



**Level:** Beginner  
**Works with:** Extended Search  
**Updated:** 01-May-2002

## Introduction to Lotus Extended Search

by  
Lauren  
Wendel

Do your users need to search Domino databases in multiple Domino domains, Web and content sources, data indexed to a Microsoft Index Server system, and enterprise data from a RDBMS databases? If so, then you should get to know IBM Lotus Extended Search.

IBM Lotus Extended Search is a scalable, server-based technology that searches across many parallel content and data sources and returns query results to a Notes or Web application. Previously known as Domino Extended Search, the product was initially released in 1997 to extend the search capability of Domino applications to multiple back end data sources, such as DB2, Oracle, and Microsoft SQL Server databases. Since then, we've added many new features as well as support for Web application servers and more operating systems.

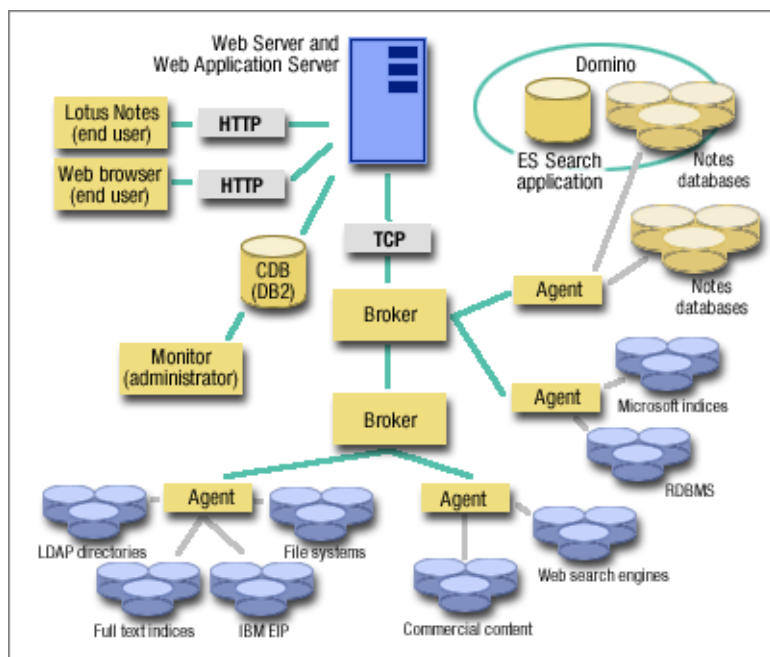
In this article, we'll present the technical architecture and components of Extended Search, give an overview of the latest feature additions in Release 3.5, and preview the upcoming IBM Lotus Extended Search Release 3.7 available mid-year 2002.

### Extended Search Overview

Extended Search uses a system of interlocking components for managing all aspects of client search requests including:

- Query string to source language interpretation
- Data security verification
- Number of search results desired
- Results ranking
- Server scalability and load balancing to support a growing number of users in organizations that require search capabilities

The following diagram shows how client-side and server-side components relate to one another in the Extended Search architecture. In the upper-left corner, you see the two components of user search access: a Web browser and a Lotus Notes client. The remaining components—Brokers, Agents, and so on—constitute the server components of the Extended Search system. We'll look at each component in detail to help you understand how Extended Search works.



### User search "Client-side" components

You can access Extended Search from a Notes or Web application. To help you get started, Extended Search provides the following sample query applications that you can customize.

- The Web application contains enhanced Hypertext Markup Language (HTML) files with expanded search functionality provided by Extended Search. The sample Web application allows users to search across multiple data sources simultaneously using a Web browser. When you perform a search with a Web browser, Extended Search channels search requests and options, as specified in the HTML file, through an Extended Search server-side Servlet described in the next section.
- The Notes application shows the same set of Extended Search query options that are available in the Web application sample. With the Notes client, users submit queries, view results, and fetch documents through their Notes client. When you perform a search with a Notes application, Extended Search invokes a set of Extended Search LSXs to execute search requests and options on the client. Using the Extended Search Notes sample application, user installation of the LSXs occurs automatically the first time the user accesses the application. The server can save search results in the same Notes application, so the results can be re-used or shared with other users.

Most organizations customize these sample applications or extend their existing applications to include federated search capabilities. For example, Extended Search supplies a set of custom search tags that you can embed in HTML files. Users can enter query strings, identify various sources to be searched, and set a variety of search and retrieval options. The server saves search results in an HTML file in locations identified by Extended Search replacement variables. The Extended Search tags, which you can embed anywhere within an HTML file, do not interfere with surrounding HTML tags. So a Web designer can control nearly all presentational aspects and the subsequent look and feel of the search application.

### Server-side components

Several server-side components work together to execute a search request from a Notes or Web application.

#### **Reflector and Servlet**

The Reflector and Servlet dispatch search requests on behalf of the Extended Search LSXs or on behalf of a Web browser to the Broker. The Reflector and Servlet exist on the machine that hosts your Web application server. Extended Search supports several Web servers, including Lotus Domino, IBM Websphere, Microsoft IIS, and IBM HTTP (Apache).

Extended Search uses the Hypertext Transfer Protocol (HTTP) to invoke the Reflector and the Servlet. The Reflector is a Common Gateway Interface (CGI) program used to service Notes client requests, and the Servlet is a Java program used to service Web browser requests. This two-tiered approach allows the client application to

use an industry-standard protocol (HTTP), so the client can use many Web server-related features like socks, proxies, and Secure Sockets Layer (SSL) technology.

The Reflector and Servlet can communicate with an Extended Search server that resides on a machine other than the Web server. This provides added flexibility when planning resource capacity and performance because it's possible to dispatch search requests from primary to secondary Brokers running on different servers.

### **Broker**

The Broker is a central part of the Extended Search server. It receives search requests and acts as a resource coordinator to dispatch the query and return results to the calling client application. To support performance and scalability, a given Extended Search domain can contain multiple Brokers. Each Broker has its own network address and typically performs the following tasks:

- Validates the user
- Obtains a list of the data sources available to the user defined within the search application
- Submits queries to target data sources managed by Extended Search Agents and Links
- Aggregates and sorts search results to a single set of results, or hitlist, according to the definitions of the search application
- Retrieves source documents (note that for Web-based searches, the Web browser uses the URL returned in the hitlist to retrieve the document)

### **Configuration database**

The Broker obtains information about the resources it manages from the Extended Search configuration database (CDB). The CDB, managed by IBM DB2 Universal Database, contains information about data sources that have been defined for search. It stores network addresses, field-level usage controls, and the names of the application search programs used to access, search, and fetch documents from each data source. The CDB may also include specific configuration options per data source. For example, it can identify specific security DLLs to invoke for data source search authentication, or "user exit."

You can manage the CDB using the Administration applet to modify Extended Search components. The Administration applet includes search data source Discoverers. These are programs that can query a source server for available applications to search and that can automate the process of loading the CDB with default information about new data sources available on specific servers.

### **Operations and performance console and Monitor**

When you start an Extended Search and a Broker is launched, Extended Search displays a scrollable console window in the Administrator applet. This console window initially contains startup messages and ends with a command prompt. Other than infrequent messages that are displayed in the text window, the Extended Search server runs in an uninformative, quiet mode, processing messages and service requests.

The Extended Search Monitor monitors server activities. After you invoke the Monitor for each Extended Search server, you can observe server activity from the Administration applet. With the Monitor, you can adjust and refresh server configurations directly without having to stop and restart the individual server or Extended Search domain, which contains servers, Brokers, Agents, and Links communicating across the network to service aspects of search operations.

The Monitor can run independently of the Broker. You can start and stop the Monitor any number of times without affecting the processing of the Extended Search server. More than one Monitor can connect to the same Extended Search server simultaneously. You can run the Monitor remotely to check the status of the Extended Search server from a location other than the host machine console.

### **Agents**

Extended Search Agents are programs that respond to search and retrieval operations targeted toward a particular data source. Agents load the appropriate source Link and translator modules from query string to source syntax when a request for a specific data source type is made. The Agents then call upon these module libraries for query translation and for connect, disconnect, search, and retrieval operations.

Agents sort the results by relevance rank, then truncate the results to the maximum number of hits specified in the original search request application. We recommend that Agents reside on the same machine as the data source, so the Agents can service search requests faster. Agents can also use a data source's remote APIs for access. More than one copy of an Agent can run on a single server to handle concurrent search and retrieval requests. You can dedicate an Agent to a single data source, to a group of sources of a particular type, or to a range of sources that have a mixture of Link requirements.

### **Data source search Links**

An Extended Search Link translates a search and retrieval request sent by a Broker into the native calls of each data source's programming interface. The Link then executes the translated request against each data source. Extended Search applications translate the search request string into an internal representation called Generalized Query Language (GQL). This makes the individual query languages transparent to the user and enables Extended Search to search multiple and varied data sources in parallel.

Here's an example of how Links work: an Agent invokes an already activated Notes Link when it receives a GQL string search request for information in a Notes database. The Notes Link translates the query from the Extended Search Generalized Query Language (GQL) into the Notes native syntax. Then the Link issues the appropriate Notes Object Interface (NOI) calls to the Notes C/C++ API (FTSearch) and gets the results.

Each source Link stores its search results in a hitlist. It sends the hitlist back to the Agent to aggregate the results according to the desired maximum, to sort by relevance, and to return the results to the Broker for further processing. The Broker aggregates and sorts the results with hitlists that are returned by other Agents and returns a single hitlist to the requesting user through the search application.

### **Extended Search Link types**

Users can search and retrieve documents from repositories that include Lotus Notes 4.x and 5.x, Domino.Doc, Lotus Connectors, and Domino R5 Domain Index. They can also search 18 popular News sites and Web search engines like [HotBot](#) and [Excite](#). Extended Search also searches file systems (including uncompressed file attachments), LDAP Directories, and other sources like output from Microsoft Index Server and Microsoft Site Server.

An e-mail content search Link searches Domino and Microsoft Exchange mail servers. You can also search for information from ODBC databases and IBM Enterprise Information Portal content and data sources.

Extended Search includes a Web crawler that traverses Web sites on a scheduled basis and outputs results in XML or HTML format. Lotus Business Partner Atlantic Decisions provides a Verity search Link.

### **Extended Search Link Toolkit**

You can create your own Links with the Extended Search Link Toolkit. To create Web site Links, you can use Intelligent Algorithms infoGIST Toolkit to create a custom Web source definition file that searches the Web. Refer to the Extended Search documentation for more information about these toolkits.

## **Parallel, direct searches of source content**

Extended Search does not require a centralized index. Instead, it accesses information directly using search and retrieval methods native to each data source. This design allows Extended Search to provide a distributed, heterogeneous search across many different data repositories through a single, efficient, and easy-to-use point of access.

The approach has several advantages:

- Search results always provide current information obtained directly from the data source (unless the Web crawler has been used to extract and store source content for search on a scheduled basis)
- Data remains within its original source database
- Overall storage requirements are reduced and the cost of re-indexing resources are eliminated because duplication of the data into a centralized index isn't necessary

At the same time, there are advantages to indexing content especially in situations where a large volume of content will be aggregated for search. Extended Search supports a growing number of indexing technologies to leverage indexed data. Many Extended Search customers today search across indexed and non-indexed content and data sources from their Extended Search-enabled applications.

## **Managing security across content sources**

We designed the Extended Search architecture to provide layers of security that protect source content at several levels:

### **Web server user authentication**

Extended Search uses the Web server environment to verify user access to the search applications. The Web server can use private and public key encryption, digital certificates, and passwords to authenticate users and to ensure that only authorized users can access search applications.

### **Search application level authentication**

At the application level, you can specify access controls for individual search applications with the Administration applet. Not only can you restrict access to data sources, but you also have complete control over which fields a user can search and which results he can view or retrieve. You can even set a "user exit" for a particular search application to invoke a specific security verification routine.

For example, a user exit can communicate with a secure database to confirm the user's identity, and the application can invoke this user exit for each data source searched. A user exit can also verify the user individually or as a member of a defined group or return a different user ID to be used within the Extended Search domain and search application. This feature can prove useful because not all IDs are appropriate for all sources or for all content in a specific source.

### **Broker level authentication**

You can define custom user exits at the Broker level to selectively deny or approve access to individual data sources named in the search request.

### **Agent level authentication**

You can also define custom user exits at the Agent level. Optionally, Extended Search can invoke additional security rules during Agent processing. For example, an Agent level user exit might further validate a user identity prior to processing the search request. Before returning results, the exit might filter the content and any defined constraint values and, if necessary, delete items that fail to pass this post-processing inspection.

### **Link level authentication**

Extended Search works with the individual data sources, each of which may have their own security mechanisms to permit or deny access to content. You can set security for user exit verification at the Link level of search processing.

## **New features of the current release**

The currently available Extended Search Release 3.5 introduced many new search options, performance improvements, and platform support, including:

- Operating system support for Microsoft Windows NT and 2000, IBM AIX, and Sun Solaris 7 and 8
- JSP search examples of customizable Web applications
- EJB enablement to communicate system processing requests with a configuration application
- Integration with Lotus K-station and Websphere Portal that includes predefined search portlets to Extended Search sources
- Support for XML search results
- Saved queries for shared use and re-application
- Scheduled queries that run at specific times and that store search results
- National Language Support (NLS-enablement) to search content and to develop applications in any language
- Enhanced troubleshooting that includes improved error reporting and additional logging controls
- Web crawling to retrieve Web pages from selected Web sites for subsequent indexing or searching
- Performance enhancements

## **Features coming soon**

Extended Search Release 3.7 will be available mid-year 2002 from Lotus and IBM Websphere Portal 4.1. This upcoming release will include useful new features and enhancements such as:

- Search data source configuration wizards
- Support for IBM DB2 7.2 as the configuration data store
- Internationalized versions of documentation and product interfaces
- Native Microsoft SQL Server and Microsoft Access search links

In addition, the Extended Search team is currently working toward Release 4.0 of Extended Search, which will include a number of key search sources for IBM Websphere Portal - Juru Index, Lotus Discovery Server, and Lotus QuickPlace as well as search integration with Lotus Sametime. We have a number of search option and administration enhancements planned and are always interested in your feedback and desire to participate in upcoming pre-release beta programs. Please communicate with us through our newly launched [Extended Search forum](#) on the [Lotus Developer Domain](#). We also encourage you to review many existing product information white papers and FAQs on the [Extended Search](#) page of the Lotus Web site.

### **ABOUT THE AUTHOR**

Lauren Wendel is product manager for Lotus Extended Search, Lotus Discovery Server, and expertise technologies. Previously,

Lauren worked with the Lotus Enterprise Integration team for five years, overseeing the initial releases of Lotus Enterprise Integrator, DECS, ERP Connectors, and the Connector API Toolkit. She has also worked as a developer consultant within the Lotus Business Partner program, and previously, within the 1-2-3 engineering team. Lauren's also managed systems planning at Wells Fargo Bank, Citibank, Duke University School for Executive Education, and Grant Thornton Ltd. She enjoys running the "occasional" marathon and sings with a community chorus.