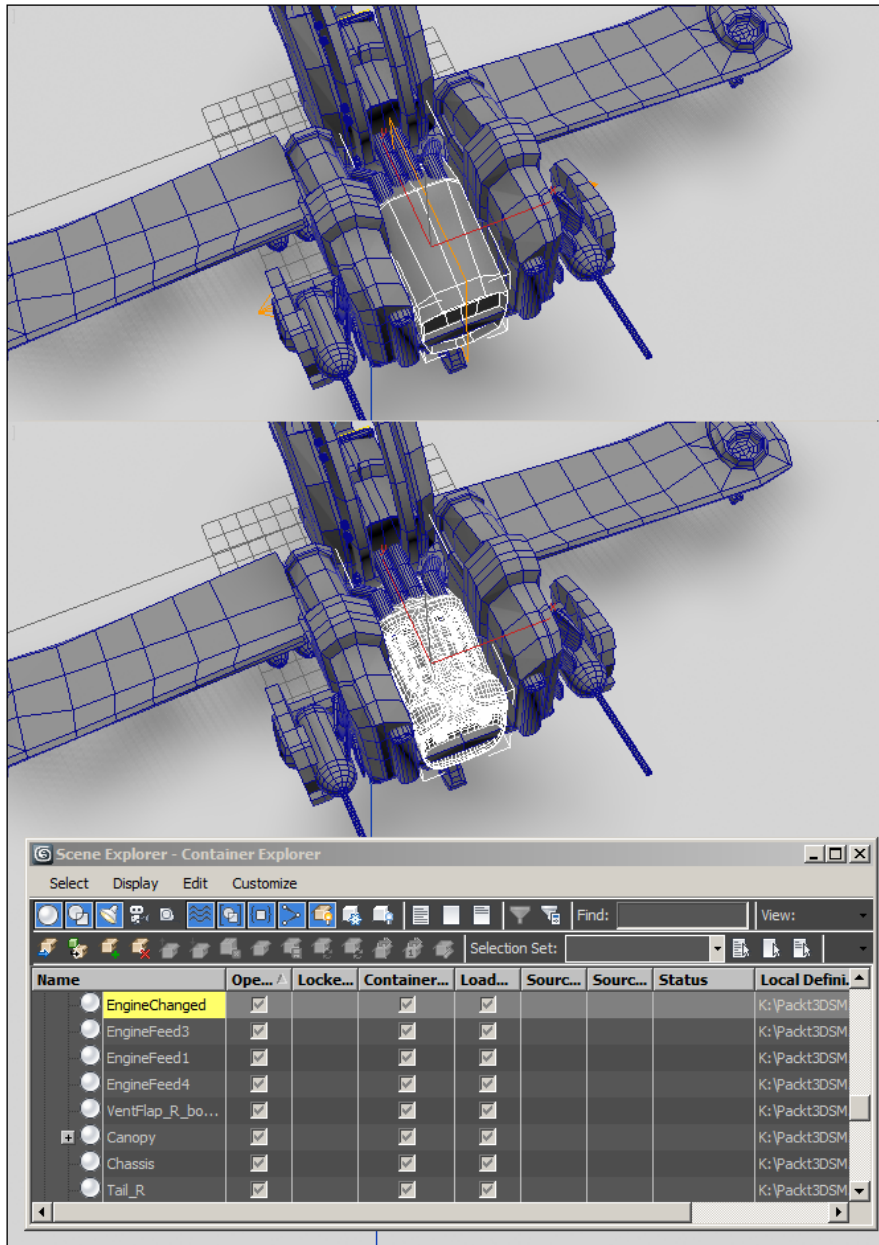



## Editing a Container

In this lesson, we'll create a simple example of a container to show how you can populate a scene, using the following steps, and update their content across scenes without a fuss:

1. Open the scene `\Packt3dsMax\Chapter 2\BikeEditEngine.max`. This scene already contains a backdrop made of local objects and a Container, `Container001`, which collects together all the components of the bike.
2. Press `Ctrl + Alt + O` to open the **Container Explorer**, and then highlight the entry `Container001`.
3. Click on the checkbox **Open?** for `Container001`. Nothing happens. For this object, the Source rule is **Edit in Place**. Instead, within the **Command Panel**, in the **Modify** tab , under **Manage Container**, you should click on the **Edit in Place** button, as shown in the previous screenshot. Now, you'll see the container shows that it is open and its **Status** is **Editing in Place**. A  icon now shows next to the `Container001` entry, which you can expand to view its contents. Scroll down the list to find **EngineBay**, or search for it using the **Select by Name** tool (*H*).

4. In the main toolbar, there is a field **Create Selection Set**; type EngineBay so you can grab it again quickly, later. Don't forget to hit *Enter* after typing.
5. Go to the **Tools** menu, choose **Rename Objects...**, and type in the **Base Name** field EngineChanged.




6. We could simply edit the current object, but as you may not yet be familiar with modeling tools, we will simply replace the model, as shown in the previous screenshot. Go to the **File** menu and choose **Import | Replace**. Browse to `\Packt3dsMax\Chapter 2\` and choose `EngineChanged.max` then click on **Open**. In the **Replace** dialog, highlight the entry **EngineChanged** then click on **OK**. You'll be prompted **Do you want to Replace Materials along with Objects?**. In this case, it doesn't matter which you choose. Click on **Yes**.
7. You should now see a much more detailed model in place of the previous one. Save the Container to keep this change. In the **Container Explorer** (`Ctrl + Alt + O`), highlight the entry `Container001`, and in **Command Panel | Modify** , turn off **Edit in Place**. Then click on **Save**.

## Creating a Container

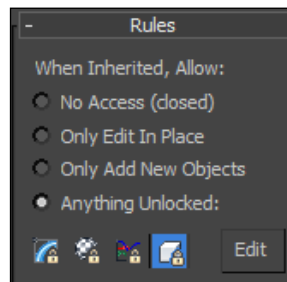
Let's assume you already have a scene with some objects you want to collect together into a container.

To create a container of your own, go to **Command Panel | Create**  and choose the category **Helpers** .

Highlight **Container** and click and drag it in the viewport.

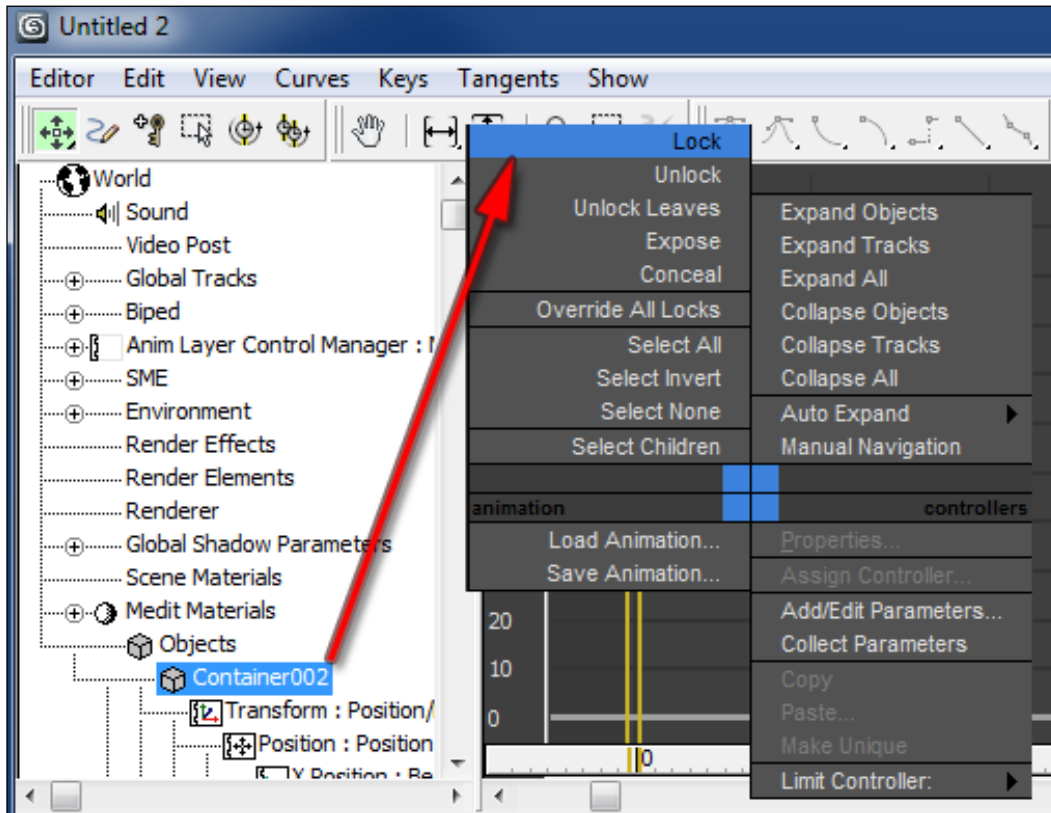
In **Command Panel | Modify** , for the added helper, click on the **Add** button and select, from the list of scene objects, all you want to include.

Further down, in the section **Rules for When Inherited, Allow:**, the container defaults to **No Access (closed)**. Change this as required. The **Anything Unlocked** option is the most open, and you can lock by category using the icons below the rules.



If you press **Edit**, you will expose a floating **Track View**, next to locking icons, and can browse the scene objects. If you then right-click on any object displayed in the contextual **Quad** menu, you'll see **Lock** and **Unlock** commands, which let you set granular access to container contents.

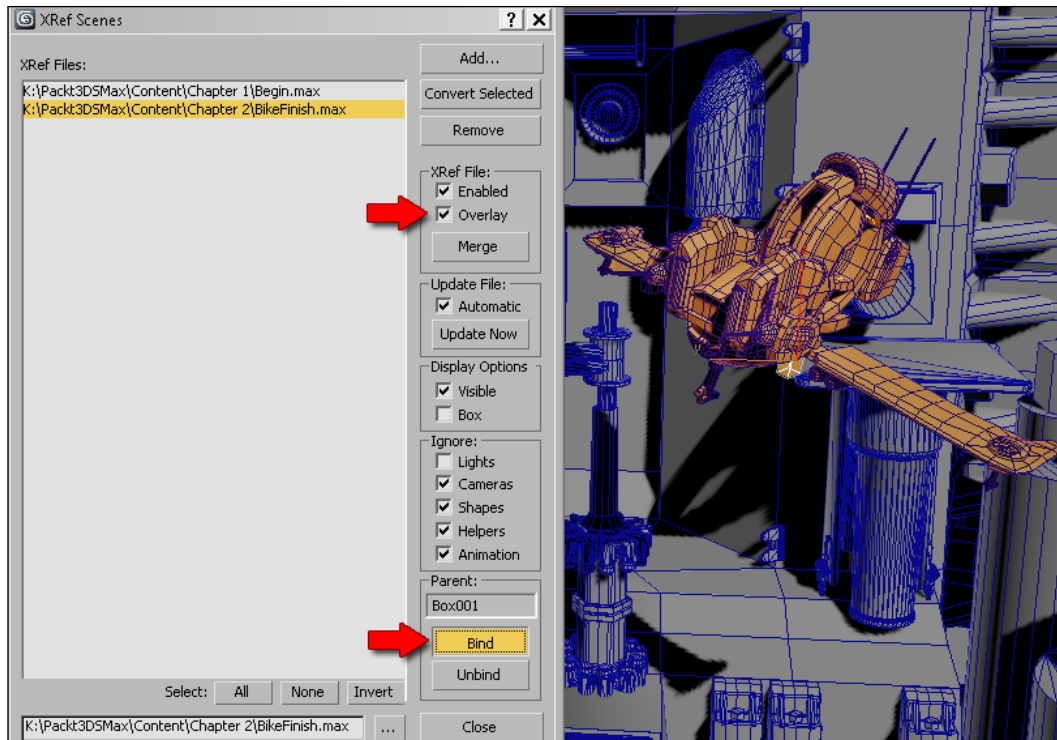
This is shown in the next screenshot:



## XRefs

**XRef Objects** and **XRef Scenes** are kind of like phantoms that you place in your working or master scene that actually exist (for creation and editing purposes) in another scene. They can't be modified in scenes into which they are placed but can be used to **Snap To** or **Align To**. It is not possible to move, rotate, or scale XRef content, but you can link it to a local dummy object that is transformed instead, and they'll inherit the transform. An example of XRef usage would be if you are required to build and animate a plane flying across a backdrop where the environment is only partly built (that job being perhaps the responsibility of another colleague).

You could XRef the existing backdrop into your scene and start animating your plane, and whenever the other artist saves their scene as they work, the content is updated. You wouldn't be able to change the backdrop, but you'd have its content visible to reference during the animation task. You can create circular references for XRefs (if the environment artist needed to see your plane updating, for example) or mask part of the XRef content from other users. See **Help | Autodesk 3ds Max Help... | Managing Scenes and Projects | References | External References (XRefs) | XRef Scene** in the **Overlays** topic for examples of how multiple scene references might work.



To access **Reference** settings, choose **File | Reference | XRef Scene** (or **XRef Objects**).

If you turn off the **Overlay** flag, shown in the previous screenshot, for an existing XRef Scene, you can cause circular external references to occur. If you don't notice this, it wouldn't be detected until you or another user tries to open one of the scenes in the project.



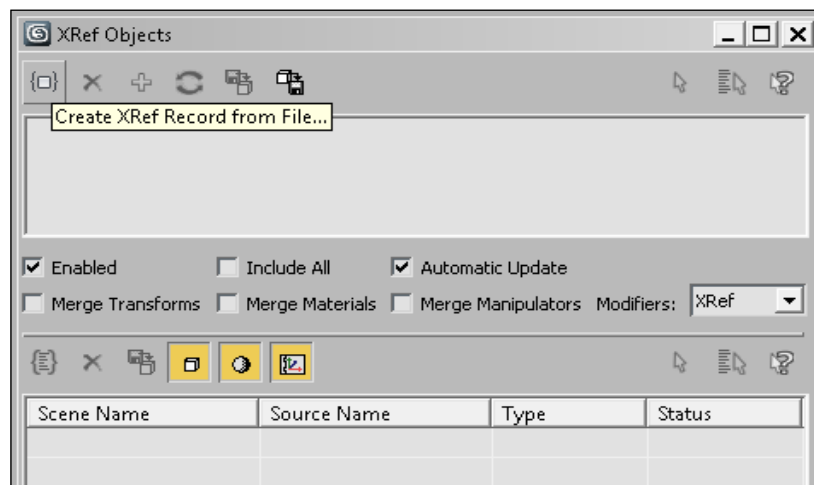
The concept of an Overlay is implemented through a toggle in the **XRef Scenes** dialog that pops up when you choose **File | Reference | XRef Scene**, after clicking on **Add** and browsing for a suitable scene. Linking XRef content to an object in the scene is also done via the **XRef Scene** dialog and requires you to specify an object to bind to after you choose **File | Reference | XRef Scene** then click on **Bind**. In the same dialog, the filters under **Ignore** let you filter out unwanted features of the XRef content, such as animation and lights local to the XRef Scene.

Like Containers, XRef content can be nested. An XRef Scene containing other XRef Scenes must have its own **Update File: Automatic** toggle ticked on for the nested content to update within it.

## Working with an XRef object in a master scene

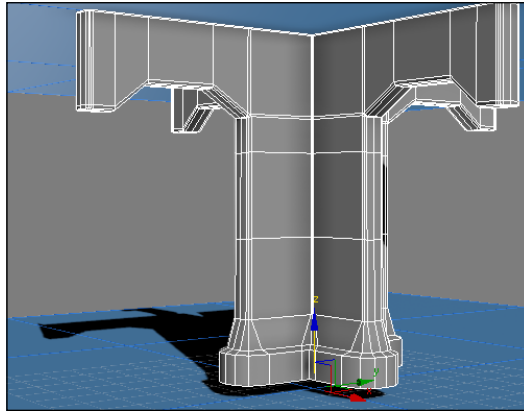
XRef Objects resemble XRef Scenes, letting you place objects between scenes that update from the source, but the interface differs. Let's import an object as an XRef in a working scene, then update it at its source, and then verify the change in the working scene.

1. Open the scene `\Packt3dsMax\Chapter 2\XRef_WorkingScene.max`. This is a scene that has a floor, a ceiling, and a camera.
2. Go to the **File** menu and choose **References**, then click on **XRef Objects**.

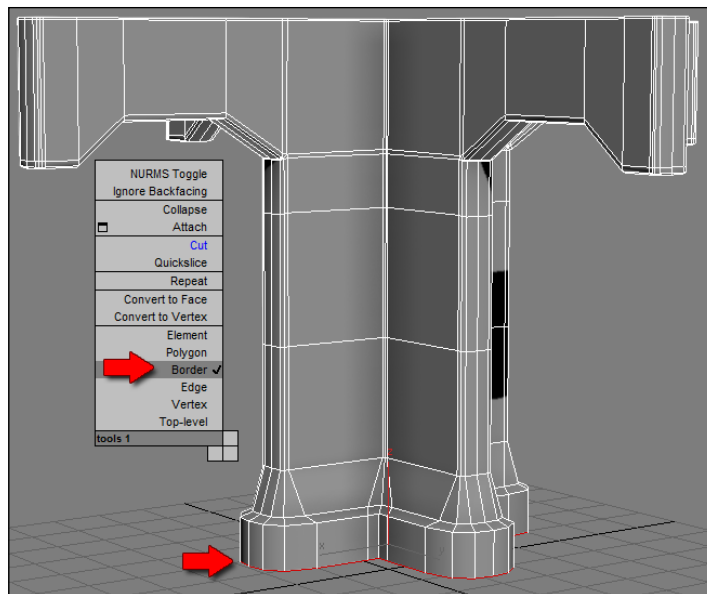


3. A floating **XRef Objects** dialog appears, as shown in the previous screenshot.
4. Click on the first icon, **Create XRef Record from File...**, and browse to choose `\Packt3dsMax\Chapter 2\XRefObject.max`.

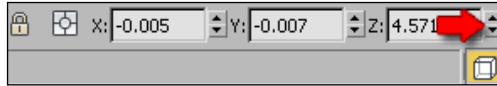
5. The scene will then appear in the panel below and will also show in the current scene, relative to its position in the source scene.
6. Note that the object, a modular column, is slightly off the floor. Let's imagine that our project requires us to keep the floor at its current height, so we have to change the XRef. Actual changes in production would likely be more dramatic.



7. Save the scene as `XRef_WorkFile.max`, and open `\Packt3dsMax\Chapter 2\XRefObject.max`, the source file for the column.
8. There, right-click and choose the **Borders** editing mode, or press 3, then click on the lowest edge of the column, as shown in the following screenshot:



9. Press *W* to enter **Move** mode, and then at the bottom of the screen, right-click on the spinner for the **Z** position to automatically set the selected border edges to **0.0**.



10. Save the file, and then open again the `XRef_WorkFile.max` file you saved before. Note that the XRef content will now be sitting nicely on the floor at **World Z**, there too.

That's how easy it is to update content from an asset referenced in multiple scenes.

## Summary

This supplemental coverage of Container and XRef scenes lets you see that 3ds Max provides strong tools for collaborating in complex scenes. At first encounter, Containers probably seem really confusing because it sets content at one remove from the user and you have to be aware of the possible connections between assets and users. Topics related to this are Scene States, Autodesk Vault, and the Asset Library found in the Utilities section of the Command Panel. While interesting topics, to cover them all would take us away from our major focus on modeling. This chapter's topic was handling assets, broadly speaking, and in *Chapter 3, The Base Model – A Solid Foundation in Polygon Modeling*, we begin to closely examine modeling processes.

