

**FLEX BUILDER 2.0.1 UPDATES  
TO FLEX 2: ADVANCED VISUAL PROGRAMMING  
Copyright© 2007 – Leo Schuman, Adobe Systems Inc.**

**Importing Projects into Workspace**

**Problem:** unable to import existing projects into workspace. Since this video course was released, Adobe has released Flex Builder 2.0.1. This update changed the structure of the metadata within project files in a manner which prevents the updated tool from recognizing project files created in prior versions of Flex Builder. So, if you are using Flex Builder 2.0.1, there is no way to globally import the project files distributed with the current version of the course as described in the Flex 2: Advanced Visual Programming video.

To determine if you are using Flex Builder 2.0.1, watch the start-up splash screen, or go to Help > About Adobe Flex Builder inside the product. The version should read exactly "Version 2.0.1". If it does not, you are not using Flex Builder 2.0.1.

**Solution:** take the following steps in Flex Builder 2.0.1 to import and configure each project, at the start of each lesson. This assumes you have created the workspace as directed in the video.

1. Right-click over the Navigator View, and select New > Flex Project
2. Leave Basic selected, and press Next
3. Name the project for the current lesson (e.g., Part 01 Lesson 01)
4. Uncheck Use Default Location
5. Browse to the current project files (e.g., ...\\Project Files - Flex 2 AVP\\Part 1 Lesson 01)
6. Press Next
7. Next to Main application file, press Browse
8. Select one file in your project as the initial default application (e.g., NaturalExperiences.mxml). The default application can be re-assigned at any point.
9. Press Finish, and the project should be created in the Navigator View
10. Now, three linked folders (ne\_assets, data, and gallery) must be created, pointing to shared assets used in the video
11. Right-click over the new project, and select New > Folder
12. Verify that the project itself is selected as the parent folder
13. Specify ne\_assets as the folder name
14. Press Advanced
15. Check Link to folder in the file system
16. Press Variables
17. Press New
18. Name the variable ne\_assets
19. Click into the Location field
20. Press Folder
21. Browse to ...\\Project Files - Flex 2 AVP\\ne\_assets
22. Press OK to create the variable
23. Ensure the new ne\_assets path variable is highlighted, and press OK to select it
24. Press Finish, creating a linked folder to the path defined in the new variable
25. Repeat steps 11 through 24 to create two more linked folders, using path variables
  - a. linked folder name: data
  - b. path: ...\\Project Files - Flex 2 AVP\\\_sharedData
  - c. linked folder name: gallery
  - d. ...\\Project Files - Flex 2 AVP\\\_sharedImages
26. Notice that a small arrow icon appears next to a linked folder name in the Navigator View
27. You should now be able to run the file
28. The linked folders must be created for each lesson, however you need only select the appropriate path variable
29. Alternately, you can browse directly to the correct folder (as specified above) for each linked folder, and omit using path variables entirely

## Filtered Images Displaying Incorrectly, and with Slow Performance

**Problem:** in the completed application and during later walkthroughs, the Gallery does not display properly when filtered, and the application performs slowly. Once again, a change related to the Flex Builder 2.0.1 release impacts the files used in this course. In this case, it is behavior related to the ArrayCollection class.

**Solution:** To manage the updated behavior of the ArrayCollection class, in the Flex Builder 2.0.1 release, two changes must be made to the code used in the NaturalExperiences.mxml files (whichever versions you're working with, presuming you've reached the point in the course where the Gallery is created). Code in bold must be added to the corresponding functions, which already exist in the course files. First, the photoData ArrayCollection must now be sorted prior to filtering. Second, the automatic data binding update behavior, new with Flex Builder 2.0.1, must be disabled before the loop, and re-enabled after, otherwise Flex will attempt to continually update its binding to objects within the ArrayCollection during the looping process, severely degrading performance.

```
// handle the photo XML data retrieved via HTTPService
private function photoHandler(event:ResultEvent):void {
    // add the retrieved data to the array collection
    photoData = event.result.photos.image;
    trace("photoData: " + photoData.length);
    //set the array length in application
    categoryItemCount = photoData.length;
    // re-sort the updated ArrayCollection
    var categorySort:Sort = new Sort();
    categorySort.fields = [new SortField("filename")];
    photoData.sort = categorySort;
    photoData.refresh();
}

private function calculateColumnPositions():void
{
    [ALL CODE PRIOR TO LOOP OMITTED FOR BREVITY]

    // disable the data binding before the loop begins
    photoData.disableAutoUpdate();

    // loop over all the images in this category
    for (var i:int = 0; i < categoryItemCount; i++) {
        [ALL CODE INSIDE LOOP OMITTED FOR BREVITY]
    }

    // re-enable the data binding after the loop finishes
    photoData.enableAutoUpdate();
}
```