

Level: Intermediate
Works with: Discovery Server
Updated: 01-Jul-2002

by Wendi Pohs
 and [Dick McCarrick](#)

As explained in the *LDD Today* article "[Creating meaningful K-map taxonomies](#)," the Discovery Server K-map is a critical tool for helping you manage your collective store of corporate knowledge. The K-map is a visual representation of your content, organized and structured in a way that complements full-text search, helps your users navigate to the precise information they need to achieve their goals, and reveals affinities between user profile and document information. Discovery Server creates the initial K-map, which you can modify and adapt to the needs of your organization.

But we're not going to lie to you—building your first K-map takes creativity and work. This is especially true if you attempt to build an all-inclusive, ready-to-deploy K-map rather than a more limited pilot. Fortunately, you may be able to jump-start a great deal of this effort by working with your Discovery Server administrator to import an existing operating system file hierarchy into your K-map and using it as the basis for your taxonomy scheme. This can be a significant time-saver for sites that currently store a great deal of content in a well-organized and maintained file/folder structure. Your K-map can then serve as the front-end to your data repositories, helping users quickly find the documents they want, without slogging through disk after disk, hoping the cryptic file and folder names somehow lead them to where they want to go. Equally important, the K-map can also point them to other documents in other repositories, creating categories of documents that extend far beyond the original folders upon which they were based.

There's an additional point to consider: if you already have a taxonomy in another format, you might want to create a file system format and import it into your K-map, even if this file/folder hierarchy doesn't already exist. At first this might seem counter-intuitive. Why create a separate file structure just to import into K-map? Wouldn't you save yourself a step by just using Discovery Server features to create the K-map from scratch? It's certainly a valid question. But as we'll explain in this article, in some cases using an existing taxonomy to create the file structure first and then importing it to mold your K-map is a viable option.

This article describes how you can import a file system taxonomy into a K-map. We explain how to prepare for and use the Taxonomy Import sample program that accompanies this article to import the taxonomy and offer tips for categorizing the resulting K-map to make it more meaningful to your users.

This article assumes you're an experienced Java programmer who is familiar with basic Discovery Server terminology and features. You can download the Taxonomy Import sample program from the [Sandbox](#). This Java program is also part of the Discovery Server 2.0 API Toolkit.

To view Lotus Discovery Server 2.0 API product information, see the [LDD Discovery Server page](#). For Discovery Server documentation, see the Lotus [Documentation Library](#). And for more information about Discovery Server and knowledge management in general, consult *Practical Knowledge Management: The Lotus Knowledge Discovery System* by Wendi Pohs, which is available from [IBM Press](#).

File system taxonomy versus the K-map

Many of us are probably more familiar with taxonomies than we realize. For example, Windows users see one whenever they open Windows Explorer and browse through a disk—all those folders, subfolders, and files comprise a very simple taxonomy scheme.

Of course, actually finding something through Windows Explorer can be tedious at best, especially if you're searching through hundreds or even thousands of documents. You might navigate through level after level of nested folders, looking for the document you need, only to discover you're searching on the wrong disk, or maybe even the wrong PC. So you start over, or (all too often) just give up.

This is where a K-map can make your life easier. You can import a file system hierarchy into a K-map and use it as the foundation for your taxonomy. You can then use the full suite of Discovery Server features to help users search, determine document and usage value, and derive affinities from this content. Perhaps most important, you can add documents from other sources, like Notes databases or Lotus QuickPlaces, giving users access to information stored in computers across your organization.

Sounds exciting—but before you get underway, you should consider the following:

- Do you have an existing taxonomy structure you'd like to use as the basis for a K-map?
- What changes can you make to your file system to help make the resulting K-map more useable?
- How does the import program work?
- What will you need to do to the taxonomy once you've imported it into your K-map? In particular, what should you be thinking about when converting file folders into K-map categories?

Finally, there's another question to ponder. Should you create a brand new file/folder taxonomy, just to import into your K-map? At first, this may not seem to make a lot of sense. Isn't the whole point of importing a taxonomy into K-map to preserve an *existing* file structure? Good point, but consider this: you may already have a clear idea about which categories and documents you'd like to see in your K-map. When you instruct Discovery Server to build your first K-map, chances are these categories won't be there, at least not in the exact hierarchical structure you may envision. In all likelihood, you'll use the K-map editor to rename and restructure categories, moving them around and creating new ones until you have the exact K-map you want.

You can minimize some of this editing by first creating a file/folder hierarchy in your operating system and then importing it into a K-map. This way you control how categories are initially created, named, and structured. This can save work if you have an existing taxonomy or if you're familiar with your content and know how you want it organized. At the same time, you should bear the following in mind:

- Include at least 15 to 20 representative documents in each folder.
- Be flexible in allowing Discovery Server to create additional categories. One of the major strengths of Discovery Server is its ability to quickly read many thousands of documents, analyze them, and find connections and affinities you might not have thought about—in essence, tell you things about your corporate data you might not have already known.
- Creating a new file system taxonomy to import is a way to jump-start your K-map. Don't devote the rest of your life to trying to gather all your documents from computers scattered throughout your site, figure out what's in them, create categories accordingly, and then stuff each one into the correct slot—that's Discovery Server's job. The file system structure should only be used to create K-map categories. You can later populate these categories with documents using Discovery Server itself.

Although creating a new file/folder hierarchy to import into K-map may not be the best solution for all sites, it is a useful option for some sites. And it does offer your users the advantage of working with a hierarchy that looks familiar to them

Preparing to import your taxonomy

There are several things to think about when you use the Taxonomy Import sample program we provide to import an existing file/folder structure:

- The Taxonomy Import program uses the Discovery Server 2.0 API, the programmatic interface for accessing metrics, search, K-map, and spidering functionality. The API gives you hooks into the Discovery Server code that provides these features. This lets you tap into Discovery Server functionality when writing your own Java programs or when using the sample programs we provide. The Discovery Server 2.0 API is installed separately from Discovery Server.
- You need the Java 1.1.8 JDK to run Taxonomy Import. You can download the Java JDK from the [Sun Java Web site](#).
- You run the Taxonomy Import from the Discovery Server command prompt, so you'll need to gather the information the Discovery Server needs in advance. You provide this information when you type the

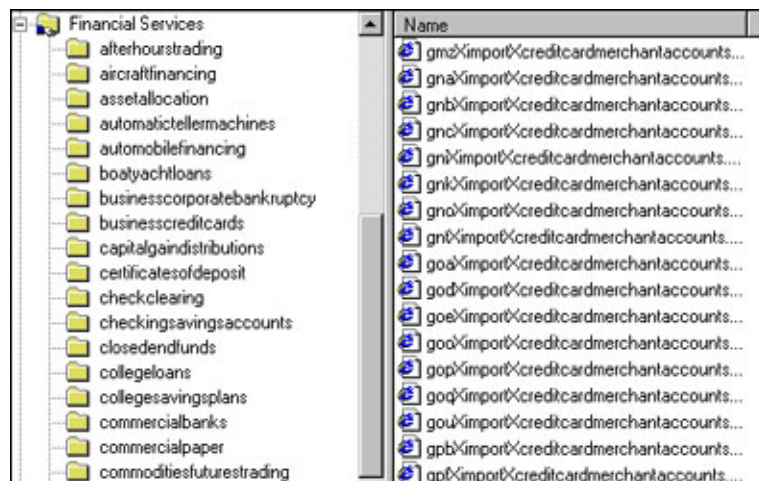
command line to run Taxonomy Import.

- You need to add yourself to the Administrators, KmapEditors, KDSAdministrators, and DiscoveryServers groups to run the Taxonomy Import sample program.
- You need to be sure that the File System spider is enabled on your Discovery Server.

Content considerations

Before you start, decide on a Top Term to use for your K-map. We've chosen Financial Services for the example in the following screen. In this case, we want to create a K-map to locate documents from many sources in a large financial institution. We know these documents fall into certain categories with very familiar names like Investment or Mergers and Acquisitions, so we want to preserve this naming system. We also know we could easily come up with 10 to 20 documents that represent the kind of content we'd expect in each of these categories. In some cases, the folders already existed, but we had librarians and content experts create folders and add content to some of the folders. We just want to jump-start the process; we know we'll need to rearrange and add additional categories and documents later.

Before running Taxonomy Import, our file system looked like this:



Notice that we didn't create a subject hierarchy at this point; we merely collected existing files and folders. We planned to rename categories and rearrange the hierarchy, using the K-map Editor, later on. Although the file system we used contains HTML files, import can collect most of the file types you'll have on your system. But remember an important Discovery Server caveat: the better the metadata, the better the Discovery Server results.

Running the Taxonomy Import sample program

As we mentioned before, you run the Taxonomy Import program from the Discovery Server command prompt using a number of switches. In the example described in this article, the full command line we used is:

```
taximport C: C:\jdk1.1.8 /tt:Financial Services /rt:5 /f:\wlpohs\Financial Services /ksvr:wlpohs.test.com /kspwd:password /kusr:user /kupwd:passwd
```

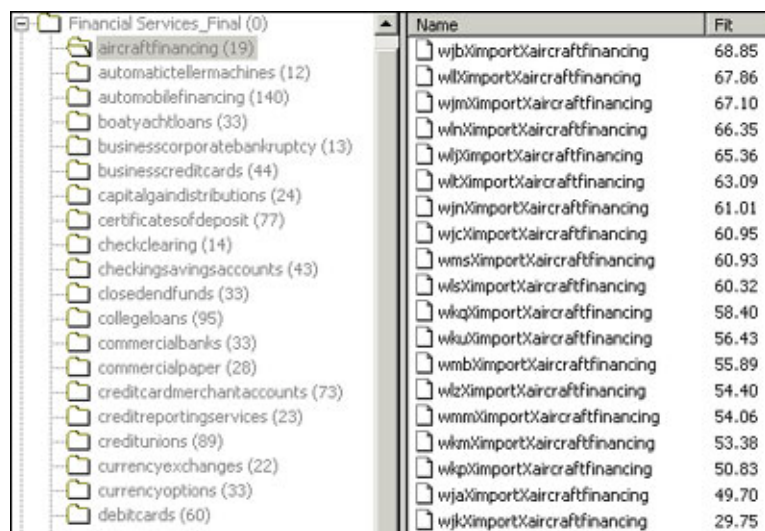
Notice we included the following switches:

- /tt:Financial Services
The \tt switch tells the program what you want to see as your Top Term inside the K-map Editor. Once Taxonomy Import runs, you'll see this category directly under the Home category.
- /rt:5
The /rt switch tells Taxonomy Import what kind of repository to expect. For import, you'll always use 5, which tells the program to use the file system.
- /f:\wlpohs\Financial Services
The /f: switch indicates which file path the program should use to get your information. You can start anywhere in your file system. In our example, we had a Financial Services folder at the top level of a machine called wlpohs and we wanted to use it and all its subfolders to build the K-map.
- /ksvr:wlpohs.test.com
The /ksvr: switch tells the program the name of the Discovery Server machine.
- /kspwd:password
The /kspwd: switch is the server password.

- /kusr:user
The /kusr: switch is the Discovery Server user name. This will usually be the Administrator, but you can use your user name if you are a member of one of the groups mentioned above.
- /kpwd:passwd
The /kpwd: switch is the password for the user you entered.

Once you've told it where to go and what to do, Taxonomy Import taps into the existing Discovery Server processes. First, it creates a temporary category in your K-map to hold all imported documents. It then creates a Data Repository record that you can view in the Discovery Server Control Center. You can monitor its progress by looking at this record. You can also watch the output that appears in the command prompt window.

Taxonomy Import takes a little time to complete. When you see a message such as "147 files(s)done," you can look at your new K-map using the K-map Editor. Our Financial Services K-map initially looked like this through the K-map Editor:



Name	Fit
wjbXimportXaircraftfinancing	68.85
wllXimportXaircraftfinancing	67.86
wjmXimportXaircraftfinancing	67.10
wlnXimportXaircraftfinancing	66.35
wljXimportXaircraftfinancing	65.36
wltXimportXaircraftfinancing	63.09
wjnXimportXaircraftfinancing	61.01
wjcXimportXaircraftfinancing	60.95
wmsXimportXaircraftfinancing	60.93
wlsXimportXaircraftfinancing	60.32
wkqXimportXaircraftfinancing	58.40
wkuXimportXaircraftfinancing	56.43
wmbXimportXaircraftfinancing	55.89
wlzXimportXaircraftfinancing	54.40
wmmXimportXaircraftfinancing	54.06
wlmXimportXaircraftfinancing	53.38
wkpXimportXaircraftfinancing	50.83
wjaXimportXaircraftfinancing	49.70
wjkXimportXaircraftfinancing	29.75

Category care and feeding

All K-maps—even ones created through a file system import—require at least a little editing before they're ready to be rolled out for general use. An imported taxonomy can help minimize this step by letting you predetermine the K-map categories and how they're arranged hierarchically. But creating a K-map this way does bring with it several points to consider.

For example, when the K-map Builder creates a category, it does so by examining and analyzing the contents of a number of documents, calculating fit values for them, and then grouping them together accordingly. Therefore, documents in a category are likely to have high fit values, because Discovery Server determined this before creating the category. A category created through a file system import, however, has not undergone this process. Documents within that category are there only because they were located in the folder from which the category was created. These documents may or may not have similar content. For example, you might import a folder called Drafts containing documents that have nothing in common other than that they're waiting to be reviewed. The fit values in a category such as this could be very low. This could impair an important feature, the ability to "train" Discovery Server to classify documents into the categories you want. If Discovery Server is unable to find significant similarities between the contents of documents in a category, it won't be able to determine which new documents should go into this category in the future.

To help avoid this, you should be sure that each category you create has at least one sibling category at its level. This way, Discovery Server can better distinguish between the categories and determine which documents belong in each. Also, try to keep the number of documents per category reasonably balanced. If a category has far more documents in it than a sibling category, Discovery Server will tend to place new documents into it, regardless of how high the fit values are in the smaller category. Ideally, parent categories should contain between three and nine child categories.

In addition, you should consider starting small. Create your K-map with a limited number of documents, rather than your full content set. When the fit values for categories looks high, you can start adding more documents to

the K-map. At the same time, don't go *too* small. Avoid creating categories with fewer than ten documents. If necessary, combine smaller categories.

If you subsequently add new documents to your K-map and they don't end up where you expect, you should try the following:

- Evaluate the fit values of the documents in the category. If the fit values are high, check the number of documents in the category and in its siblings. Balance the number of documents in the categories whenever you can.
- Check the number of sibling categories. Create a higher level parent category to avoid having too many siblings at the same level.
- Move documents with fit values lower than 30 into more similar categories, or create a large "holding bin" category that you can subdivide later.
- Check the documents in the Uncategorized category. You may be able to create a new category if these documents are similar. If you do this, be sure to retrain this new category and evaluate the new fit values.
- Subdivide any disproportionately large categories at any level. Make sure the Maximum Number of Documents setting does not exceed 500 documents. Set the K-map setting to automatically subdivide categories with more than 500 documents to avoid large categories.

In our example, after an initial editing pass (which took only a couple of hours) our imported taxonomy was ready for categorization. To improve categorization performance, we created one set of parent categories and moved several child categories under each. We promoted each of these parents to the top level and deleted our original Financial Services working category. We cleaned up category names where needed and examined the fit values for the documents in each category. Finally, we published all the categories so they would be visible to users through the K-map UI. In its final form, our imported K-map looks like this:



Importing a taxonomy: one way to a quicker K-map

As we've shown, importing an operating system file/folder taxonomy is a useful technique for building a K-map. This might not be the best option for everyone. But for sites with well-organized file systems, this could save a significant amount of time. We suggest you consider importing your file system taxonomy when planning and creating your first K-map. It could help you get a head start on deploying and using your K-map, an important tool for understanding and navigating the vast and varied landscape of your accumulated corporate knowledge.

ABOUT THE AUTHOR

Wendi Pohs is a principal taxonomy specialist on the Discovery Server team and the author of a book about knowledge management methodologies, *Practical Knowledge Management: The Lotus Knowledge Discovery System*, published by IBM Press. Wendi joined Lotus Development Corporation in 1996 and has worked on various projects as a spec writer, online help

Lotus Developer Domain: Importing a file system taxonomy into a K-map
www.lotus.com/ldd/today.nsf

designer, and user assistance manager. Prior to joining Lotus, Wendi worked at the American Mathematical Society and at Digital Equipment Corporation. Wendi received her BA and MILS degrees from the University of Michigan.