

How To Build an Effective Knowledge Map

by Jason Bargent

In April 2001, IBM/Lotus Software released its long-awaited flagship product in the Knowledge Management arena — the Knowledge Discovery System, or KDS. This product launched Lotus into the knowledge management portal and information/expertise location market. Two main components of KDS were K-station and Lotus Discovery Server (LDS). The Lotus Discovery Server is a back-end knowledge server that indexes, or *spiders*, documents and your organization's directory to create a *taxonomy* of documents, expertise, and associated relationships (see "Glossary of K-map-related Terms," below, for an explanation of these and other terms).

The new release of LDS (v1.1) removes its dependency on Lotus K-station, making the Discover Server a standalone offering. New APIs also provide greater ability to build a custom end-user interface to LDS (as used by IBM partners TheBrain, <http://www.thebrain.com>, and Atomica, <http://www.atomica.com>).

An important capability of LDS is that it can generate a knowledge map, or K-map — a single interface that displays discovered people, places (communities), and information from multiple systems both internal and external to your organization. A K-map provides users easy access via a Web browser to specific information, which can save time, increase productivity, and encourage reuse.

This article focuses on how to build an effective K-map. (For more detailed steps on building a K-map, see "[11 Steps to Building a Knowledge Map](#).") But before we dive in, it's important to know what the Lotus Discovery Server can and can't do and how it works to better illustrate a K-map's functionality and capabilities.

Lotus Discovery Server Components

The main components of LDS are the K-map, people profiles, spiders, and the K-map Editor. (For more information about these components, see the various Lotus Discovery Server white papers and product documentation at <http://www.lotus.com/km>.)

K-map. The K-map is the taxonomy created from the spidered data sources. End users can employ a browser to search or browse for information and locate subject matter experts using the K-map. The K-map lets users search and sort information by its value (or ranking), title, date, author, and data source. They can perform searches against people, documents, document authors, people who know about a particular subject, the content of a person's profile, and categories within the taxonomy.

People profiles. You use people profiles to store information about people within your organization (e.g., name, job title, department) and an individual's expertise from his or her relationship to the data contained within the K-map (this expertise is known as affinities). People profiles are tightly integrated into the K-map so that whenever a person's name appears as the author of a document, clicking that name shows that person's profile along with his or her contact information and any discovered or self-declared affinities. A K-map can also search for experts on a particular subject using the affinity information stored within the people profiles. People profiles are synchronized with a primary source such as the corporate directory (e.g., Lotus Domino Directory, Lightweight Directory Access Protocol — or LDAP) and can be supplemented with data from other sources such as Lotus Domino databases (e.g., a Lotus Domino database that contains HR information or past project information).

Spiders. To build the K-map, LDS spiders the identified sources of data. Spiders are available to include data from Lotus Domino databases, file systems, and Web sites (internal and external). With the soon-to-be-released XML spider, you can easily add third-party data into the K-map. And by leveraging traditional Lotus Domino integration tools such as Domino Enterprise Connection Services (DECS) and Lotus Enterprise Integrator (LEI), you can also include non-Domino databases such as Oracle and SAP as sources for the K-map.

K-map Editor. Once the K-map is initially generated by spidering its identified data sources, subject matter experts (i.e., those who use and know the data) must teach the system the correct categorization and placement of documents within the K-map. To perform this task, they use the K-map Editor, a Microsoft Windows application that has an intuitive drag-and-drop interface.

K-map Security

As you'd expect, the Lotus Discovery Server limits user access to the K-map and the K-map Editor by a user name and password that's required for authentication. However, a successful Knowledge Management system that provides browse-and-search functionality across source data must also reflect user privacy as defined at the source. For example, it would be disastrous if a private HR discussion room containing information about upcoming redundancies was spidered, and the results were available for everyone to see in the K-map.

To overcome such problems, you can select only that information you want to include in the K-map (by using selection criteria). The spiders also reflect the security of the source documents in the K-map. For example, users who aren't included in a file system ACL, a Lotus Domino database, or a document ACL can view only information that they'd typically have access to when they browse or search the K-map. This technique ensures the integrity of the source information and that end users see only what they're allowed to.

Creating a K-map

When building a K-map, the Lotus Discovery Server won't automatically produce a working taxonomy for your organization. A process is required in which the subject matter experts teach the system. This method of teaching is best done in several iterations (see "[11 Steps to Building a Knowledge Map](#)" for more information). Because the system is dynamic, it will learn from each iteration, with each cycle reducing the K-map editors' efforts as the system correctly places new documents.

The steps required to build a K-map are as follows:

1. Identify requirements: For a K-map to be successful, it must be aligned with a business goal or purpose. The smaller the scope of this goal, the easier it is to measure the success of the K-map. Often, you might build enterprise-wide K-maps by progressively adding to departmental K-maps. Alternatively, you might have a working K-map for each department.
2. Conduct an information audit: Before you can target data sources for inclusion into the K-map, you must conduct an information audit. Initially, subject matter experts must identify all possible sources for the data and answer specific questions to ensure that they've captured enough information about a particular topic.
3. Define the information sources: Refine the list generated from the information audit and remove duplicate data, outdated information, and information of little value. The objective is to make a new list that will be the total source of information for the K-map. You then define a prioritized list known as the *training set*. This list identifies the most useful documents from which to train the initial K-map. During the information audit, it's important to check the validity of each data source, including its *metadata* and security rights.
4. Modify the stopword list: Update the default LDS list of words excluded from the K-map. You should modify this "stopword list" to include any organizational- or industry-specific terms.
5. Configure the Lotus Discovery Server: The first time you use LDS, you must configure the type of spiders to use, how many spiders to use, and various rules against each of the spiders. All these factors can affect performance and the result of the K-map and are dependent on the data sources and the business objective of the final K-map.
6. Generate the people profiles: As I mentioned earlier, people profiles are generated from a primary source such as your organization's Lotus Domino or LDAP directory. The Lotus Discovery Server uses its own method of storing personal information about people in your organization.
7. Set up data sources for spidering: Before building the initial K-map, you must configure each data source. Starting with the training set, you define each data source to its lowest level to ensure that spider-specific questions can be answered (e.g., how many links to follow for a Web spider, how many subfolders to include for a file system spider).
8. Create the K-map: Once you've successfully set up and reviewed the data sources, you can then create the K-map. To do so, the appropriate spider must read the contents of the data sources and position it into the K-map. The resulting K-map is unusable and must be trained by subject matter experts.
9. Train the K-map: The process of training the K-map is an iterative cycle of adding in new data sources, using

the K-map Editor to teach the system, and repeating this cycle to include all data sources. During the editing process, K-map editors (i.e., subject matter experts) use the Microsoft Windows-based K-map Editor tool. They review the *category* labels of the taxonomy and then review the position of each document in the category. As each iteration progresses, less activity is required as the system learns from the document placement and adjusts new documents accordingly.

10. Generate affinities: Once subject matter experts have trained the K-map, you're ready to generate affinities — the associations between people and information. To generate affinities, the system reviews an individual's connection to a document (e.g., whether he authored the document) and then updates that person's people profile with the category name for subject areas to which he or she has a close affinity. Users can approve their affinities before they're submitted and edit them; managers can recommend affinities.
11. Test the K-map: Before you release the K-map into the user community, it's best to undergo a level of end-user testing. This usability testing evaluates the effectiveness of the K-map and its taxonomy and affinities before going into production.

Once you've deployed the K-map, the spiders will automatically detect and reflect any changes to the identified data sources (e.g., additions will be categorized according to their fit in the K-map).

K-map editors must periodically monitor and edit the K-map to ensure correct categorization of new documents, addition of new categories where appropriate, creation of subcategories from large ones when necessary, and maintenance of views for uncategorized documents.

Considering whether to include new data sources will also affect the maintenance time K-map editors require. Many K-maps evolve over time as users establish new information requirements. Once they've added new sources, K-map editors should repeat the process of editing as defined in step 9. (For a snapshot of guidelines to follow when creating a K-map, see "Ensuring the Success of a K-map," below.)

Making a Strong Commitment

Above all, the success of a Lotus Discovery Server implementation depends on how well you understand and communicate the correct expectations. Creating a K-map requires a strong commitment from business or content experts to initially teach the system.

Although you may invest a lot of time implementing a successful K-map, the results are always well worth the effort and will produce tangible Knowledge Management results. The bottom-up approach that Lotus Discovery Server uses to generate the K-map ensures rapid results that are factual and a true representation of your organization's data and expertise (because the taxonomy is created from your true data, not what your data should be). You'll find other useful Knowledge Management references at <http://www.lotus.com> and at the Notes.Net KDS discussion forum (<http://www.notes.net/kdsforum.nsf>).

Jason Bargent is a solution architect for IBM Software Group Services in Melbourne, Australia.

Ensuring the Success of a K-map

To ensure the success of a K-map, you can follow these few simple guidelines:

- Don't skip any steps in the K-map building process.
- Read the documentation.
- Include data that is of value and exclude data that is not.
- Ensure that you have quality metadata.
- Identify a business objective or "pain" against which the K-map can be aligned.
- Start with a small K-map and introduce more data sources over time.
- Ensure that you conduct end-user testing before you release the K-map.
- Ensure that subject matter experts — not non-experts such as IT staff — train the K-map.

— J.B

Glossary of K-map-related Terms

Affinity: The term that Lotus Discovery Server (LDS) uses to define relationships between people and information, indicating that someone is an expert or experienced in a particular subject. A person's affinity can be automatically discovered, nominated, or self-declared. Users can view affinities from the K-map or a person's profile.

Category: A group of associated information in the taxonomy. The category name is also used to derive the affinity of a person who is related to a document.

Metadata: The data (such as the author, title, or date created) that describes a document in some way. Metadata can be included in the file properties of popular products such as Microsoft Word (select File, click Properties) or metatags in HTML code.

Spider: The process that LDS uses to read and extract useful information from the identified data sources. A spider exists for different data sources (Lotus Notes, file system, a Web spider).

Taxonomy: The classification of organizations' knowledge, indicating relationships between information by grouping relevant or similar information into categories and subcategories. The taxonomy in LDS is represented in the Knowledge Map, or K-map.

Training Set: The initial collection of data sources used to create the K-map. The training set is elected from the overall data sources as being a true representation of the data.