Lotus Developer Domain
The Technical Resource for Lotus software
LDD Today

Level: Intermediate
Works with: Discovery Server
Updated: 02-Dec-2002

by Robin Rouse,
Hiroyuki Miyamoto,
and Karen Bjorkman

What is the "official" language of knowledge? While French and English might be the official languages of the Olympic Games, knowledge knows no language limitations. Whether you work in a large multinational corporation or a small business, your company is probably creating electronic information in multiple languages every day. Discovery Server speaks the language of this global information and can help you to organize and leverage that knowledge. From the end-user experience to the inner workings of the spiders that glean the data, Discovery Server has been engineered to meet your multilingual knowledge needs.

This article provides an overview of Discovery Server's multilingual capabilities as well as tips for supporting the deployment of Discovery Server in a multinational, multilingual environment. It assumes familiarity with Discovery Server.

Multilingual support in Discovery Server 2.0

Discovery Server can process information and display its user interface (UI) and end-user Help documentation in the following languages:

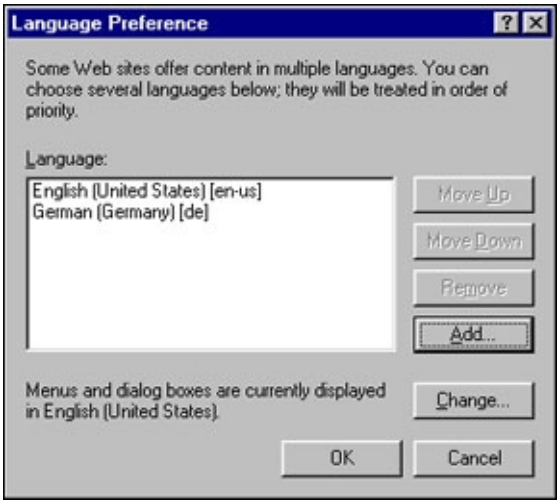
- Brazilian Portuguese
- Danish
- Dutch
- English
- Finnish
- French
- German
- Italian
- Japanese
- Korean
- Norwegian (Bokmal and Nynorsk)
- Simplified Chinese
- Spanish
- Swedish
- Traditional Chinese

There is support for character sets handling and date/time format for all of the locales associated with these languages. Also, English Discovery Server can be installed on a supported operating system in any of the above languages. (The Administration user interface and documentation are always in English.)

The national language versions (NLV) of Discovery Server are available on a CD or can be downloaded from the Web (for Discovery Server demos and downloads, see the **Lotus Discovery Server home page**). Each NLV of Discovery Server comes with that language's resource set and the English resources. The user interface language

that Discovery Server displays to an end user—the language of the text found on the labels, drop down boxes, buttons, and so on—is determined by two factors:

- The language preference of the user's browser settings (in Internet Explorer, this is specified in the Internet Options dialog on the Language tab)



- The language resource set installed on the server

Discovery Server first checks the language preference of the user's browser, and then checks to see if the server has that language's resources installed. If the server has the language resource, it serves UI in that language. Otherwise, it defaults to serving the UI in English. Proper display of the characters in any given language is dependent on proper font support on the end-user's client machine.

Search

The search feature of Discovery Server is language agnostic. Users can search for any term in any language that Discovery Server supports regardless of the language version of Discovery Server installed on the server. Discovery Server returns all documents that match the search query regardless of the end user's language preference. In other words, someone using the English version of Discovery Server can search for a term in French, and Discovery Server displays all matching documents that contain that French term. Remember that Discovery Server is not a translation tool. (If you search on "chien" [dog in French], you do not get documents that contain the word "dog" [in English]. However, Discovery Server has that capacity though the synonym dictionary. See the Synonym dictionary section later in this article for more information.)

If you perform a search on a term—say JAVA—Discovery Server returns all documents that match that search regardless of the language of the document. Documents are returned in their original language. Even though you may not understand the language of a particular document, it may still be useful to you. For example, if your search returns documents in a language you are not familiar with, you can paste the contents of the document into a Web site that instantly translates. This gives you the gist of what the document is about. If the document seems useful in your research, you can pursue having it translated or contact the author of the document.

There are a couple of things to keep in mind pertaining to having multiple languages contained in your K-map and subsequently available for a search query. With the exception of the Synonym dictionary, Discovery Server is entirely Unicode enabled. As long as you configure your browser properly for multilingual display and your operating system has proper font support, all text, taxonomy labels, document titles, summaries, and author names display correctly. Additionally, to properly display the contents of a document that has been returned by the search, the native application (for example, Lotus Notes) must have proper font support installed as well. If the operating system does not have proper font support or if your browser is not correctly enabled for Unicode display, you often see dots (...) or small squares in place of the correct characters. If characters do not display correctly, see your support organization about installing multilingual font support (for your operating system) and/or Unicode-enabling your browser.

Synonym dictionary

To make the search more effective, Discovery Server provides thesaurus or synonym capability via a synonym dictionary. For example, you can tell Discovery Server that Big Blue is equivalent to IBM and International Business Machines. When someone searches for IBM, the search results include all the documents about Big

Blue as well as International Business Machines. This feature can be used on a limited basis to create synonyms between languages. So if you wanted a search on the word car to return all the documents that also include the word coche (the Spanish word for car), you can include a synonym relationship between those two words in the thesaurus. The synonym feature supports ASCII characters.

Discovery Server's linguistic engine

The linguistic engine of Discovery Server performs language identification, summary generation, and tokenization of documents. The linguistic engine's first task is to identify the language of the document and then use that information for subsequent processing. So, accurate language detection is the key to improving the tokenization quality. Narrowing the language list is especially useful with double-byte character sets (DBCS) or Asian languages.

To narrow the language list, do the following:

1. Open the XML file \Lotus\DS\inxight\lx-2\lang33\std.langid-config in a text editor. Remember to back up the file before making any changes.
2. Under the tag <encodings-languages-covered> delete the language(s) you don't need.

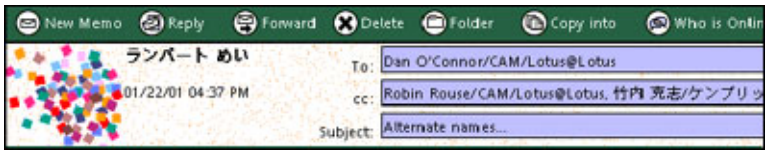
```
<encodings-languages-covered>
<!-- Some list tags have been omitted from this code sample -->
<list key = "utf_16">
<item key = "english" />
<item key = "french" />
<item key = "german" />
<item key = "italian" />
<item key = "spanish" />
<item key = "portuguese" />
<item key = "dutch" />
<item key = "swedish" />
<item key = "finnish" />
<item key = "danish" />
<item key = "bokmal" />
<item key = "nynorsk" />
<item key = "simplified-chinese" />
<item key = "traditional-chinese" />
<item key = "korean" />
<item key = "japanese" />
</list>
</encodings-languages-covered>
```

3. Save the file.
4. Restart the server.

The change applies only to newly processed documents.

Profiles: Alternate name support

Discovery Server alternate name support lets you specify an alternate name using non-ASCII characters. Alternate name (AltName) is a Domino feature which allows a user to have his or her name in his or her native character set. This feature is useful for user names that cannot be written in ASCII characters, as shown in the following screenshot:



In Domino, the AltName is a fully authenticated Notes ID. However, in Discovery Server AltName is for display purpose only. It's incorporated via the Domino Directory and added to the user's alias list. For some users, it's a preferred name representation. To display AltName, set the Notes.ini variable KDS_ENABLEALTNAME=1 and set the client browser's preferred language to the language that matches the AltName language. If the client machine has proper font support, the alternate name displays instead of the primary ASCII name. This applies to K-map UI and Profile dialog.

For more information about Domino Alternate Name support, see the *LDD Today* article, "**Think globally: Creating a multilingual Notes/Domino environment**."

Spider basics

Spiders are the tool that Discovery Server uses to glean information from the data repositories. Discovery Server 2.0 ships with the following spiders: Notes spider, Web spider, File System spider, XML spider, and Exchange spider. These spiders generate XML that contains the content of the original document and attachments, as well as metadata about the document for further processing.

From a multilingual perspective, the overview of the spider process is:

1. Read a document from the repository. If the document is from the file system, a Web site, or an attached file, the spider uses the file filter module or the HTML filter to process the document.
2. Convert the document to Unicode from its native encoding and encode it in a structured XML.
3. Detect the human language of the document using the linguistic engine.
4. Generate a summary for the document using the linguistic engine if one does not already exist.
5. Tokenize the document using the linguistic engine according to the rules associated with the language determined in step 3.
6. Count the tokens and include a vector of token counts in the XML for the K-map builder.

Each of the different spiders has some unique characteristics when it comes to multilingual documents. The following table provides a brief summary of each spider.

Spider	Description
Notes spider	The Notes spider is fully multilingual-enabled thanks to LMBCS (the Lotus Multi-Byte Character Set). Attachments, stored in Lotus Notes documents are processed using the file filter module. The file filter's job is to extract the text and to strip any nonsignificant data (font attributes, images, etc.) from the attachment and then pass it on for XML-encoding. To get optimal fidelity from documents stored as attachments, consider upgrading their format to a file type which supports correct file filtering to Unicode, such as Lotus Notes. You should check the release notes of Discovery Server for a list of issues with specific file types. (The Notes spider is also used for QuickPlace, Domino.Doc, and Notes email.)
File System spider	The File system spider relies heavily on the file filter. Therefore, the same considerations apply as they do with the Notes spider. The only file type that is not filtered by the file filter is HTML files. HTML files are filtered through Domino's HTML parser module.
Web spider	The Web spider uses Domino's HTML parser. The HTML parser is very good. If you know that you will always spider Web sites that match the code page of the operating system of the server, you are likely not to have any multilingual or code page issues with the Web spider. Unfortunately, though, many Web sites do not express their encoding properly according to W3C standards. If you run across this, one way to get around it is to configure the "preferred codepage list" in Domino. For configuring Domino's MIME preference, see the *LDD Today* article "**Worldwide messaging: Using International MIME in R5**."
XML spider	The Discovery Server XML spider uses the Xerces/Xalan XML parsing engine, and thus the supported character sets are constrained by those supported by the engine. The list of supported languages for Xerces/Xalan is fairly thorough and covers all major character sets. However, when you are generating XML data, we recommend that you always generate it in one of the Unicode encoding schemes, namely UTF-8 or UTF-16BE, so that you are guaranteed that your XML is read correctly by any XML parser.
Exchange spider	The Exchange spider processes attachments in the same way as the Notes spider. The document text is retrieved from Microsoft Exchange using Messaging Application Program Interface (MAPI) and is returned in Unicode format.

The K-map builder
The K-map builder works mainly with the tokens that the spiders supply. It maintains each unique token and document ID. What matters most in good multilingual processing is how accurately tokens are extracted from documents. Tokenization consists of segmentation, case normalization, stemming, and POS tagging, which are all performed by the linguistic tool during the spidering process. The following types of tokens are thrown away and are never passed to the K-map builder:

- Numbers
- Hexadecimal numbers
- Punctuation
- Apostrophe/conjunctions and the equivalents in other languages
- Possessives and the equivalent in other languages
- Single alphabet letters in isolation, such as a, b, c, and so on
- Email addresses, file locations like C:\DS, and URLs
- Tokens containing any of the following symbols: `~!@$%^&*()+-=[]{}\|;:'",.<>/?
- Conjunctions, such as "and" and "or" (only applies to Chinese)
- Time phrases, such as Monday (only applies to Chinese)

The spiders pass the remaining tokens to the K-map builder, but before further processing is done, the K-map builder performs stopwords filtering. Stopword filtering is a process of tossing away less meaningful and too frequently used words, making sure that they are not used for document categorization. The stopwords are provided in a Unicode text file. They are located in the \Lotus\DS\Data directory and named stopwords_*xx* *.txt. The suffix in the file name indicates a language.

Because stopwords are stored in a simple text file, users can add (or remove) words that are less important for the organization or vice versa. If you see many less meaningful words appearing in your initial taxonomy, you may want to add those words to one of the stopwords list and rebuild the taxonomy. For instance, in Lotus, you find the word Lotus appearing often in documents. Adding Lotus to the stopwords list removes the possibility that Lotus will become a category, which helps reduce the "noise" in your taxonomy.

Another step in the K-map builder is case normalization in Western languages. This prevents Conference and conference from being counted as two distinct tokens. Many language versions of Discovery Server are shipped with the lowercasing everything switch turned on (German is switched off). It's configurable via the Notes.ini setting DS_LOWERCASE_KMAP.

The K-map Editor
The K-map Editor application is installed and runs on a local client machine. The K-map Editor user interface (UI) language is determined by the operating system (OS) language (as opposed to the browser language preference the K-map UI uses) of the machine that it is installed on, as well as the language resources available on the server. When you install the K-map Editor from the server, it installs whatever language resources are available on the server. For example, if the server is Japanese Discovery Server, it has the Japanese K-map Editor installer. At runtime, K-map Editor tries to present its UI in the language of the client OS. If that language's resources are not available on the client, the K-map Editor defaults to English. The K-map Editor can display documents, category labels, or document titles in any language that Discovery Server supports regardless of the UI language. The K-map Editor allows the user to change the font to display the proper character for multilingual taxonomies. To display fonts from different character sets on different code pages, you must have a full Unicode font installed. For more information on Unicode fonts, see the **Unicode FAQ - Fonts and Keyboard** Web page.

Best practice: Generate taxonomy based on the single common business language

Now that you've read about the Discovery Server's multilingual features, you can use the tips in this section to plan and create a taxonomy for your multilingual environment. As with the creation of any taxonomy, a well-designed multilingual K-map begins by examining the business goal of this taxonomy, its audience, and the data in the organization that best addresses the questions you are trying to answer or the problems you are trying to solve. The ideal taxonomy for your business application represents the mental model you offer your users. (For further information on building K-maps, see the *LDD Today* article "**Creating meaningful K-map taxonomies**.")

Addressing a multilingual audience and data set presents some specifically linguistic issues, so consider the following:

- Three factors shape the initial taxonomy: one, the distribution of spoken languages and locales for the audience and expected access of the audience to the secondary languages represented; two, the distribution of languages in the data set; and three, its subject matter and how technical that content is. It is very important that you consider these factors when assembling the training set from which the initial taxonomy is

built, because the initial taxonomy influences the way documents are later classified. For example:

- If your data set is heavily weighted towards one language—90% English and 10% German, say—your initial taxonomy might contain one or two categories in which all German documents are placed, regardless of content. It is also likely that the categories containing German documents will exist in their own subsection of the taxonomy.
- If the same distribution of languages exists for a data set consisting of IT or otherwise highly technical data, it is more likely that German documents and categories will be scattered among the predominantly English categories. In fact, in the IT business area multilingual data sets work out well because IT uses so many of the same (English) terms.
- Distribution of languages in the initial taxonomy influence the way in which documents in other languages are later classified. If your initial taxonomy is built using English and German documents, then classifying Spanish documents later on will probably cause the Spanish documents to be placed in the Uncategorized category, unless the Spanish documents share some technical tokens, for example, Lotus, IBM, OS/2, and so on.
- If your data set consists of team rooms for three different locales: UK, Spain, and Japan, and there is an equal distribution of documents in English, Spanish, and Japanese, your initial K-map is likely to subdivide into three distinct subsections. In that case, if your business goal specified multilingual categories, you might distribute languages into categories as suggested later in this article; if it better serves your audiences, you may decide to divide the three into separate K-maps for separate distribution.
- How pertinent is the language difference—are the documents in Spanish contributed by a Spanish-speaking sales force and fairly coherent, targeted to the locale of the specific language? Or is the language difference not relevant to the goal of the taxonomy, authored only incidentally in Spanish, its topic common to the international audience?

- When editing labels for a multilingual taxonomy, you may be tempted to offer translations of the labels in all represented languages to allow access across the languages. Remember, though, that affinity generation pulls these labels into the profiles where long strings of text may become unwieldy.

Example of a taxonomy containing multiple languages

Here is a suggested procedure for creating a taxonomy that contains documents in multiple languages, taking for example, a data set that contains mostly English, but some German documents.

1. Create an English training set—a selected subset of your data set that contains the 1% of the documents, written in the common business language, that best represents your business goal and the mental model you want to present.
2. Spider the training set to create an initial taxonomy. Edit this initial English taxonomy until you are satisfied with the category labels and the distribution of documents in the categories.
3. Enable classification and spider a larger subset of your data—the next best 10%. In our example, these repositories contain German documents.
4. Edit the resulting taxonomy, pulling the German documents into the English categories if they relate to them, and keeping them in separate subcategories if they do not.
5. Add more data repositories from your entire data set. Documents from the entire data set will enter existing categories, with the German documents already in the categories attracting the new documents in German, and English documents attracting English documents with similar content. When a substantial number of new documents appear, check to make sure you are satisfied with the categories and the distribution of documents in them.

Conclusion

In your multilingual, multinational environment, you need a tool to help you organize and find your business' e-knowledge faster. Discovery Server is that tool. You can organize your e-knowledge and enable your employees to find the people, places and things that they need to do their jobs more efficiently no matter where on the planet they work or which languages they speak. Discovery Server speaks the language of knowledge.

ABOUT KAREN BJORKMAN

Karen Bjorkman has been involved with taxonomies for a long time. Currently an Information Developer for Notes and Domino User Assistance, she performed her first index on a collection of *Nature and Science* magazines at age 11. She joined Lotus Development Corporation in 1987 and became an IBM employee in 1999. Past projects include writing and editing for 1-2-3, SmartSuite, Notes, and Discovery Server. She has a Bachelor of Arts in writing from Barnard College and an Master of Arts in literature from Boston University. Her hobbies include thinking about playing bass, hiking, and the relationship between Gnosticism and holistic science. Previous lives have included theater, teaching, journalism, and marriage. She lives with Hannah, 8, Sarah, 6, and Grace the cat in Arlington, MA.

ABOUT HIROYUKI (HIRO) MIYAMOTO

Hiro Miyamoto is the lead Globalization engineer for Discovery Server. In his many years at Lotus, he has worked on a variety of

Lotus products, including Approach, 1-2-3, eSuite, and K-station. He began his career as a software programmer for Digital Equipment in Tokyo where he worked on X-Window standards and implementation for Digital's platforms. Hiro has a Bachelor's degree in Electrical Engineering from Waseda University in Tokyo, Japan. He lives outside of Boston and in his spare time enjoys exploring his adopted country. However, Hiro's spare time is limited since his second child has recently started walking and climbing everywhere.

ABOUT ROBIN ROUSE

Robin works in Global Product Development in the Lotus software division of IBM. One of her previous gigs at Lotus included co-writing the Domino Designer 5.0 Application Developer's Guide. She started life at Lotus in 1994, as a pre-kindergarten teacher in the Lotus Children's Center, but in 1997, she traded kids for computers because computers don't throw up on you! Now she is living happily ever after. Robin has a Bachelor of Liberal Arts from Harvard University and a Master's of Science in Education from Wheelock College. She has a daughter Sara, who is in high school, and a son Ryan, who is a student at the University of California, Santa Cruz. Robin's husband, Brian Gallagher, is a Quality Engineer on the Mobile Notes team. You can catch Robin at 6 AM every morning at puppy playgroup with her lovable mutt Hazel.

