IDOL

IDOL™ Getting Started Guide

Version 7.6
Document Revision 0
09 February 2012

Power

Autonomy
an HP company

Notice

This documentation is a proprietary product of Autonomy and is protected by copyright laws and international treaty. Information in this documentation is subject to change without notice and does not represent a commitment on the part of Autonomy. While reasonable efforts have been made to ensure the accuracy of the information contained herein, Autonomy assumes no liability for errors or omissions. No liability is assumed for direct, incidental, or consequential damages resulting from the use of the information contained in this documentation.

The copyrighted software that accompanies this documentation is licensed to the End User for use only in strict accordance with the End User License Agreement, which the Licensee should read carefully before commencing use of the software. No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner.

This documentation may use fictitious names for purposes of demonstration; references to actual persons, companies, or organizations are strictly coincidental.

Trademarks and Copyrights

 ACI API, Alfresco Connector, Arcpliance, Autonomy Process Automation, Autonomy Fetch for Siebel eBusiness Applications, Autonomy, Business Objects Connector, Cognos Connector, Confluence Connector, ControlPoint, DAH, Digital Safe Connector, DIH, DiSH, DLH, Documentum Connector, DOH, EAS Connector, Ektron Connector, Enterprise AWE, eRoom Connector, Exchange Connector, FatWire Connector, File System Connector for Netware, File System Connector, FileNet Connector, FileNet P8 Connector, FTP Fetch, HTTP Connector, Hummingbird DM Connector, IAS, IBM Content Manager Connector, IBM Seedlist Connector, IBM Workplace Fetch, IDOL Server, IDOL, IDOLme, iManage Fetch, IMAP Connector, Import Module, iPlanet Connector, KeyView, KVS Connector, Legato Connector, LiquidOffice, LiquidPDF, LiveLink Web Content Management Connector, MCMS Connector, MediClaim, Meridio Connector, Meridio, Moreover Fetch, NNTP Connector, Notes Connector, Objective Connector, OCS Connector, ODBC Connector, Omni Fetch SDK, Open Text Connector, Oracle Connector, PCDocs Fetch, PLC Connector, POP3 Fetch, Portal-in-a-Box, RecoFlex, Retina, SAP Fetch, Schlumberger Fetch, SharePoint 2003 Connector, SharePoint 2007 Connector, SharePoint 2010 Connector, SharePoint Fetch, SpeechPlugin, Stellent Fetch, TeleForm, Tri-CR, Ultraseek, Verity Profiler, Verity, VersiForm, WebDAV Connector, WorkSite Connector, and all related titles and logos are trademarks of Autonomy Corporation plc and its affiliates, which may be registered in certain jurisdictions.

Microsoft is a registered trademark, and MS-DOS, Windows, Windows 95, Windows NT, SharePoint, and other Microsoft products referenced herein are trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group.

AvantGo is a trademark of AvantGo, Inc.

Epicentric Foundation Server is a trademark of Epicentric, Inc.

Documentum and eRoom are trademarks of Documentum, a division of EMC Corp.

FileNet is a trademark of FileNet Corporation.

Lotus Notes is a trademark of Lotus Development Corporation.

mySAP Enterprise Portal is a trademark of SAP AG.

Oracle is a trademark of Oracle Corporation.

Adobe is a trademark of Adobe Systems Incorporated.

Novell is a trademark of Novell, Inc.

Stellent is a trademark of Stellent, Inc.

All other trademarks are the property of their respective owners.

Notice to Government End Users

If this product is acquired under the terms of a **DoD contract:** Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of 252.227-7013. **Civilian agency contract:** Use, reproduction or disclosure is subject to 52.227-19 (a) through (d) and restrictions set forth in the accompanying end user agreement. Unpublished-rights reserved under the copyright laws of the United States. Autonomy, Inc., One Market Plaza, Spear Tower, Suite 1900, San Francisco, CA. 94105, US.

09 February 2012

Contents

Part 1 IDOL Systems

Part 2 Install and Run IDOL Server

Figures

About This Document

This document is for IDOL server users and administrators. It is intended for readers who are responsible for setting up, using and maintaining an IDOL server installation and who are familiar with enterprise software administration.

Documentation Updates

The information in this document is current as of IDOL version 7.6. The content was last modified 09 February 2012.

You can retrieve the most current product documentation from Autonomy's Knowledge Base on the Customer Support Site.

A document in the Knowledge Base displays a *version number* in its name, such as *IDOL Server 7.5 Administration Guide*. The version number applies to the product that the document describes. The document may also have a *revision number* in its name, such as *IDOL Server 7.5 Administration Guide Revision 6*. The revision number applies to the document and indicates that there were revisions to the document since its original release.

It is recommended that you periodically check the Knowledge Base for revisions to documents for the products your enterprise is using.

To access Autonomy documentation

1. Go to the Autonomy Customer Support site at

   https://customers.autonomy.com

2. Click **Login**.
3. Enter the login credentials that were given to you, and then click **Submit**.

   The Knowledge Base Search page opens.

4. In the **Search** box, type a search term or phrase. To browse the Knowledge Base using a navigation tree only, leave the **Search** box empty.
5. Ensure the **Documentation** check box is selected.
6. Click **Search**.

   Documents that match the query display in a results list.

7. To refine the results list, select one or more of the categories in the **Filter By** pane. You can restrict results by
   - **Product Group**. Filters the list by product suite or division. For example, you could retrieve documents related to the iManage, IDOL, Virage or KeyView product suites.
   - **Product**. Filters the list by product. For example, you could retrieve documents related to IDOL Server, Virage Videologger, or KeyView Filter.
   - **Component**. Filters the list by a product's components. For example, you could retrieve documents related to the Content or Category component in IDOL.
   - **Version**. Filters the list by product or component version number.
   - **Type**. Filters the list by document format. For example, you could retrieve documents in PDF or HTML format. Guides are typically provided in both PDF and HTML format.
8. To open a document, click its title in the results list.

   To download a PDF version of a guide, open the PDF version, click the Save icon  in the PDF reader, and save the PDF to another location.

Related Documentation

The following documents provide more details on IDOL:

Document	Description
IDOL Server Administration Guide	Describes how to configure and use IDOL server.
Distributed Action Handler Administration Guide	Describes how to configure and use the Distributed Action Handler (DAH) with IDOL server.
Distributed Indexing Handler Administration Guide	Describes how to configure and use the Distributed Index Handler (DIH) with IDOL server.
Distributed Service Handler Administration Guide	Describes how to configure and use the Distributed Service Handler (DiSH) to manage IDOL services and licences.
Distributed Load Handler Administration Guide	Describes how to configure and use the Distributed Load Handler (DLH) to manage the load on different IDOL servers.
Intellectual Asset Protection System Administration Guide	Describes how to set up and use security for IDOL server.
IDOL Eduction User Guide	Describes how to set up and use eduction with IDOL server to extract entities such as names, statistics, and locations from unstructured content.
Autonomy Collaborative Classifier User Guide	Describes how to set up and use the Autonomy Collaborative Classifier (ACC) to manipulate information-classification structures and indexes.
Autonomy Business Console User Guide	Describes how to set up and use the Autonomy Business Console (ABC) to manage the content of data indexed in IDOL servers.
IDOL Administration User Guide	Describes how to use the IDOL Administration interface to set up and administer your IDOL server installation.

Document	Description
File System Connector Administration Guide *HTTP Connector Administration Guide* Other connector guides, as needed.	Describes how to configure and use Autonomy connectors to aggregate data from different sources to index into IDOL server.
Portal-in-a-Box Administration Guide *Portlets User Guide*	Describes how to set up and use the Portal-in-a-Box user interface, and how to add and manage the portlets.
Retina Administration and User Guide	Describes how to set up and use the Retina user interface to search and manage IDOL server data.

Conventions

The following conventions are used in this document.

Notational Conventions

This document uses the following conventions.

Convention	Usage
Bold	User-interface elements such as a menu item or button. For example: Click **Cancel** to halt the operation.
Italics	Document titles and new terms. For example: ■ For more information, see the *IDOL Server Administration Guide*. ■ An *action command* is a request, such as a query or indexing instruction, sent to IDOL Server.

Convention	Usage
`monospace font`	File names, paths, and code. For example: The `FileSystemConnector.cfg` file is installed in `C:\Program Files\FileSystemConnector\`.
`monospace bold`	Data typed by the user. For example: ■ Type **`run`** at the command prompt. ■ In the **User Name** field, type **`Admin`**.
`monospace italics`	Replaceable strings in file paths and code. For example: `user` *`UserName`*

Command-line Syntax Conventions

This document uses the following command-line syntax conventions.

Convention	Usage
`[ optional ]`	Brackets describe optional syntax. For example: `[ -create ]`
`|`	Bars indicate “either | or” choices. For example: `[ option1 ] | [ option2 ]` In this example, you must choose between `option1` and `option2`.
`{ required }`	Braces describe required syntax in which you have a choice and that at least one choice is required. For example: `{ [ option1 ] [ option2 ] }` In this example, you must choose `option1`, `option2`, or both options.

Convention	Usage
`required`	Absence of braces or brackets indicates required syntax in which there is no choice; you must type the required syntax element.
`variable` `<variable>`	Italics specify items to be replaced by actual values. For example: `-merge` *`filename1`* (In some documents, angle brackets are used to denote these items.)
`...`	Ellipses indicate repetition of the same pattern. For example: `-merge` *`filename1`*`,` *`filename2`* `[,` *`filename3`* `... ]` where the ellipses specify, *`filename4`*, and so on.

The use of punctuation—such as single and double quotes, commas, periods—indicates actual syntax; it is not part of the syntax definition.

Notices

This document uses the following notices:

CAUTION A caution indicates an action can result in the loss of data.

IMPORTANT An important note provides information that is essential to completing a task.

NOTE A note provides information that emphasizes or supplements important points of the main text. A note supplies information that may apply only in special cases—for example, memory limitations, equipment configurations, or details that apply to specific versions of the software.

TIP A tip provides additional information that makes a task easier or more productive.

Autonomy Product References

This document references the following Autonomy products:

- Autonomy Intelligent Data-Operating Layer (IDOL) Server
- Distributed Action Handler (DAH)
- Distributed Index Handler (DIH)
- Distributed Service Handler (DiSH)
- Distributed Load Handler (DLH)
- Autonomy Collaborative Classifier (ACC)
- Autonomy Business Console (ABC)
- Autonomy IDOL with Administration
- Query Manipulation Server (QMS)
- Intellectual Asset Protection System (IAS)
- IDOL Eduction
- File System Connector
- HTTP Connector
- Notes Connector
- Autonomy KeyView
- Portal in a Box
- Retina
- Autonomy Content Infrastructure API (ACI API)

Autonomy Customer Support

Autonomy Customer Support provides prompt and accurate support to help you quickly and effectively resolve any issue you may encounter while using Autonomy products. Support services include access to the Customer Support Site (CSS) for online answers, expertise-based service by Autonomy support engineers, and software maintenance to ensure you have the most up-to-date technology.

To access the Customer Support Site, go to

https://customers.autonomy.com

The Customer Support Site includes:

- **Knowledge Base**: The CSS contains an extensive library of end user documentation, FAQs, and technical articles that is easy to navigate and search.
- **Case Center**: The Case Center is a central location to create, monitor, and manage all your cases that are open with technical support.
- **Download Center**: Products and product updates can be downloaded and requested from the Download Center.
- **Resource Center**: Other helpful resources appropriate for your product.

To contact Autonomy Customer Support by e-mail or phone, go to

http://www.autonomy.com/content/Services/Support/index.en.html

Contact Autonomy

For general information about Autonomy, contact one of the following locations:

Europe and Worldwide	North and South America
E-mail: autonomy@autonomy.com	E-mail: autonomy@autonomy.com
Telephone: +44 (0) 1223 448 000 Fax: +44 (0) 1223 448 001	Telephone: 1 415 243 9955 Fax: 1 415 243 9984
Autonomy Corporation plc Cambridge Business Park Cowley Rd Cambridge CB4 0WZ United Kingdom	Autonomy, Inc. One Market Plaza Spear Tower, Suite 1900 San Francisco CA 94105 USA

Document Revision History

Document Revision	Release Date	Notes
7.6 revision 0	09 February 2012	Updated installation instructions with new screenshots for 7.6.0.0 installers.
7.5 revision 3	21 July 2011	Added new chapters: ■ Configure IDOL. ■ Security in IDOL ■ Applications ■ IDOL Performance ■ Tutorial: Index data into IDOL
7.5 revision 2	02 June 2011	Added installation instructions for UNIX console mode.
7.5 revision 1	20 September 2009	Added minor updates to IDOL installer instructions.

PART 1 IDOL Systems

This section describes IDOL server and describes the various set ups that you can use in your IDOL system.

- Introduction to IDOL
- Types of IDOL Systems
- Applications
- Security in IDOL

CHAPTER 1

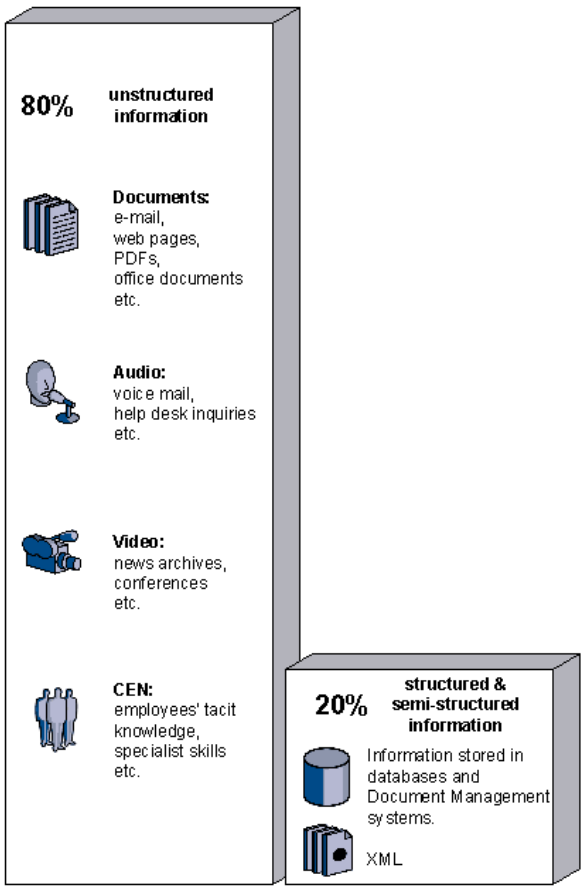
Introduction to IDOL

Autonomy Intelligent Data Operating Layer (IDOL) server integrates unstructured, semi-structured, and structured information from multiple repositories through an understanding of the content. It delivers a real time environment to automate operations across applications and content, removing all the manual processes involved in getting information to the right people at the right time.

Autonomy Infrastructure

Information that you need to conduct business successfully includes the following types.

Figure 1 Types of business information



More than 80% of all data in an enterprise is unstructured information. Unfortunately, attempts to leverage this immense and strategic resource often fail because many businesses lack the requisite technology to understand and effectively utilize content that resides outside the scope of structured databases. Similarly, unstructured processes are equally unwieldy yet comprise the bulk of business operations. While conventional business process management tools can tackle well-defined processes, they predictably falter under the weight of byzantine business rules and the complex logic required by dynamic processes prevalent in a multitude of business applications.

Autonomy's ability to process all forms of digital information on a single platform offers a unique solution to a growing number of applications and devices that are increasingly dependent on utilizing unstructured information. Autonomy employs a unique combination of technologies involving Bayesian Inference, Shannon's Information Theory, and over 140 patents, to enable computers to form a contextual understanding of all digital content, as well as understand people's interaction with the data. Autonomy's technology eliminates the traditionally manual and costly operation of processing and analyzing information by performing these functions automatically and in real-time. This represents substantial savings for every type of organization and industry, as evidenced by the significant penetration of Autonomy's technology in a diversity of vertical markets.

Autonomy's unique combination of technologies provides:

- Accuracy
- Speed and performance
- Scalability
- Security
- Language Independence
- Easy integration
- Support for any content format
- Support for future technologies

Autonomy is therefore able to power any application dependent upon unstructured and structured information, including:

- Pan-Enterprise Search
- Regulatory and Legal Content Management
- Customer Interaction Solutions
- eCommerce
- Web Solutions
- Security and Surveillance
- Digital Asset Management
- Business Process Management
- Information Governance
- eDiscovery

Key IDOL Components

The following section describes the core components that IDOL systems use.

IDOL Server

Autonomy's *Intelligent Data Operating Layer* (IDOL) gathers and processes unstructured, semi-structured, and structured information in any format from multiple repositories using IDOL connectors and a global relational index. It can automatically form a contextual understanding of the information in real time, linking disparate data sources together based on the concepts contained within them. For example, IDOL can automatically link concepts contained in an e-mail message to a recorded phone conversation, which can be associated with a stock trade. This information is then imported into a format that is easily searchable, adding advanced retrieval, collaboration, and personalization to an application that integrates the technology.

Administration

IDOL Administration provides a distributed, Web-based infrastructure for managing IDOL components and services.

Refer to the *IDOL Administration User Guide* for more information.

NOTE If you are using the Standalone version of IDOL, the administrative interface is not provided, and you must manage components and services by editing the configuration files directly.

Connectors

Connectors enable automatic content aggregation from any type of local or remote repository (for example, a database, a Web site, a real-time telephone conversation and so on). Connectors form a unified solution across all information assets within the organization.

Security

Autonomy provides the software infrastructure that automates operations on unstructured information. This software infrastructure is based on IDOL server.

IDOL server makes it possible for organizations to process digital content automatically and allows applications to communicate with each other. It consists of data operations that integrate information by understanding content, and is therefore data agnostic.

The Intellectual Asset Protection System (IAS) provides an integrated security solution to protect your data:

- **Front end security**. At the front end, authentication checks users are allowed to access the system on which result data is displayed.
- **Back end security**. At the back end, entitlement checking and authentication combine to ensure query results include only documents the user is allowed to view from repositories the user is allowed to access.
- **Secure communications**. You can encrypt communications between ACI servers and any applications using the Autonomy ACI API. You can also configure IDOL Server and its other components to use Secure Socket Layer (SSL) communications.

Related Topics

- "Security in IDOL" on page 61

Interfaces

- **Portlets** are windows that can be set up in Autonomy's Portal-in-a-Box or third party portals. Each portlet contains an application that allows the portals' end users to benefit from a variety of IDOL server functionality.
- **Retina™** is an easy-to-use Web interface application that provides a full scale of retrieval methods that adjust to user proficiency.
- **Autonomy Desktop Suite™** brings the power of Autonomy to every desktop. Conducting a real-time analysis of the ideas involved in the content of any opened desktop application, the Desktop Suite ActiveKnowledge or Active Windows Extensions module provides real-time links to relevant internal and external information without users being diverted from their work in progress to perform a search or retrieval operation.
- **Autonomy Collaborative Classifier (ACC)** manipulates information-classification structures and indexes.
- **Autonomy Business Console (ABC)** allows you to manage the content of data indexed in IDOL servers. By creating and modifying different business projects, you can predefine which results are returned to users, and how the user views them. You can also view statistical information to help you refine your business projects and make them more effective.
- **ACI API** uses HTTP to allow custom-built applications (for example C, PHP, TCL, Java, Com or Com+ applications) to communicate with Autonomy ACI servers.

Related Topics

- “Applications” on page 57

Distributed systems

Autonomy distribution solutions facilitate linear scaling of systems through faster action execution and reduction of processing time.

- **DAH**™ (Distributed Action Handler) enables the distribution of ACI (Autonomy Content Infrastructure) actions to multiple Autonomy IDOL servers, providing failover and load balancing.
- **DIH**™ (Distributed Index Handler) enables distributed indexing of documents into multiple Autonomy IDOL servers, providing failover and load balancing.

IDOL Server Operations

Autonomy IDOL can perform the following intelligent operations across structured, semistructured, and unstructured data.

- Agents
- Automatic Query Guidance
- Channels
- Collaboration
- Dynamic Thesaurus
- Expertise
- Mailing
- Retrieval
- Summarization
- Viewing
- Alerts
- Categorization
- Cluster Data
- Dynamic Clusters
- Eduction
- Hyperlinks
- Profiles
- Spelling Correction
- Taxonomy Generation

NOTE Your license determines which of these operations your IDOL installation can perform.

Agents

Users can create agents in IDOL to find and monitor information that is relevant to their interests. The agents collect this information from a configurable list of Internet and intranet sites, news feeds, chat streams, and internal repositories.

Alerts

IDOL analyzes data when it receives new documents and compares the concepts in documents with user agents. If new data matches a user agent it immediately notifies the user by e-mail or a third-party system (for example by SMS or a pager).

Automatic Query Guidance

IDOL finds the most salient terms and phrases in query results and automatically clusters these terms and phrases. It uses the clustered phrases to provide a hierarchical set of queries that guide users to the result area they are looking for.

Categorization

IDOL can automatically categorize data. The flexibility of Autonomy Categorization allows you to precisely derive categories using concepts found within unstructured text. This process ensures that IDOL server classifies all data in the correct context with the utmost accuracy. Autonomy Categorization is a completely scalable solution capable of handling high volumes of information with extreme accuracy and total consistency.

Category Matching

IDOL accepts a category or piece of content and returns categories ranked by conceptual similarity. This ranking determines the most appropriate categories for the piece of content, so that IDOL can subsequently tag, route, or file the content accordingly.

Channels

IDOL can automatically provide users with a set of hierarchical channels with highly relevant information pertinent to the respective channel. Channels are similar to agents, aggregating information that is relevant to the channel concept. Usually, administrators set up channels that are available to all users.

Cluster Information

IDOL automatically clusters information. Clustering takes a large repository of unstructured data, agents, or profiles and automatically partitions the data to cluster similar information together. Each cluster represents a concept area within the knowledge base and contains a set of items with common properties.

Collaboration

IDOL automatically matches users with common explicit interest agents or similar implicit profiles. This information creates virtual expert knowledge groups.

Dynamic Clusters

When it executes queries, IDOL automatically clusters the query results, and then in turn clusters the first set of clusters further to produce subclusters. This process allows you to generate a hierarchy of clusters that allows users to navigate quickly to their area of interest.

Dynamic Thesaurus

When it executes queries, IDOL can automatically suggests alternative queries, allowing users to quickly produce a variety of relevant result sets.

Eduction

Eduction is a tool that you can use to extract an *entity* (a word, phrase, or block of information) from text, based on a pattern you define. The pattern can be a dictionary of names such as people or places. The pattern can also describe what the sequence of text looks like without listing it explicitly, for example, a telephone number. The entities are contained inside grammar files.

When you use Eduction with IDOL, Eduction extracts the entities as the document is indexed and adds them into fields for easy retrieval.

The Eduction capability of IDOL is described in the *Eduction User Guide*.

Expertise

IDOL accepts a natural language or Boolean search string and returns users who own matching agents or profiles. This process allows instant identification of experts in a subject, eliminating time-consuming searches for specialists, and unnecessary researching of subjects for which expert knowledge is already available.

Hyperlinks

You can automatically generate hyperlinks in real time. These link to contextually similar content, for example to recommend related articles, documents, affinity products or services, or media content that relates to textual content.

IDOL server automatically inserts these links when it retrieves the document. This process means that new documents can reference older documents, and that archived documents can link to the latest news or material on the subject.

E-Mail Users

IDOL matches the agents and profiles against its document content in regular intervals, and automatically notifies users of documents that match their agents or profiles by sending them e-mail.

Profiles

IDOL automatically creates interest and expertise profiles for users, in real time.

You can create interest profiles by tracking the content that a user views and extracting a conceptual understanding of it. IDOL then uses this understanding to keep user interest profiles up to date. You can use interest profiles to:

- target information on users.
- recommend content to users.
- alert users to the existence of content.
- put users in touch with other users who have similar interests.

You can create expertise profiles by tracking the content that a user creates and extracting a conceptual understanding of it. IDOL uses this understanding to keep user expertise profiles up to date. You can use expertise profiles to trace users who are experts in particular subject areas.

Search and Retrieval

IDOL offers a range of retrieval methods, from simple legacy keyword search to sophisticated conceptual querying.

Examples of the kind of query that is possible with IDOL server include:

- Conceptual Matching
- Advanced Keyword Search
- Boolean/Bracketed Boolean Search
- Exact Phrase Search

- Field Restrictions
- Field Text Search
- Fuzzy Search
- Parametric Search
- Proper Names Search
- Proximity Search
- Soundex Keyword Search
- Synonym Search

Spell Check

IDOL can automatically spell check the query text it receives and suggest correct spelling for terms that its dictionary does not contain.

Summarization

IDOL accepts a piece of content and returns a summary of the information. IDOL can generate different types of summary.

- **Conceptual Summaries**. Conceptual summaries contain the most salient concepts of the content.
- **Contextual Summaries**. Contextual summaries relate to the context of the original inquiry. They provide the most applicable dynamic summaryin the results of a given inquiry.
- **Quick Summaries**. Quick summaries include a few sentences of the result documents.

Taxonomy Generation

IDOL's automatic taxonomy generation feature can automatically understand and create deep hierarchical contextual taxonomies of information. You can use clustering, or any other conceptual operation, as a *seed* for the process.

The resulting taxonomy can:

- provide insight into specific areas of the information.
- provide an overall *information landscape*.
- act as training material for automatic categorization, which then places information into a formally dictated and controlled category hierarchy.

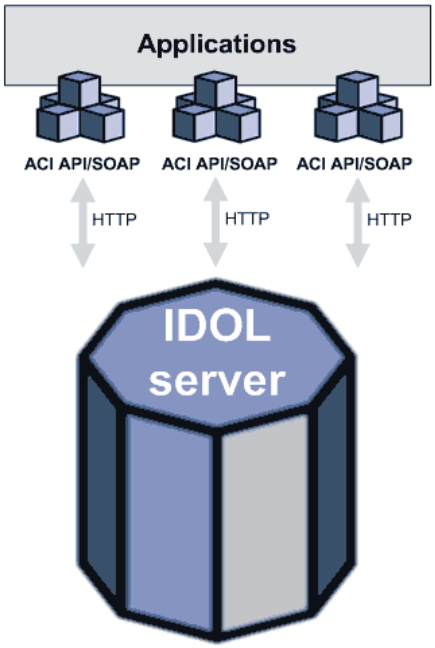
View Documents

IDOL server uses Autonomy KeyView filters to convert documents into HTML format for viewing in a Web browser.

IDOL System Architecture

IDOL server uses the ACI (Autonomy Content Infrastructure) Client API to communicate with custom-built applications that retrieve data using HTTP requests. It implements this communication over HTTP using XML and can adhere to SOAP.

Figure 2 ACI client API



Actions

When communicating with IDOL server, there are two main types of action:

- ACI (Autonomy Content Infrastructure) Actions
- Index Actions

ACI Actions request information or perform operations and return results. Different IDOL components accept different ACI actions. For example, the `Query` action requests results from the Content component, while the `UserRead` action requests user information from the Community component.

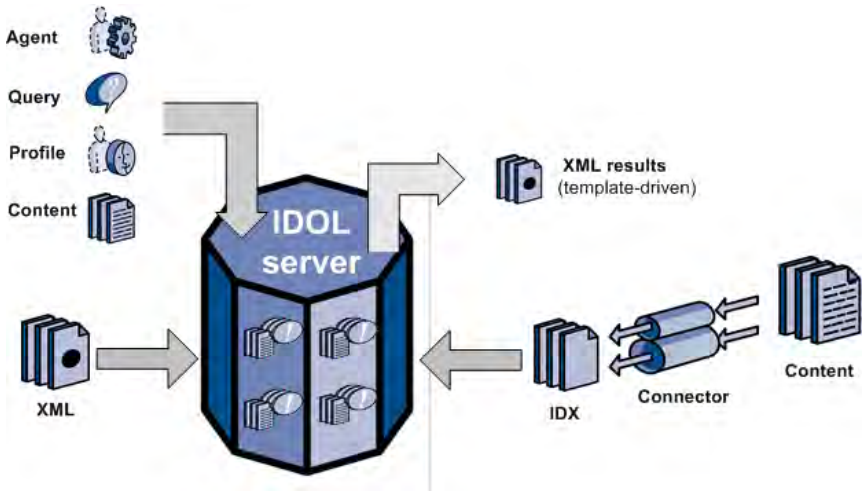
Index actions maintain the IDOL server data index. You use them to index data, delete data, and perform operations on the data within IDOL server.

Index and Query

You index documents into IDOL in Autonomy IDX format or in XML format (directly or using a Connector). IDOL stores the concepts of the document. In response to queries, agents, profiles or content, it returns a link to the result document. IDOL also returns a percentage weighting, which indicates how relevant the result document is to the original query.

IDOL can return results as XML (even if the document was not in XML format when it was indexed) or other formats, such as plain text, using XSLT:

Figure 3 Index and Query IDOL server



Security

It is often necessary to ensure that users can access or retrieve only data that they are authorized to view. IDOL enables you to set permissions for users or groups of users, to protect information, and ensure only the correct people can access it.

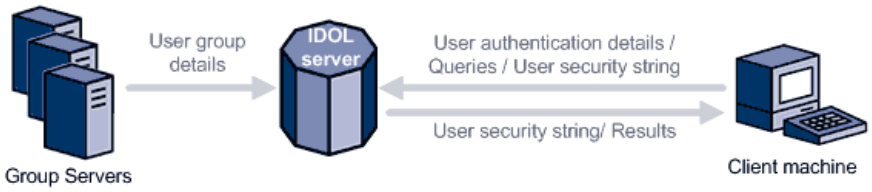
For more information on security, refer to the *IAS Administration Guide*.

Text Queries

IDOL contains data that has been aggregated from one or more repositories. In this example each of the repositories has its own group server that stores the repository user names and the groups that these users belong to. IDOL aggregates this security information from the group servers.

When users log onto a client, their authentication details are sent to IDOL. IDOL returns the user security details to the client, which stores them until the client logs off or the session times out. Every time users send a text query from a client, the client attaches their security details to the query string that it sends to IDOL.

IDOL uses the security information in the query string to check the user permissions. It matches the security string against the document access control lists (ACLs). IDOL returns documents that match the query and that the user has permission to see.

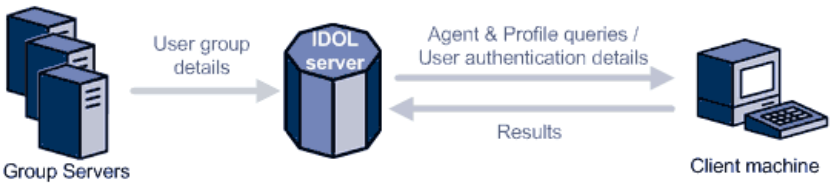


Agent, Profile and Category Queries

IDOL contains data aggregated from one or more repositories. In this example, each of the repositories has its own group server that stores the repository user names and the groups that these users belong to. IDOL aggregates this security information from the group servers.

The client sends an agent, profile, or category query to IDOL. IDOL (which stores all agents and profiles) matches this agent, profile or category against the documents it contains.

IDOL uses the information that it receives from the group servers to check the user permissions. It returns documents that match the query and that the user has permission to see.



Community Queries

IDOL stores user agents and profiles, to match them against community queries (that is, any type of query that returns agents or profiles).

When a client sends a community query to IDOL, IDOL matches it against the agents and profiles it stores. It returns matching agents, profiles, or both to the client.

CHAPTER 2

Types of IDOL Systems

You can set up IDOL systems in different ways to suit the requirements of your organization. This chapter discusses various IDOL setups and the use cases for each.

- IDOL Packages
- IDOL Core Components
- IDOL System Setups
- Common IDOL Installations and Configurations

NOTE These setups are not mutually exclusive.

IDOL Packages

There are two IDOL packages available:

- IDOL Standalone
- IDOL Administration

The main difference is that you set up and run IDOL Standalone without the IDOL Administration interface, and IDOL Administration includes a user interface. The core functionality is the same in both IDOL Standalone and IDOL Administration.

You configure IDOL Standalone by editing the IDOL Server configuration file directly, whereas you configure IDOL Administration using the user interface.

IDOL Core Components

The following are common IDOL components.

Service	Description
IDOL Server 	A central repository that holds indexed data. IDOL server also includes Agentstore, Category, Community Content, Index Tasks, IDOL Proxy and View. Refer to the *IDOL Server Administration Guide* for more details
Agentstore 	Indexes, collects, manipulates and stores agent information.
Category 	Stores predefined or customized categories in which data is organized.
Community 	Stores information about users, roles, and permissions.
Content 	Indexes, collects, manipulates and stores data.
DiSH 	Distributed Service Handler. Manages IDOL licenses.
IDOL Proxy 	Routes actions to the correct components. IDOL Proxy is also responsible for starting, restarting and stopping IDOL components in a unified IDOL server configuration.
IndexTasks 	Manipulates the index files and IDX files (for example, adds additional fields relevant to the content) before it passes the data to Content for indexing.
View 	Converts documents to HTML format for viewing in a web browser.

Service	Description
Statistics Server 	Accumulates events from client applications or from a script that reads IDOL log files, then uses that data to report statistics. Refer to the *IDOL Server Administration Guide* for more information.
Query Manipulation Server 	Manages promotions, modifies queries to IDOL server, and manipulates results from IDOL server. Refer to the Query Manipulation Server Administration Guide for more information.
DiSH 	Distributes and manages licenses for all IDOL server components. Refer to the *DiSH Administration Guide* for more information.
DAH 	Distributes action requests across IDOL servers or Content components. Refer to the *DAH Administration Guide* for more information.
DIH 	Distributes indexing requests across IDOL servers or Content components. Refer to the *DIH Administration Guide* for more information.
DLH 	Determines the current load of connected IDOL servers or DAHs and specifies which server the client must use. Refer to the *DLH Administration Guide* for more information.
Connectors 	Gather data from different sources for indexing into IDOL server. By default, an IDOL server Standalone installation includes the File System Connector and the HTTP Connector. The File System Connector gathers data from system files, while the HTTP Connector gathers data from Web Servers (internet or intranet). The IDOL Server with Administration installation also includes the Notes Connector, which gathers data from Lotus Notes databases. Refer to the Connector Administration guides for more details.
Omni Group Server 	Collects security information from security repositories and uses it to ensure that users can access only documents that they have permission to access. Refer to the *Intellectual Asset Protection (IAS) Administration Guide* for more information.

IDOL System Setups

The flexibility of IDOL allows you to create systems that suit your environment and the unique needs of your organization. You can set up IDOL in the following ways:

- Unified Setup
- Component Setup
- Distributed Setup
  - Distributed Unified Setup
  - Distributed Component Setup
- Front End Applications Setup

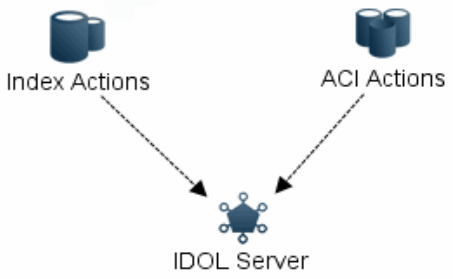
The installation procedures to support these setups are described in “Common IDOL Installations and Configurations” on page 47.

Unified Setup

A unified setup is the most basic setup. It uses an integrated IDOL Server (rather than separate IDOL components). You configure all IDOL component operations using the IDOL server configuration file.

In a unified setup, you index data into IDOL server (for example by using connectors, such as the File System Connector or HTTP Connector). IDOL stores and processes the data centrally. Similarly, IDOL processes actions centrally.

Figure 4 Unified IDOL server



Related Topics

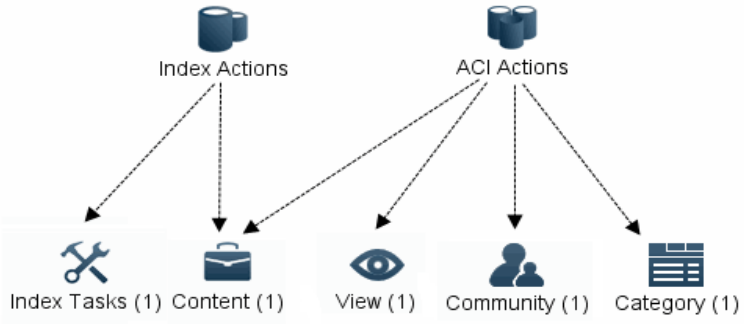
- “Distributed Unified Setup” on page 44
- “Unified Setup (Basic Installation)” on page 48

Component Setup

A component setup does not use IDOL server, but rather a combination of its components, such as Content, Community, IDOL Proxy, Group Server and so on. This advanced setup is for administrators who are already familiar with IDOL architecture.

This setup can be useful if you do not require all the IDOL components. You can set up components on separate hardware for enhanced speed performance for indexing and querying. You configure components separately, which can be useful for optimization and adjustments, as well as troubleshooting.

Figure 5 Component setup



You configure each component with its own configuration file. For example, you configure the `content.exe` in the `content.cfg` file, you configure the `category.exe` in the `category.cfg` and so on.

Each component server must also contain any extra files or modules that you configure.

NOTE There are also other dependencies between components, which you must configure in the component configuration files. For example, the Community and Category configuration files must contain the host and port details for the Agentstore component.

Related Topics

- “Distributed Component Setup” on page 45
- “Component Setup (Advanced Installation)” on page 49
- “Retrieval-Only Setup (Advanced Installation)” on page 53

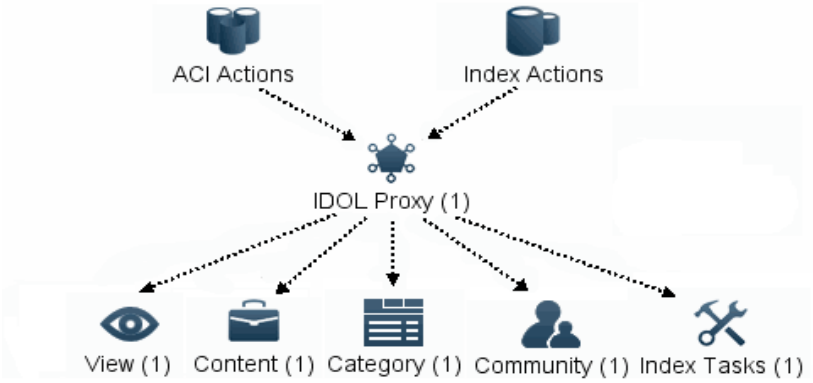
IDOL Proxy

You can use a standalone IDOL Proxy when you install components on different servers (typically for better performance). IDOL Proxy forwards all requests to the appropriate component. For example, it sends a `Query` action or Indexing request to the Content component, and it sends a `UserRead` action to the Community component.

In a unified IDOL Server, IDOL Proxy starts, stops and restarts components. In a standalone configuration such as the one represented by Figure 5, IDOL Proxy cannot start, stop or restart IDOL components. If a component becomes inactive for any reason, you must restart it separately. IDOL Proxy reports an error when it cannot contact a component. When the component becomes available, IDOL Proxy starts to forward actions again.

The configuration file for IDOL Proxy must contain the host IP address and ACI port number for each of the IDOL components. It dynamically configures other ports (such as the index port) when the components start up.

Figure 6 Component setup with IDOL Proxy



Distributed Setup

A distributed setup involves using a Distributed Index Handler (DIH) and Distributed Action Handler (DAH) to route actions to multiple IDOL server instances, or Content servers (in a component setup).

This kind of setup is effective for load-balancing among components or servers, as well as having no single point of failure. DIH and DAH balance indexing and action requests among the IDOL Servers (in a unified setup) or Content servers (in a component setup). You can either set up the distributed system in mirror mode or non-mirror mode.

Related Topics

- “Distributed Setup (Advanced Installation)” on page 50

- “Distributed Setup Optimized for Retrieval (Advanced Installation)” on page 51

Mirror Mode

In a mirrored setup, you store the same set of data in each instance of IDOL Server (or Content). The IDOL servers are exact copies of each other and you must configure them in the same way.

Run the DIH in mirror mode to ensure uninterrupted service if one of the IDOL servers becomes inoperable. While one IDOL server is inoperable, the DIH continues to index data into its identical copies, which are also still available to return data for queries.

DIH sends all index actions to all connected IDOL Servers.

In mirror mode, you can configure the DAH to distribute ACI actions in one of two ways:

- **Load Balancing**. The DAH assigns each incoming action to just one of the connected IDOL servers (using a cumulative predictive algorithm that spreads the action load efficiently).
- **Failover**. The DAH forwards incoming actions to the first IDOL server that you list in the DAH configuration file. If this IDOL server stops responding for any reason, the DAH marks it as down and switches to the next IDOL server.

Non-Mirror Mode

In a non-mirrored system, you distribute the data equally among the IDOL servers (or Content servers).

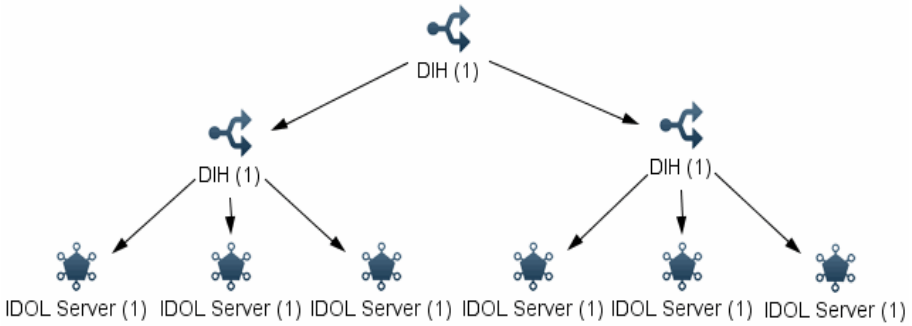
Run the DIH in non-mirror mode if the amount of data to index is too large for a single IDOL server. If the IDOL servers that the DIH indexes into are on different machines, the indexing process requires less time.

In non-mirror mode, the DAH sends ACI actions to all connected IDOL servers. You can configure the DAH to combine the results in different ways when it returns them.

Chaining Distribution Servers

You can set up multiple DIH and DAH instances in a chained configuration. For example, a parent DIH or DAH distributes actions to child DIH or DAH servers, which in turn distributes to child IDOL servers.

Figure 7 Chained DIH servers



In this configuration, the parent DIH and DAH distributes actions to child DIH and DAH servers in the same way as it distributes to child IDOL servers. Each child DIH or DAH accepts all IDOL server actions and forwards them.

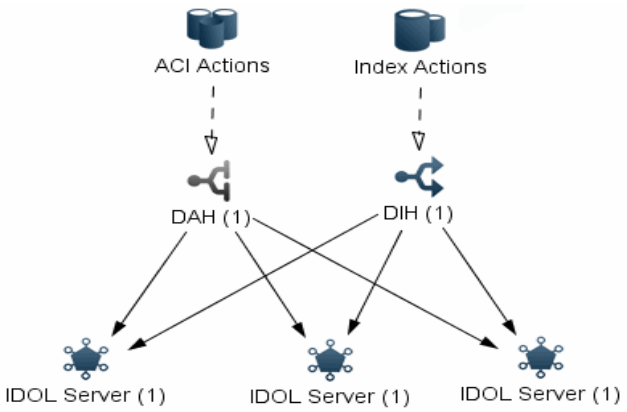
Some actions may have a different effect when you send them to a child DIH or DAH server rather than an IDOL server, because the actions goes to multiple IDOL servers.

Chaining provides an extra level of redundancy both at the DIH or DAH, and the IDOL server level. It also distributes network traffic and system load over a larger number of computers. A chained configuration provides a pool of IDOL servers that are both fault-tolerant for maximum availability and distributed for the best performance.

Distributed Unified Setup

The key services of a unified distributed setup are IDOL Server, DIH and DAH. The DIH and DAH route indexing operations and actions to multiple instances of IDOL Server.

Figure 8 Unified distributed setup



You can configure the DIH and DAH in two ways:

- As standalone components, using separate DIH and DAH configuration files.
- Using the IDOL server configuration file. In this case, the DIH and DAH are part of a parent IDOL server, while distributing actions and indexing operations to child Content servers.

Related Topics

- "Unified Setup" on page 40
- "Distributed Setup (Advanced Installation)" on page 50
- "Distributed Setup Optimized for Retrieval (Advanced Installation)" on page 51

Distributed Component Setup

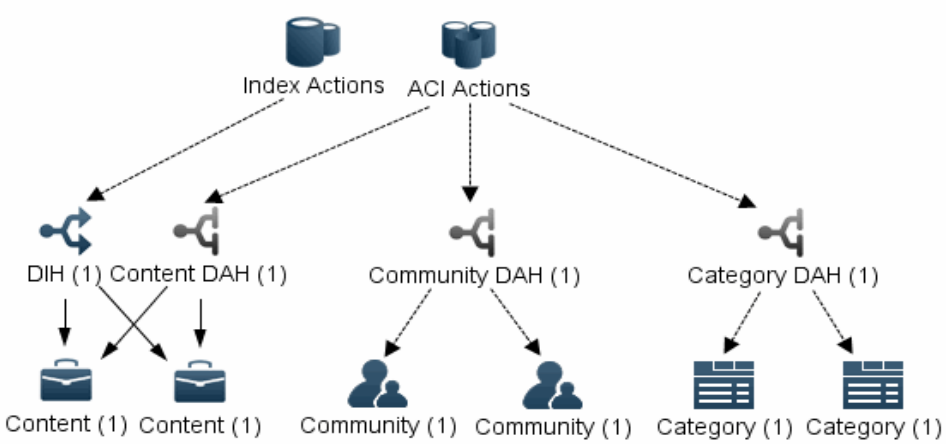
A distributed system with standalone components uses a combination of IDOL components as well as DIH and DAH. You configure the IDOL components, and the DIH and DAH, separately using their own configuration files.

Figure 9 shows an example scenario where you configure all components as standalone, with two Content components, two Community components and two Category components.

- Index actions are sent to the DIH, which distributes them between the two Content components.
- ACI actions are distributed between the three DAHs (for example by a front-end application, or an IDOL Proxy).
  - Actions for Content are sent to Content DAH, which distributes actions between the two Content components.
  - Actions for Community are sent to Community DAH, which distributes actions between the two Community components.
  - Actions for Category are sent to Category DAH, which distributes actions between the two Category components.

  DAH cannot distribute all ACI actions in non-mirror mode, so the Community and Category components in this example must be mirrored.

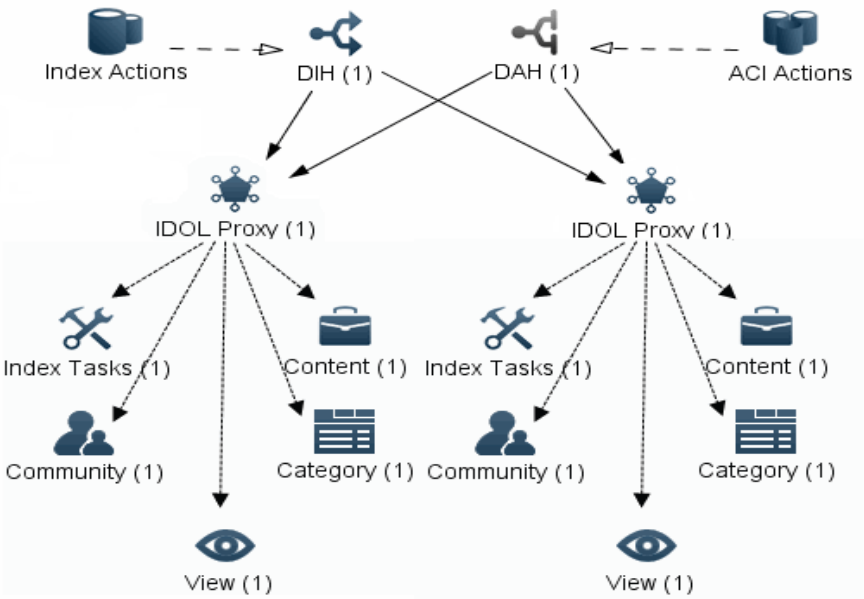
Figure 9 Components in a distributed setup



Alternatively, you can configure the DIH and DAH to communicate with several instances of IDOL Proxy. For example, Figure 10 shows the DIH and DAH send actions to two instances of IDOL Proxy. Each IDOL Proxy connects to a set of IDOL components, and can distribute actions to the relevant components.

In this scenario, IDOL Proxy automatically routes actions to the appropriate component, whereas in the example in Figure 9, you must use another method to route actions correctly.

Figure 10 Components in a distributed setup using IDOL proxies.



Related Topics

- “Component Setup” on page 41
- “Distributed Setup Optimized for Retrieval (Advanced Installation)” on page 51
- “Retrieval-Only Setup (Advanced Installation)” on page 53

Front End Applications Setup

If you use multiple front-end applications, it can be beneficial to use a setup where a separate IDOL installation hosts each application. For example, you can set up an installation of Retina to communicate with one IDOL system, with any required IDOL servers, distribution and security. Then you set up a custom front-end application to communicate with a separate IDOL system.

Separating the applications in this way has several advantages:

- It is straightforward to use different data sets for each application, if required. Each different installation can have its own set of connectors, indexing data from the required sources.
- It can be easier to manage security. For example you can separate an application for use on an intranet from another application that is accessible over the Internet.
- You can configure each IDOL instance individually, to optimize performance and functionality for its associated front-end application.
- You can optimize and scale resources more efficiently.
- It is straightforward to update your application or IDOL installation without causing interruptions to other applications.
- It is easier to test the effects of new configurations.

Common IDOL Installations and Configurations

The type of IDOL system that you want to run may affect choices you make during the installation process. It is also necessary to configure the system differently.

This section lists some common setups for IDOL systems, that you may like to think about when installing IDOL.

NOTE The following example installation instructions apply to IDOL Standalone. In IDOL Administration, the decisions required at installation are different, and you design the system in IDOL Administration and then deploy it. The same setups are possible. For more information, refer to the *IDOL Administration User Guide*.

Related Topics

- “IDOL System Setups” on page 40
- “Install IDOL” on page 72

Unified Setup (Basic Installation)

Using the **Quick Install** option with the IDOL Standalone Installer installs the most common settings for IDOL. This option installs IDOL server with all the main components such as:

- IDOL Proxy
- Content
- Community
- Category
- Agentstore
- IndexTasks
- View

The installation also provides an IDOL Server configuration file containing common settings for these components, except for IndexTasks, which is not configured by default. To activate IndexTasks, you must configure tasks in the IDOL server configuration file.

The installation also provides a configuration file for the Agentstore component. You configure this component separately from the other IDOL components.

When you start the IDOL server, it starts all components, except for IndexTasks. IndexTasks is started only if it is configured.

Quick Install also includes:

- Distributed Service Handler (DiSH)
- Retina web application
- HTTP Connector

- File System Connector

For many systems, this option is the most appropriate set-up. However, to include secured search, you may also need to install Omni Group Server (OGS). Additionally, to scale larger systems it may be more suitable to use a distributed setup, or a component setup.

Related Topics

- “Unified Setup” on page 40
- “Other IDOL Components (Individual Component Installation)” on page 54

How to Install

To use the basic IDOL installation, you can select the **Quick Install** option when you install IDOL. Alternatively, you can select the **Advanced Install** option, and select only IDOL Server and DiSH from the list of options, along with any other applications that you require.

Component Setup (Advanced Installation)

A system using IDOL components can be useful if you want to:

- use only the required IDOL components to optimize the most common tasks.
- simplify configuration.
- simplify starting, stopping and re-initializing specific components.
- scale resources separately for different components according to their requirements.

For systems where you have large processing requirements for certain tasks, you can separate the IDOL components onto their own servers. This separation allows you to isolate and scale resources independently for different functions, according to their individual requirements. You can also place the components on their own hardware.

You may want to set up IDOL Proxy, to route actions to all the IDOL components that you use. In this type of system, you send all actions to IDOL Proxy, which then routes them to the appropriate component. You can also use DIH and DAH for distribution for improved performance or fault tolerance.

Related Topics

- “Component Setup” on page 41
- “Retrieval-Only Setup (Advanced Installation)” on page 53
- “Other IDOL Components (Individual Component Installation)” on page 54

How to Install

For this installation, select the **Advanced Install** option when installing IDOL. You can then select just the IDOL server from the component options. From the IDOL server installation you can select the relevant components, and delete or move the others. You must create configuration files for each of your standalone components.

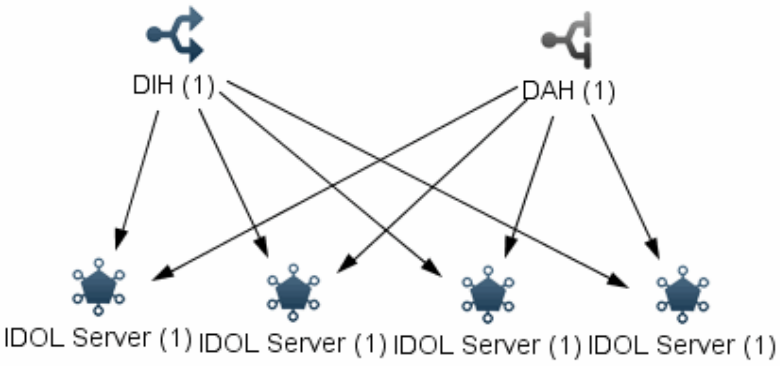
Distributed Setup (Advanced Installation)

For larger systems with a large amount of data to index and a large number of actions, a distributed setup may be the most suitable. In this setup, you configure DIH and DAH to distribute actions and index actions to different instances of IDOL server, or Content servers.

You can set up multiple instances of a basic IDOL installation, such that DIH and DAH distribute actions to unified IDOL server installations.

In this case, you also distribute components such as Category and Community, which can improve performance when you use these components extensively.

Figure 11 Distributed setup with unified IDOL servers



Related Topics

- “Distributed Setup” on page 42
- “Distributed Setup Optimized for Retrieval (Advanced Installation)” on page 51
- “Retrieval-Only Setup (Advanced Installation)” on page 53

How to Install

If you want to install multiple instances of a basic IDOL installation with distribution, you can choose **Quick Install** to install each IDOL instance. Alternatively, you can select **Advanced Install**, and install only IDOL Server, without the other components included in the Quick Install.

For the server where you are installing the Distributed Handlers, select **Advanced Install**. Select the Distributed Handlers from the list of options, and deselect all the other components. This option installs a single DAH and DIH pair along with standalone `DIH.cfg` and `DAH.cfg` configuration files. You can also select any other applications you require.

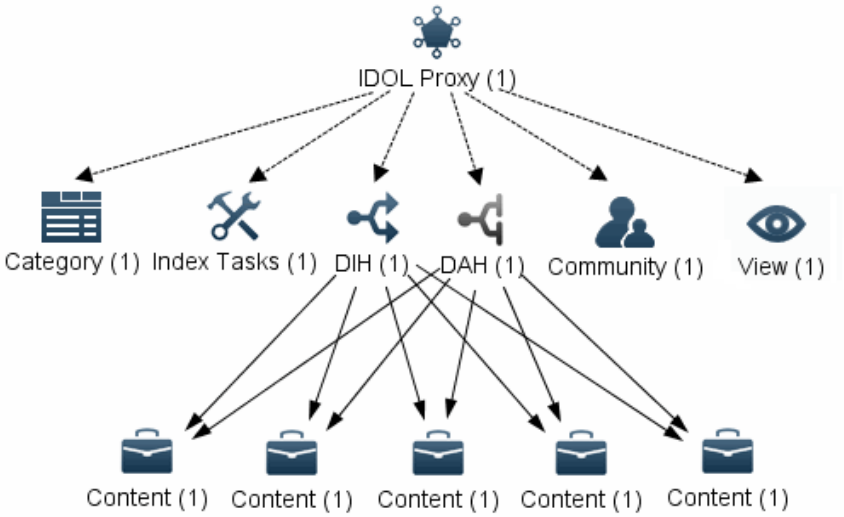
For more information about installing the DAH and DIH with IDOL, refer to the *Distributed Action Handler (DAH) Administration Guide* and the *Distributed Index Handler (DIH) Administration Guide*.

Distributed Setup Optimized for Retrieval (Advanced Installation)

In some cases, using several instances of IDOL server may mean that you have several copies of a component that you use in only one instance of IDOL server, such as View and Agentstore.

Often, retrieval and indexing requires a large amount of resources. In this case, you can install an IDOL server instance, similar to the basic installation, with a DIH and DAH, rather than the Content component. You can then set up several standalone Content components with the required configurations.

Figure 12 Distributed Content servers



In this scenario, you send all actions to the IDOL proxy component of the IDOL server. IDOL Proxy then distributes ACI actions to the appropriate components. It sends any ACI actions for the Content component to the DAH, which distributes these actions to the Content servers.

If you configure pre-indexing tasks, then IDOL Proxy sends index actions to the IndexTasks component first. Index Tasks processes them and sends them to the DIH. If you do not configure pre-indexing tasks, IDOL Proxy sends index actions directly to the DIH. The DIH then distributes the index actions to the Content servers.

The Content servers can either have:

- identical configuration and content for load balancing and fault tolerance.
- different configurations for improved performance and flexibility.

Related Topics

- "Distributed Setup (Advanced Installation)" on page 50
- "Component Setup (Advanced Installation)" on page 49

How to Install

To install this option, select the **Advanced Install** option when installing IDOL. You can either choose the `Distributed IDOL Server` install set, or you can select the distributed handlers from the list of components, as well as IDOL server. This option installs the IDOL Server without the Content component, and with the DIH and DAH.

For more information about installing the DAH and DIH with IDOL, refer to the *Distributed Action Handler (DAH) Administration Guide* and the *Distributed Index Handler (DIH) Administration Guide*.

To create the Content servers, select the **Advanced Install** option and select only IDOL server from the list of components. Once the installation is complete, you can copy the Content component and associated files and move them to separate servers. You then configure the DIH and DAH in the main installation to send actions to these Content servers.

A standalone Content server needs:

- `content.exe`
- `content.cfg` (configuration file)

The server must have a `langfiles` directory with the stoplists and conversion tables for any languages and character encodings that you use. It must also have a `modules` directory with any modules that you configure.

To use XSL templates, you also need the `templates` directory, and `autnxslt.dll` (windows) or `libautnxslt.so` (UNIX).

Secured Search (OGS Installation)

To add secured search for IDOL Server, you may need to install Omni Group Server (OGS). OGS ensures that users can access only documents that they have permission to access. OGS can collect security information from a number of different security repositories and store this information, updating it at regular intervals.

When a user starts a session with IDOL server, the client application requests security information from IDOL server. IDOL server retrieves the user security details from OGS. The client application adds the security information to every subsequent query that it sends to IDOL server. IDOL server can then compare the user security details to the document access control lists (ACL) to determine what type of access to grant.

Related Topics

- "Unified Setup (Basic Installation)" on page 48

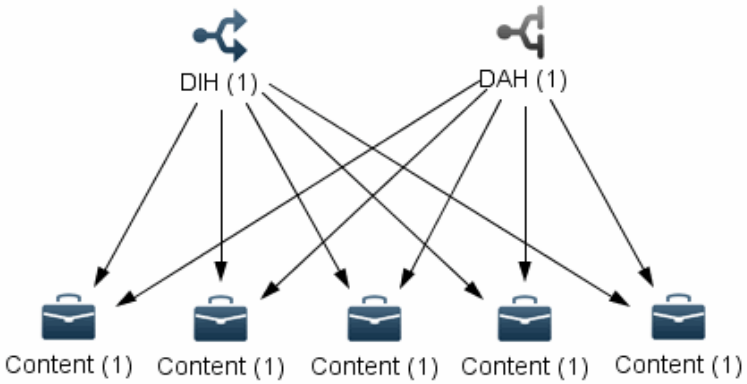
How to Install

For information on installing and configuring OGS, refer to the *Intellectual Asset Protection System (IAS) Administration Guide*.

Retrieval-Only Setup (Advanced Installation)

If you require only the basic indexing and retrieval functionality of IDOL server, a more minimalist setup may be suitable. In this case, you set up a number of Content servers, with DIH and DAH servers distributing index and ACI actions between them.

Figure 13 Retrieval-only setup



Depending on the size or requirements of the system, it can either be mirrored for fault tolerance or non-mirrored for performance. You can also use chained DIH and DAH servers to allow for both.

Related Topics

- “Distributed Setup (Advanced Installation)” on page 50
- “Component Setup (Advanced Installation)” on page 49

How to Install

In this case, use the **Advanced Install**, and select just the distributed handlers from the list of components, deselecting all other components. This option installs a single DAH and DIH pair along with standalone `DIH.cfg` and `DAH.cfg` configuration files.

To create the Content servers, select the **Advanced Install** option and select only IDOL server from the list of components. After the installation is complete, you can copy the Content component and associated files and move them to separate servers. You can then configure the DIH and DAH to send actions to these Content servers.

A standalone Content server needs:

- `content.exe`
- `content.cfg` (configuration file)

The server must have a `langfiles` directory with the stoplists and conversion tables for any languages and character encodings that you use. It must also have a `modules` directory with any modules that you configure.

To use XSL templates, you also need the `templates` directory, and `autnxslt.dll` (windows) or `libautnxslt.so` (UNIX).

Other IDOL Components (Individual Component Installation)

It may be suitable to place other components related to IDOL server on separate servers. For example, you can place connectors on a separate server to IDOL, and configure them to index into an IDOL server or a DIH.

It might also be suitable to place other components on a separate *miscellaneous* server. For example, you can place the following functions together on their own server.

- Distributed Service Handler (DiSH)
- Statistics Server
- Omni Group Server (OGS)
- Autonomy Business Console (ABC)
- Autonomy Collaborative Classifier (ACC)

- front-end applications such as Retina and Portal-in-a-Box

Alternatively, you might want to have a separate DiSH on all servers that host IDOL services, so that you can start a single service with its own DiSH.

How to Install

Most of the applications in this example have their own installers, separate from the IDOL installer. For more information on how to install them, refer to the relevant administration guides.

Related Topics

- "Front End Applications Setup" on page 47
- "Other IDOL Components (Individual Component Installation)" on page 54

CHAPTER 3

Applications

There are many applications that you can use to access, search, and customize the data in your IDOL system. This section describes some of the available applications that you can use to access and administer your IDOL system.

- Front-End Applications
- Administrative Applications

Front-End Applications

There are a variety of applications that provide a user interface through which you can access, customize, search, and modify data in your IDOL system.

This section describes some of the available IDOL front-end applications. There are more applications available for different business purposes.

Retina

Retina is an Autonomy Web interface application that plugs into Autonomy IDOL server, providing administrators and users with a platform from which they can retrieve and view search results.

The Retina user interface provides a wide range of features, such as Retrieval, Community, Channels, and Clustering.

For more information, refer to the *Retina Administration and User Guide*.

Portal-in-a-Box

Autonomy Portal-in-a-Box delivers the most comprehensive and automated information Portal for both inside and outside the firewall in one easy-to-use package. Autonomy Portal-in-a-Box automates the most critical processes including categorization, personalization, hypertext link management and highly personalized information delivery.

For more information, refer to the *Portal-in-a-Box Administration Guide* and *Portlets User Guide*.

Autonomy Express Search

Autonomy Express Search is an easily accessible search bar that allows you to quickly search for content from a variety of sources, from different applications.

You can also use Express Search with Autonomy Miner. Miner is an extension of Express Search with the same powerful searching capability as Express Search, combined with the ability to work with a larger set of search results.

For more information, refer to the *Autonomy Express Search User Guide*.

Custom Front-End Applications

The ACI (Autonomy Content Infrastructure) Client API allows you to create your own custom front-end applications.

The ACI API enables easy communication between custom-built applications and Autonomy ACI servers, as well as simple manipulation of the results sets. It is available for a number of programming languages, including C, Java, PHP, COM, COM+ and .NET.

For more information, refer to the *ACI API Programming Guide*.

Administrative Applications

Autonomy provides several applications that allow you to manipulate and modify the data in IDOL, and to monitor the IDOL services and applications.

Autonomy Business Console

The Autonomy Business Console (ABC) allows you to manage the content of data indexed in IDOL servers. By creating and modifying different *business projects*, you can predefine which results are returned to users and how the user views them. You can also view statistical information to help you refine your business projects and make them more effective.

ABC integrates with IDOL server, so that you can manage IDOL content, and use IDOL server for authentication and categorization. You can also optionally integrate ABC with the Query Manipulation Server (QMS) and Statistics Server.

ABC business projects let you manage IDOL content.

- **Facet projects** allow you to define a set of parametric field values to control user navigation paths.
- **Promotions projects** allow you to set a wide variety of criteria (query terms, facets, and so on) to trigger business promotions.
- **Query tuning projects** allow you to associate documents with specified user queries or boost document relevance to return desired documents to users.
- **Term relation projects** allow you to define synonyms, hypernyms, and hyponyms to refine or expand search results.

There are several shared tasks that you can use with any of the main projects.

- **IDOL Statistics Summary**. If your system is connected to a statistics server, you can analyze the effectiveness of your various business projects.
- **Deleted documents**. You can permanently delete undesirable documents from Autonomy server databases.
- **Import/Export**. If you are upgrading Autonomy Business Console, you can export your data to a proprietary data file, then import it into the upgraded version.
- **IDOL Search**. You can use the basic IDOL search to find documents and to test and refine projects.
- **Content Investigator**. This advanced search interface allows you to refine your search results and view suggested documents.
- **User Management**. ABC Administrators can manage the ABC users and the user roles that determine permissions.

For more information, refer to the *Autonomy Business Console User Guide*.

Autonomy Collaborative Classifier

Autonomy Collaborative Classifier (ACC) manipulates information-classification structures and indexes. Classifying information into categories and subcategories allows users to locate individual documents easily. A user can navigate the concepts represented by each category to find relevant documents. *Classification* organizes the documents in your indexes into content hierarchies, called taxonomies. A *taxonomy* is a hierarchical organization of information in categories. The ACC Taxonomy module allows you to create, view, edit, and test taxonomies.

You can also import existing taxonomies, including Web URLs, file paths, Autonomy taxonomies, as well as corporate and other third-party taxonomies.

ACC integrates with IDOL server, but primarily uses the IDOL server Category component for categorization and taxonomy creation.

For more information, refer to the *Autonomy Collaborative Classifier User Guide*.

CHAPTER 4

Security in IDOL

This chapter provides an overview of the Intellectual Asset Protection System. For more information on security, refer to the *Intellectual Asset Protection System Administration Guide*.

Security Overview

Autonomy provides the software infrastructure that automates operations on unstructured information. This software infrastructure is based on IDOL server.

IDOL server makes it possible for organizations to process digital content automatically and allows applications to communicate with each other. It consists of data operations that integrate information by understanding content, and is therefore data agnostic.

The Intellectual Asset Protection System provides an integrated security solution to protect your data:

- Front end security

  At the front end, authentication checks that users are allowed to access the system on which result data is displayed.

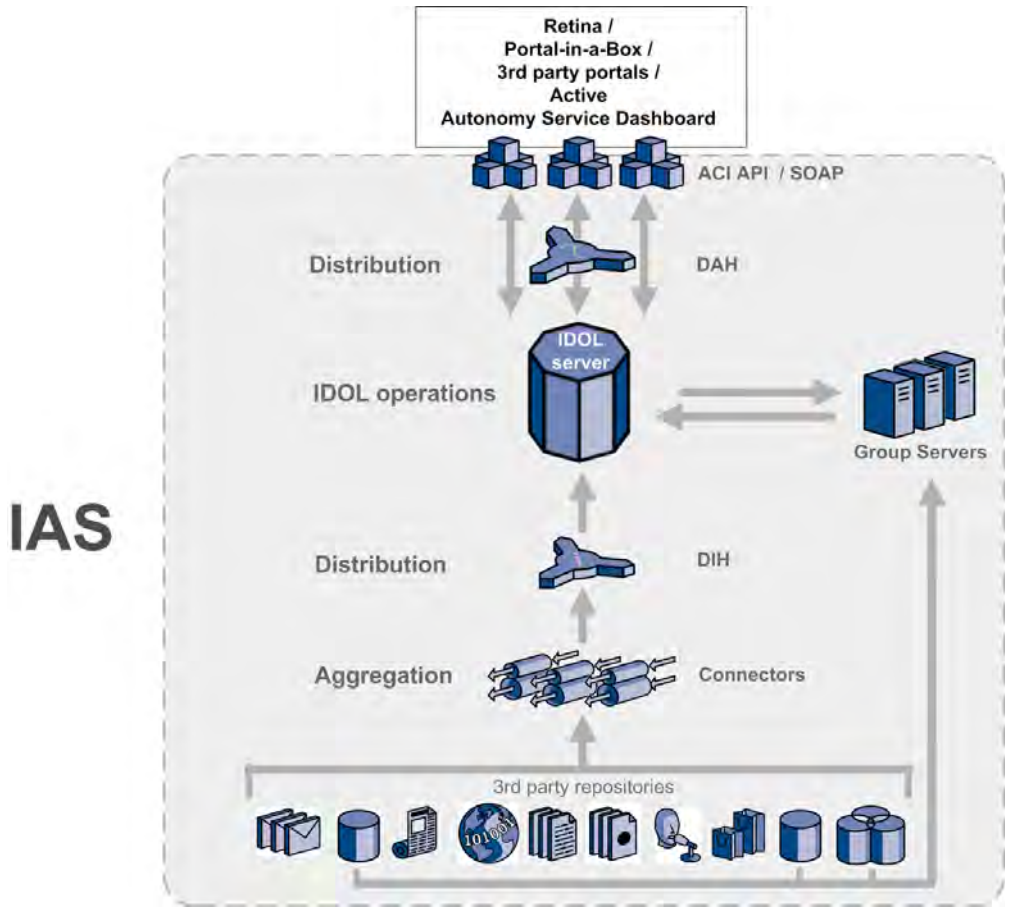
- Back end security

  At the back end, entitlement checking and authentication combine to ensure query results only include documents the user is allowed to view from repositories the user is allowed to access.

- Secure communications

  You can encrypt communications between ACI servers and any applications using the Autonomy ACI API.

Figure 14 Data Flow and Security



Front-End Security

To access Autonomy front-end applications such as Portal-in-a-Box, users must log in to the system for authentication. For your own applications, you can use the Autonomy API to customize the log-in process.

Autonomy supports industry standards for authentication to third-party systems, including the following:

- Windows NT and 2000 login
- LDAP authentication
- Other third-party software (for example, Lotus Notes)

A single log in allows a user to access all the systems for which they have permission. Autonomy IDOL server enables you to store and update user security details for this purpose.

In addition, Kerberos supports integration of IDOL into a single sign-on environment that uses Kerberos for authentication, such as Microsoft's Active Directory.

Back-End Security

Autonomy IDOL server security modules allow you to integrate Autonomy applications with third-party data repositories while enforcing security restrictions. You can thereby ensure users can only access documents for which they have the necessary permissions.

Autonomy has implemented the following security modes:

- **Unmapped.** When a user queries IDOL server, IDOL server checks the security entitlement of a user in real time at the original data repositories of documents that match the query.

  The advantage of this is the security information is up-to-date. The disadvantage is the IDOL server response can be slow because IDOL server has to connect to the original data repositories to check permissions for each result document.

  This security mode is suitable for environments where the security entitlement for documents changes frequently.

- **Mapped.** When a connector indexes documents into an Autonomy IDOL server, the Access Control List (ACL) of a document is encrypted and mapped into an IDOL server field. When a user queries IDOL server, it compares the

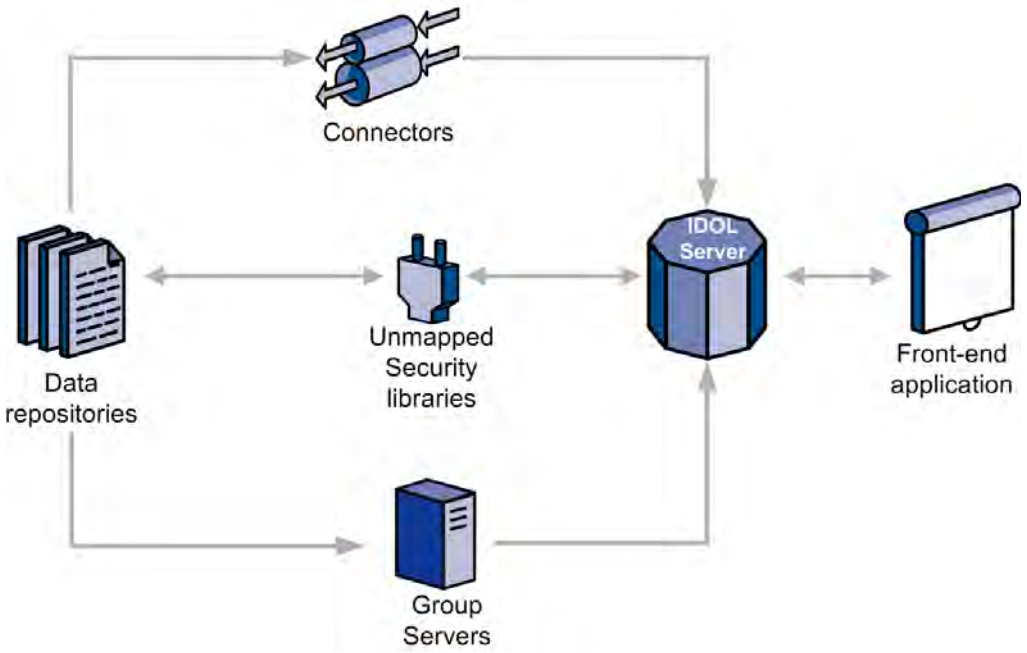
user's details against the ACLs that it contains to determine whether the user is permitted to view documents that match the query.

The advantage of this process is that IDOL server can respond faster as it does not need to connect to the original data repository to check the security information for each document. The disadvantage is that there might be a delay between the security settings changing in the original data repository and the information updating in IDOL server.

This security mode is suitable for environments where the security entitlement for documents does not change often.

Unmapped Security: System Architecture

Figure 15 Unmapped Security: System Architecture



Connectors (for example, Notes Connector or ODBC Connector) retrieve documents from data repositories and index them into an IDOL server.

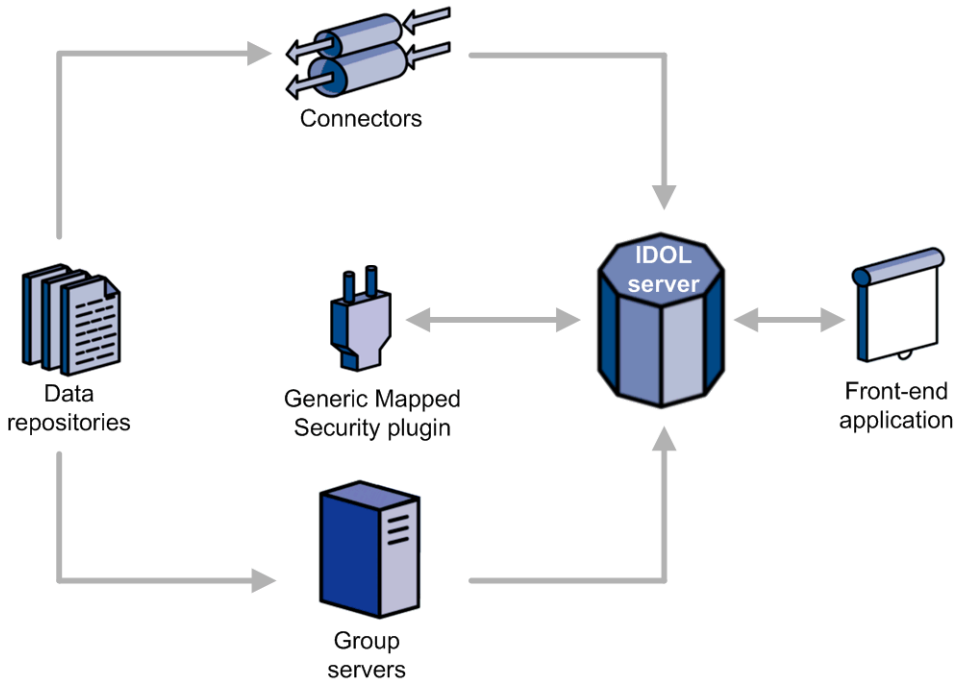
If a data repository stores users in groups, the users' group information can be stored in a group server for that repository. This makes it faster for the front end to get group information, and in some cases a group server is required because it is impossible for a front end to retrieve the group information from the repository in a reasonable time. IDOL server retrieves these groups from the group server and stores them, along with other user information such as Autonomy agents and authentication settings.

When a query is sent to IDOL server, it retrieves a list of groups to which the user belongs from its internal store.

IDOL server sends result document information and the user's security information to unmapped security libraries for the data repositories containing result documents for the user's query. The security libraries then connect to the data repositories and check the user's security details against the entitlement information for the documents matching the query. If the user is entitled to view a document, it is returned as a result to the application that sent the query—either the front-end application or IDOL server.

Mapped Security: System Architecture

Figure 16 Mapped Security: System Architecture



As in the case of unmapped security, for mapped security the connectors (for example, Notes Connector or ODBC Connector) fetch documents from data repositories and index them into an IDOL server. The connectors also index the encrypted ACL for each document into a structured field in IDOL server. An ACL contains information about which users, groups and roles are permitted to access a document.

If a data repository stores users in groups, the users' group information can be stored in a group server for that repository. This makes it faster for the front end to get group information, and in some cases a group server is required because it is

impossible for a front end to retrieve the group information from the repository in a reasonable time. IDOL server retrieves these groups from the group server and stores them, along with other user information such as Autonomy agents and authentication settings.

When a query is sent to IDOL server, it retrieves a list of groups to which the user belongs from its internal store.

IDOL server sends result document information and the user's security details to the Generic Mapped Security library plugin. The library compares user group information with each result document's ACL to determine which result documents the user is allowed to see. If the user is entitled to view a document, it is returned as a result to the application that sent the query—either the front-end application or IDOL server.

Available Security Libraries

Autonomy supplies unmapped security plugin libraries for specific repositories and a Generic Mapped Security plugin library that is supplied with IDOL.

If Autonomy does not supply a security library for a repository, you can create a custom library. This means you can, for example, create an ODBC mapped security type.

Unmapped Security Libraries

Autonomy currently supplies the following unmapped security plugin libraries:

Repository	Unmapped Security Plugin Library
Documentum	Yes
eRoom	
Exchange	Yes
FileNet	
iManage	
Lotus Notes	Yes
NetWare	Yes
NT	Yes
ODBC	Yes
Open Text	Yes
Oracle	Yes

Repository	Unmapped Security Plugin Library
PCDocs	
SharePoint	
UNIX	

The libraries act as plugins for the IDOL server, which uses them when instructed to do so in its configuration file.

Mapped Security Libraries

Use the Generic Mapped Security plugin library that is supplied with IDOL.

There are two exceptions: ODBC and Oracle. For mapped security in these two cases, create a custom mapped security type.

Secure Communications

IDOL Server and other ACI servers can be configured to communicate using encryption and Secure Socket Layer (SSL) communications.

Encrypt Communications

You can configure IDOL server for encryption of communications between front-end applications and ACI servers and then enable encryption:

- Create and configure an `[ACIEncryption]` section in IDOL server configuration file.
- Enable encryption by adding `EncryptResponse=true` to all action commands issued against ACI servers.

SSL Communications

It is also possible to configure Secure Socket Layer (SSL) communications for IDOL Server, as well as other ACI servers, front-end applications and connectors.

- The exact configuration you use depends on the component you are configuring. For more information, refer to the relevant component guide.

PART 2 Install and Run IDOL Server

This section describes how to install, run, and configure IDOL server.

- Install IDOL
- Run IDOL
- Configure IDOL Server
- IDOL Performance
- Tutorial: Index Data into IDOL

CHAPTER 5

Install IDOL

This section describes how to install IDOL and the software and hardware requirements to run IDOL.

System Requirements

This section describes the software and hardware requirements to run IDOL.

Basic Requirements

- IDOL must be installed by the system administrator (Windows only).
- You cannot run IDOL with restricted file system permissions (for example: disk quotas, file handle limits or memory limits).
- Your file system must permit file locking (for example, you cannot run IDOL on an NFS mount).
- Your network must support TCP/IP.
- If you are running anti-virus software on the machine hosting IDOL server, ensure it does not monitor the IDOL server directories, which can have a serious impact on IDOL server performance.

Supported Platforms

IDOL runs on a variety of Windows and UNIX platforms. For details of supported platforms, refer to the *IDOL 7.6 Release Notes*.

NOTE The documented platforms are the recommended and most fully tested platforms for IDOL. Other platforms may be supported and Autonomy can provide support for other platforms on request.

Recommended Hardware Specifications

Autonomy recommends the following hardware specifications.

- a dedicated SCSI disk
- 4 Gb RAM
- 100 Gb Disk
- a minimum of 2 dedicated CPU - XEON 3 GHz or above

TCP Port Requirements

On Windows platforms, if your IDOL service is receiving a large number of actions, or you have a large number of services on the same machine, your machine might run out of available TCP ports.

If your system does not have enough ports, you can edit the following Windows registry parameters:

- `TCPTimedWaitDelay`
- `MaxUserPort`

These parameters are located in:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\
Parameters\
```

For more information about these registry parameters, refer to the Microsoft Windows documentation.

Install IDOL

IDOL 7.6 includes two installations:

- IDOL with Administration

- IDOL Standalone

The IDOL with Administration installer has a multi-step installation process that installs IDOL and IDOL Administration. For details of the IDOL with Administration installer, refer to the *IDOL Administration User Guide*.

The IDOL Standalone installation is a simpler, single-pass installer that does not include IDOL Administration. This chapter describes the Standalone installer.

Install IDOL Server Standalone on Windows

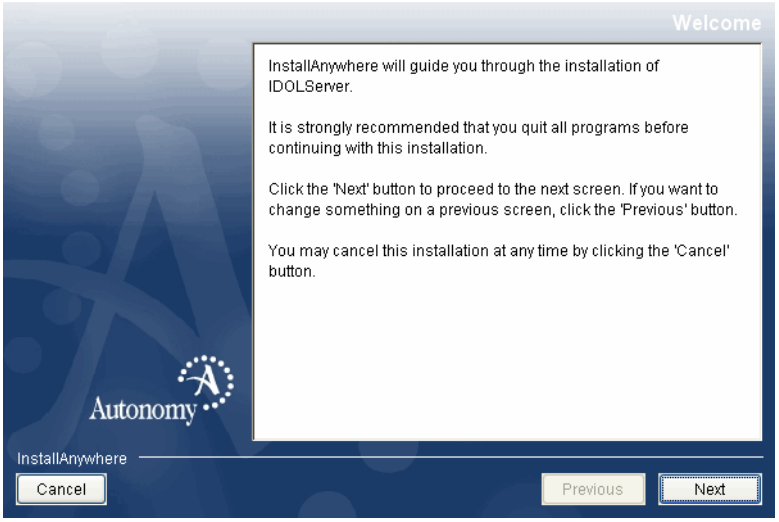
This section demonstrates the installer sequence for Windows.

To run the standalone installer program

1. Double-click the appropriate installer package:

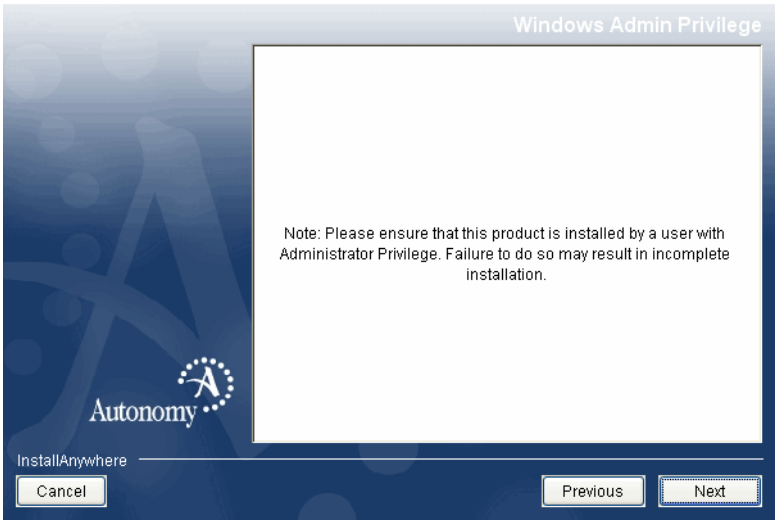
   `IDOLServer_`*`VersionNumber_Platform`*`.exe`

   The Welcome dialog box opens.



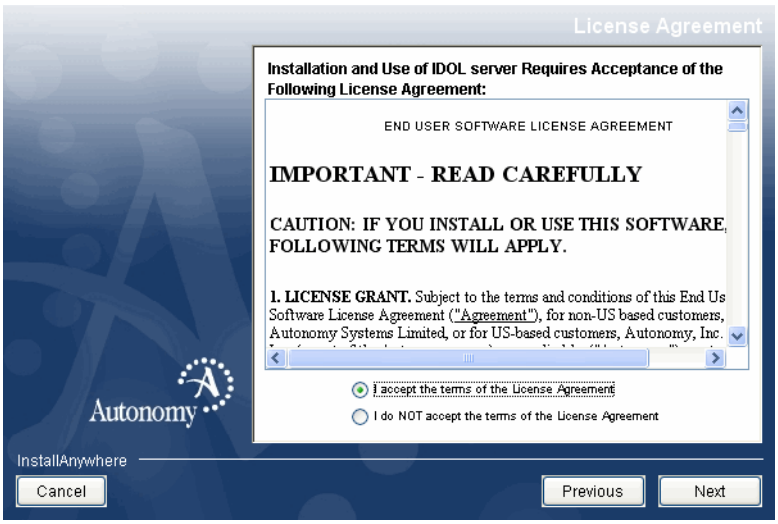
2. Read the text, and click **Next**.

The Windows Admin Privilege dialog box opens.



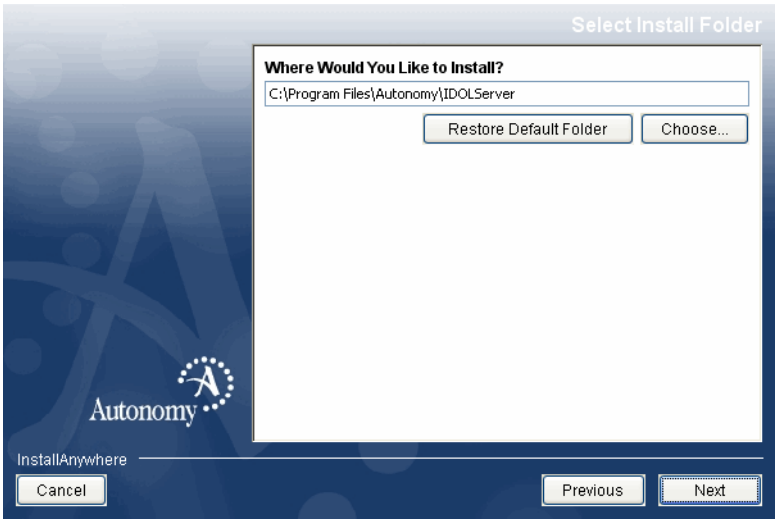
3. Ensure you have administrative privileges, then click **Next**.

   The License Agreement dialog box opens.



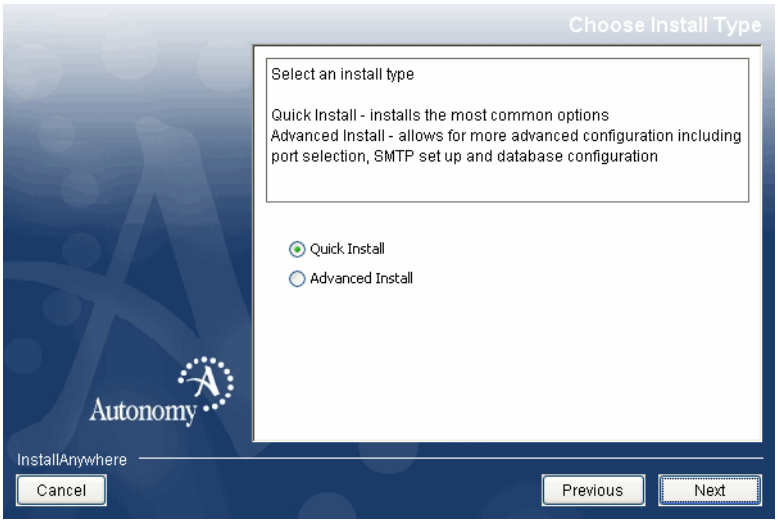
4. Read the license agreement. Select **I accept the terms of the License Agreement**, then click **Next**.

The Select Install Folder dialog box opens.



5. Specify the directory to install IDOL in. By default, it installs the system in `C:\Program Files\Autonomy\IDOLServer` on Windows and `/opt/Autonomy/idolserver` on UNIX platforms. You can click **Choose** to navigate to another location. Click **Next**.

   The Choose Install Type dialog box opens.



6. Select either the **Quick Install** or **Advanced Install** option:

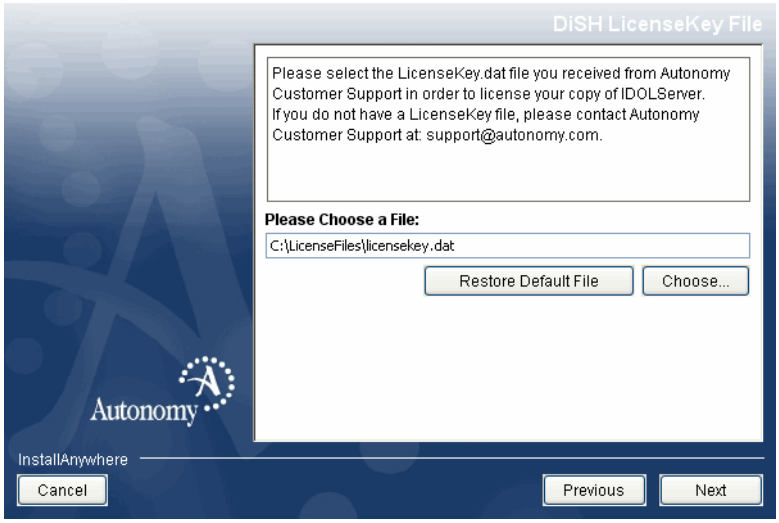
   - **Quick Install option**. To install IDOL server using common configuration settings, select **Quick Install**, then click **Next**. The installer prompts you

only for DiSH license key settings. The installer sets the following defaults for your installation.

Setting Type	Default Setting
Installed components	Installs IDOL server, the DiSH license server, HTTP Connector and File System Connector.
IDOL server host	**127.0.0.1**
IDOL server port	ACI Port: **9000** Index Port: **9001** Service Port: **9002**
DiSH server port	ACI Port: **20000** Service Port: **20002**
Prefix for services and applications	**Autonomy**
SMTP server	IP Address: **smtp.*domain*.com** Port: **25**
HTTP Connector IDOL server details	Host: **127.0.0.1** Index Port: **9001** Database: **News**
HTTP Connector service port	**7001**
File System Connector IDOL server details	Host: **127.0.0.1** Index Port: **9001** Database: **Database**
File System Connector details	ACI Port: **7000** Service Port:**10223**

- **Advanced Install option**. To modify the default settings for each component, select **Advanced Install**, and click **Next**.

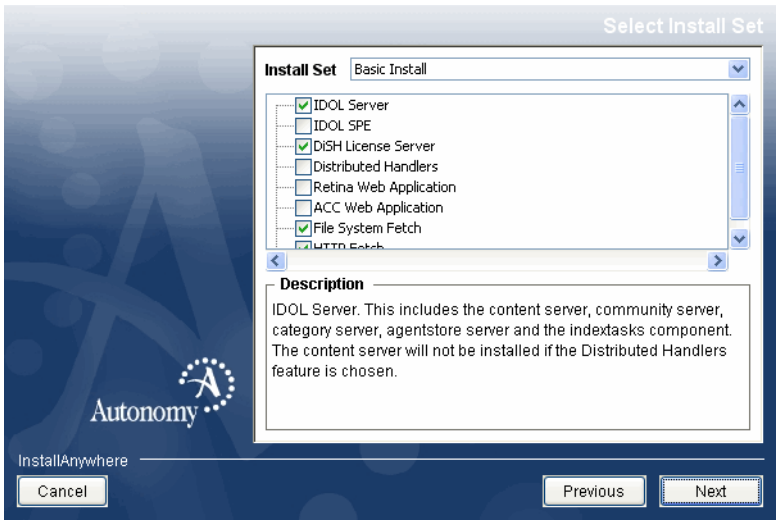
The DiSH License Key File dialog box opens.



7. Type the path or click **Choose** and navigate to the location of your Autonomy license key file (`licensekey.dat`), which you obtained when you purchased IDOL.

   If you selected **Quick Install** in step 6, the Pre-Installation dialog box opens. Proceed to step 20.

   If you selected **Advanced Install** in step 6, the Select Install Set dialog box opens. Proceed to step 8.



8. Select each component to install or select the type of installation from the **Install Set** list. The following install sets are available:

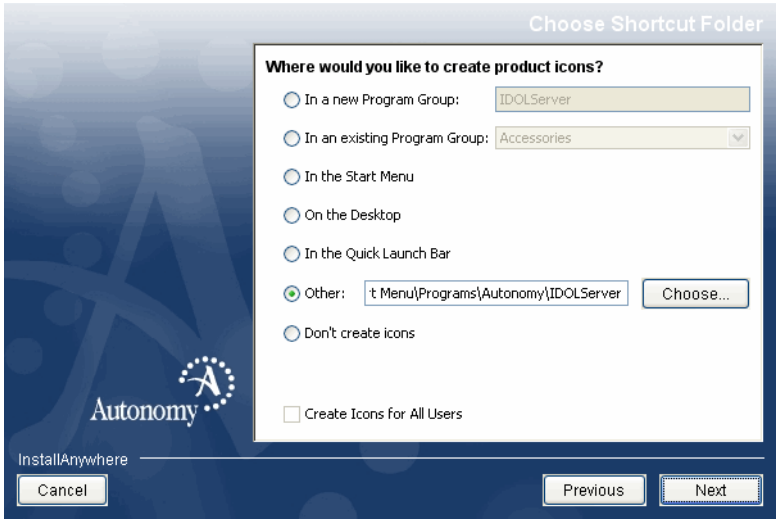
   - **Minimal IDOL Server.**

- **Full Install.**
- **Distributed IDOL Server.**
- **Distributed Handlers Only**.
- **IDOL SPE**.
- **Basic Install**
- **Custom**

If you select an Install set, it selects the appropriate components for you. Click **Next**.

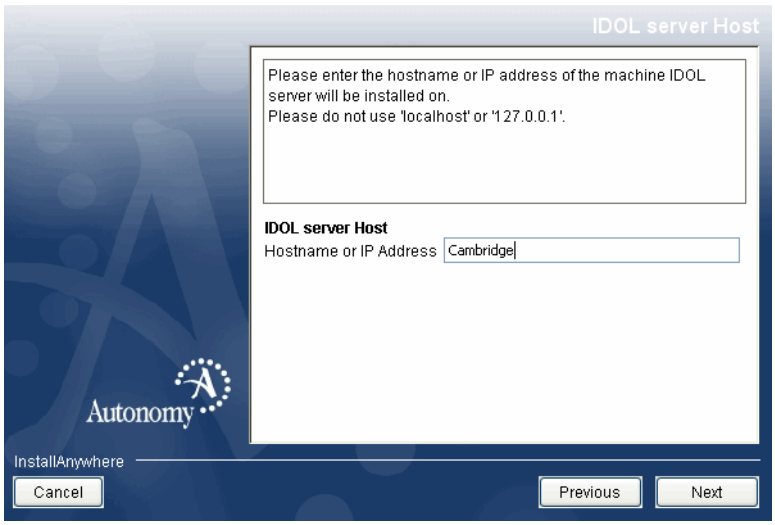
Component	Description
IDOL Server	Installs Autonomy IDOL server. This option includes the Content, Community, Category, Agentstore, and IndexTasks components.
IDOL SPE	Installs Autonomy IDOL SPE. This option includes the Content, Community, Category, Agentstore, and IndexTasks components, with the IDOL SPE configuration file. Refer to the *IDOL SPE Administration Guide* for details. You cannot install both IDOL server and IDOL SPE.
DiSH License Server	Installs an Autonomy DiSH, which is required for licensing.
Distributed Handlers	Installs Distributed Action Handler and Distributed Index Handler. Install this feature as part of a distributed architecture. If you choose the Distributed Handlers option, the Content component is not installed. Refer to the *Distributed Action Handler Administration Guide* and the *Distributed Index Handler Administration Guide.*
Retina Web Application	Installs Autonomy Retina, a Web application that provides a user interface for the functionality IDOL server supplies.
ACC Web Application	Installs the Autonomy Collaborative Classifier, which is an application you can use to manipulate information-classification structures and indexes.
File System Fetch	Installs File System Fetch, an Autonomy connector that automatically aggregates documents from file systems on local or network machines and indexes them into an IDOL server.
HTTP Fetch	Installs HTTP Fetch, an Autonomy connector that automatically aggregates documents from Web site and indexes them into an IDOL server.

The Choose Shortcut Folder dialog box opens.



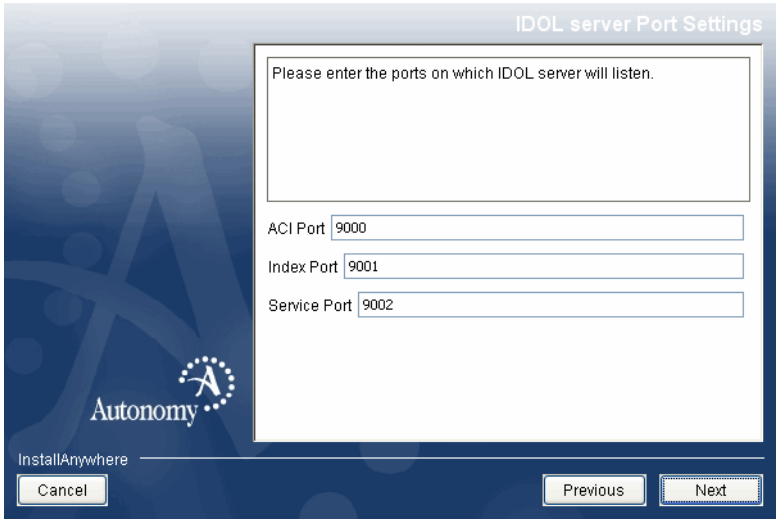
9. Select a location for shortcut icons, and click **Next**.

   The IDOL server Host dialog box opens.



10. Enter the name or IP address of the machine on which you are installing IDOL server, and click **Next**.

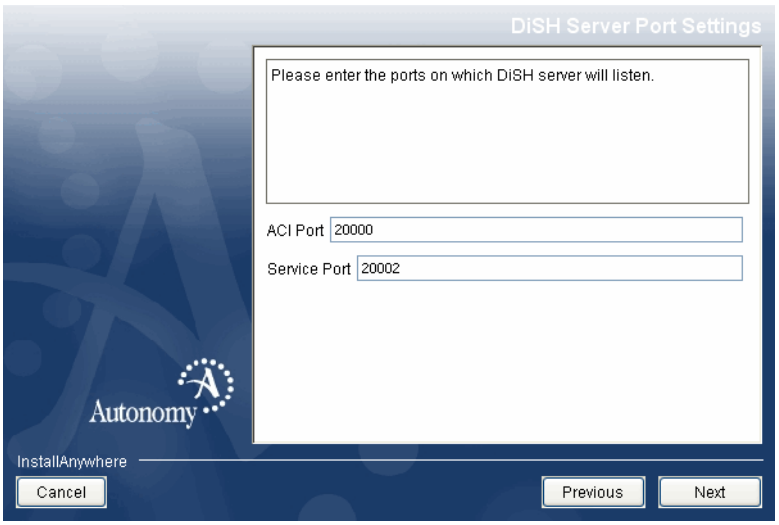
The IDOL server Port Settings dialog box opens.



11. Change the following information or accept the defaults, then click **Next**:

ACI Port	Port that client machines use to send ACI actions to IDOL server.
Index Port	Port that administrative client machines use to index documents into IDOL server (and to administer IDOL server).
Service Port	Port that IDOL server uses for DiSH communication. This port must not be used by any other service.

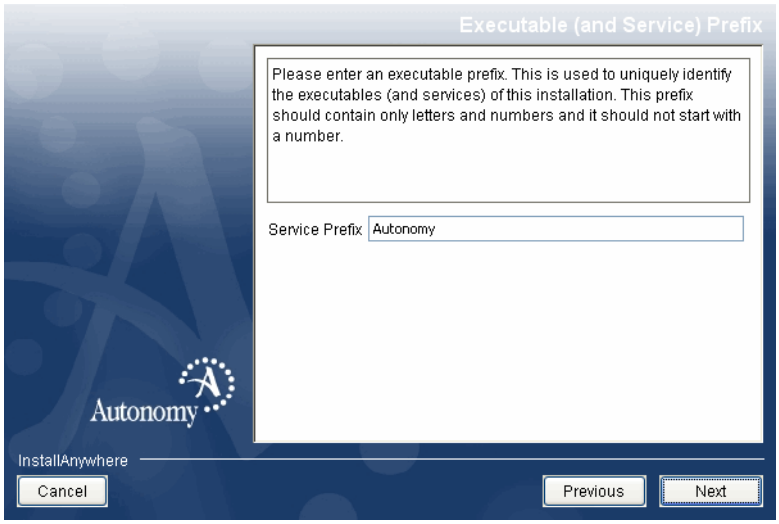
If you selected to install the DiSH server, the DiSH Server Port Settings dialog box opens.



12. Change the following information or accept the defaults, then click **Next**.

ACI Port	Port that client machines use to send ACI actions to the DiSH server.
Service Port	Port by which you send service actions to the DiSH server. This port must not be used by any other service.

The Executable (and Service) Prefix dialog box opens.



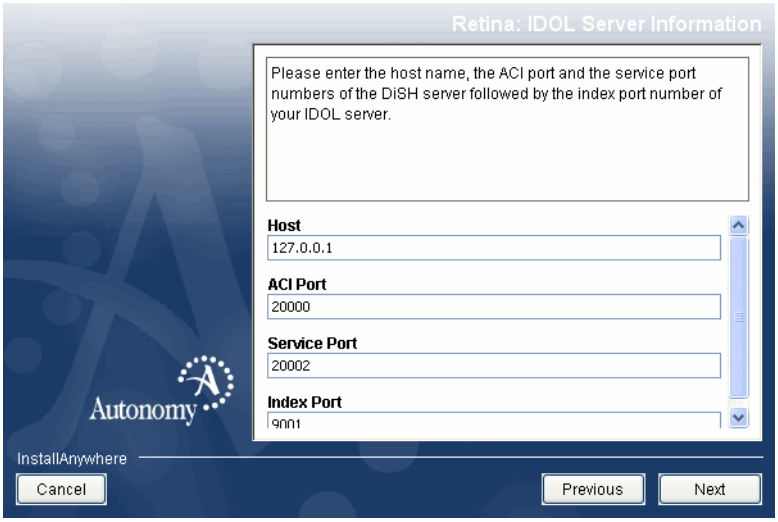
13. Enter a prefix for your installation services and executables to uniquely identify them. The prefix must contain only letters and numbers, and must not start with a number. By default, this option is set to **Autonomy**. Click **Next**.

The SMTP Server Details dialog box opens.

14. Change the following information or accept the defaults, then click **Next**:

IP Address	IP address of your SMTP server host.
Port	Port on which your SMTP server listens for SMTP commands.

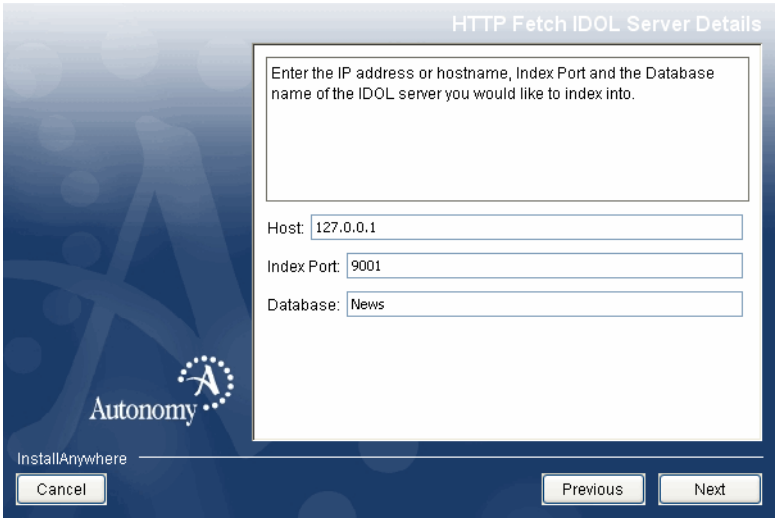
If you selected to install Retina, the Retina: IDOL Server Information dialog box opens.



15. Change the following information or accept the defaults, then click **Next**:

Host	Host name of the DiSH server.
ACI Port	Port to use to send ACI actions to the DiSH server.
Service Port	Port to use to send service actions to the DiSH server. This port must not be used by any other service.
Index Port	Port to use to index documents into IDOL server.

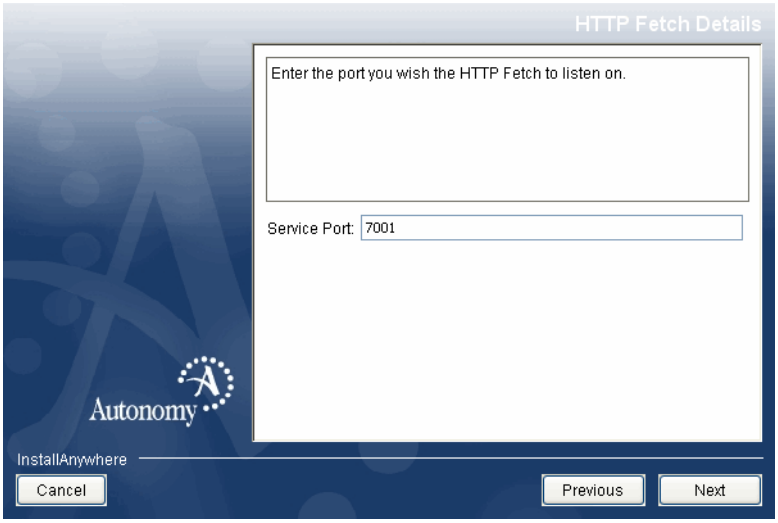
The HTTP Fetch IDOL Server Details dialog box opens.



16. Change the following details or accept the defaults, then click **Next**:

Host	IP address (or computer name) of the machine on which IDOL server runs.
Index Port	Port to use to index documents into IDOL server.
Database	Name of the IDOL server database into which to index the documents that the connector aggregates.

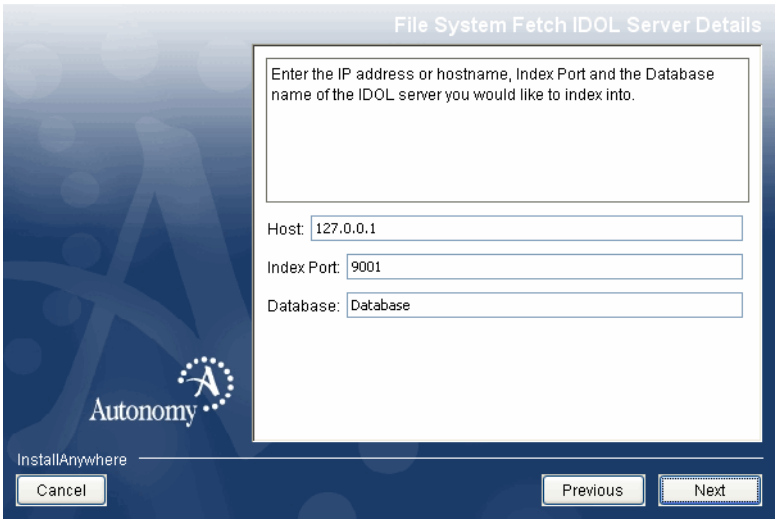
The HTTP Fetch Details dialog box opens.



17. Change the following information or accept the default, then click **Next**.:

Service Port Port by which you send service actions to HTTP connector.

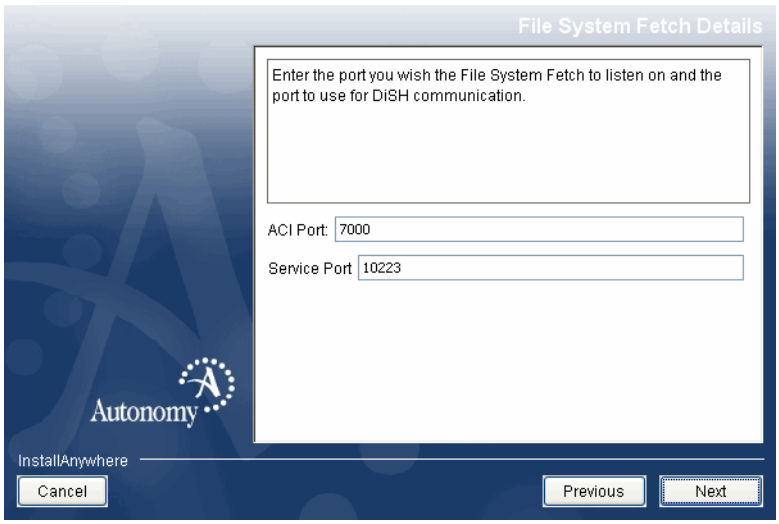
The File System IDOL Server Details dialog box opens.



18. Change the following details or accept the defaults, then click **Next**:

Host	IP address (or name) of the machine on which IDOL server is running.
Index Port	Port to use to index documents into IDOL server.
Database	Name of the IDOL server database into which to index the documents the connector aggregates.

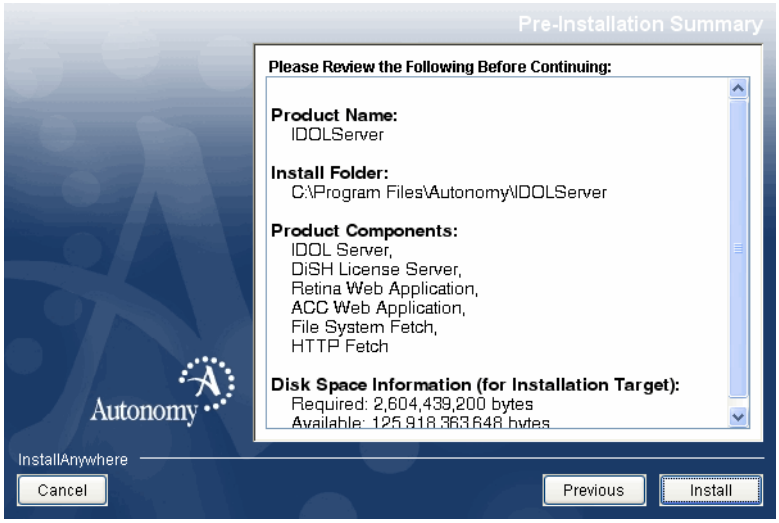
The File System Fetch Details dialog box opens.



19. Change the following information or accept the defaults, then click **Next**:

ACI Port	Port that File System Connector listens on for ACI actions.
Service Port	Port that IDOL server uses to communicate with the license server. This port must not be used by any other service.

The Pre-Installation Summary dialog box opens.



20. Check the settings you have made, and click **Install**.

    The Installing IDOL Server dialog box opens.

    The dialog box indicates the progress of the installation process. If you want to abort the installation process, click **Cancel**.

    The Install Complete dialog box opens. Click **Done** to exit the installation.

Install IDOL Server Standalone on UNIX

The following procedure describes how to install IDOL server standalone using console mode on UNIX platforms.

To install IDOL server standalone on UNIX

1. Open a terminal in the directory in which you have placed the installer, and type the following command:

   `./IDOLServer_`*`VersionNumber`*`_`*`Platform`*`.bin -i console`

   where,

`VersionNumber`	is the version number of the installer
`Platform`	is the name of your UNIX platform

   The console installer launches and displays the Welcome screen.

2. Read the information and then press ENTER.

   The license information is displayed.

3. Read the license information, pressing ENTER to continue through the text. After you finish reading the text, type **Y** to accept the license terms.

   The DiSH LicenseKey File: Directory Name screen is displayed.

4. Type the path to the location of your Autonomy license key file (`licensekey.dat`), which you obtained when you purchased IDOL. Press ENTER.

   The DiSH LicenseKey File: File Name screen is displayed.

5. Type the name of your license file. By default this is `licensekey.dat`. Press ENTER.

   The Select Install Set and Choose Product Features screen is displayed.

6. Each feature that you can install has a number. If the feature has an `[X]`, it is selected to install by default.

   Type a comma-separated list of numbers (there must be no space before or after a comma).

   - If the feature is selected, type its number to deselect it.
   - If the feature is not selected, type its number to select it.

   You can choose from the following features:

	Component	Description
1	IDOL Server	Installs Autonomy IDOL server. This option includes the Content, Community, Category, Agentstore, and IndexTasks components.
2	IDOL SPE	Installs Autonomy IDOL SPE. This option includes the Content, Community, Category, Agentstore, and IndexTasks components, with the IDOL SPE configuration file. Refer to the *IDOL SPE Administration Guide* for details. You cannot install both IDOL server and IDOL SPE.
3	DiSH License Server	Installs an Autonomy DiSH, which is required for licensing.
4	Distributed Handlers	Installs Distributed Action Handler and Distributed Index Handler. Install this feature as part of a distributed architecture. If you choose the Distributed Handlers option, the Content component is not installed. Refer to the *Distributed Action Handler Administration Guide* and the *Distributed Index Handler Administration Guide.*
5	Retina Web Application	Installs Autonomy Retina, a Web application that provides a user interface for the functionality IDOL server supplies.
6	ACC Web Application	Installs the Autonomy Collaborative Classifier, which is an application you can use to manipulate information-classification structures and indexes.
7	File System Fetch	Installs File System Fetch, an Autonomy connector that automatically aggregates documents from file systems on local or network machines and indexes them into an IDOL server.
8	HTTP Fetch	Installs HTTP Fetch, an Autonomy connector that automatically aggregates documents from Web site and indexes them into an IDOL server.

The Select Install Folder screen is displayed.

7. Type the path to the location where you want to install IDOL server, or press ENTER to accept the default path.

   A confirmation is displayed. Type **Y** to confirm the location.

   The IDOL server Host screen is displayed.

8. Type the name or IP address of the machine on which you are installing IDOL server, and press ENTER.

   The IDOL server Port Settings screen is displayed.

9. Provide information for the ports that IDOL server uses.

   a. Type the value for the ACI Port and press ENTER (or press ENTER to accept the default value).

      ACI Port Port that client machines use to send ACI actions to IDOL server.

      The IDOL server Port Settings Index Port screen is displayed.

   b. Type the value for the Index Port and press ENTER (or press ENTER to accept the default value).

      Index Port Port that administrative client machines use to index documents into IDOL server (and to administer IDOL server).

      The IDOL server Port Settings Service Port screen is displayed.

   c. Type the value for the Index Port and press ENTER (or press ENTER to accept the default value).

      Service Port Port that IDOL server uses for DiSH communication. This port must not be used by any other service.

   If you have selected to install the DiSH, the DiSH Port Settings screen is displayed.

10. Provide information for the ports that the DiSH uses.

    a. Type the value for the ACI Port and press ENTER (or press ENTER to accept the default value).

       ACI Port Port that client machines use to send ACI actions to the DiSH server.

       The DiSH Port Settings Service Port screen is displayed.

    b. Type the value for the Service Port and press ENTER (or press ENTER to accept the default value).

Service Port Port by which you send service actions to the DiSH server. This port must not be used by any other service.

If you selected to install Retina, the Retina: IDOL Server Information screen is displayed.

11. Provide information for the IDOL server and DiSH that Retina connects to.

Host Host name of the DiSH server.

ACI Port Port to use to send ACI actions to the DiSH server.

Service Port Port to use to send service actions to the DiSH server. This port must not be used by any other service.

Index Port Port to use to index documents into IDOL server.

If you selected to install HTTP Fetch, the HTTP Fetch IDOL Server Details screen is displayed.

12. Provide information for the IDOL server that HTTP Fetch connects to.

Host IP address (or computer name) of the machine on which IDOL server runs.

Index Port Port to use to index documents into IDOL server.

Database Name of the IDOL server database into which to index the documents that the connector aggregates.

The HTTP Fetch Details screen is displayed.

13. Type the following port value and press ENTER (or press ENTER to accept the default value):

Service Port Port by which you send service actions to HTTP connector.

The File System Fetch IDOL Server Details screen is displayed.

14. Provide information for the IDOL server that the File System Fetch connects to.

Host	IP address (or computer name) of the machine on which IDOL server runs.
Index Port	Port to use to index documents into IDOL server.
Database	Name of the IDOL server database into which to index the documents that the connector aggregates.

The File System Fetch Details screen is displayed.

15. Provide information for the File System Fetch ports.

ACI Port	Port that File System Connector listens on for ACI actions.
Service Port	Port that IDOL server uses to communicate with the license server. This port must not be used by any other service.

The SMTP Server Details screen is displayed.

16. Provide information for your SMTP server.

IP Address	IP address of your SMTP server host.
Port	Port on which your SMTP server listens for SMTP commands.

The Pre-Installation Summary screen is displayed.

17. Check the settings you have made, and press ENTER.

The Ready To Install screen is displayed.

18. Press ENTER to begin installation.

The Installing screen is displayed.

This screen indicates the progress of the installation process.

The Installation Complete screen is displayed.

19. Press ENTER to exit the installation.

Licenses

An Autonomy DiSH server controls the license that enables you to run Autonomy solutions. You must have a running Autonomy DiSH server that resides on a machine with a static, known IP address, MAC address, or host name.

You can revoke licenses at any time if, for example, you want to reallocate them to different clients or if you want to change a client IP address.

IMPORTANT Taking any of the following actions causes the licensed module to become inoperable.

You **must not**:

- change the IP address of the machine on which a licensed module runs (if you use an IP address to lock your license)
- change the service port of a module without first revoking the license
- replace the network card of a client without first revoking the license
- remove the contents of the license and uid directories

All modules produce a `license.log` and a `service.log` file. If a product fails to start, examine the contents of these files before submitting a support ticket.

Display License Information

You can verify which modules you have licensed by sending the following action from a Web browser to the running DiSH server.

`http://DiSHhost:port/action=LicenseInfo`

where,

DiSHhost	is the IP address of the machine where DiSH resides.
port	is the ACI port of DiSH (specified by the `Port` parameter in the DiSH configuration file's `[Server]` section).

In response, DiSH returns the requested license information. This example describes a license to run four instances of IDOL server and one instance of DiSH.

```
<?xml version="1.0" encoding="UTF-8" ?>
<autnresponse xmlns:autn="http://schemas.autonomy.com/aci/">
```

```
  <action>LICENSEINFO</action>
  <response>SUCCESS</response>
  <responsedata>
    <LicenseDiSH>
      <LICENSEINFO>
        <autn:Product>
          <autn:ProductType>DISH</autn:ProductType>
          <autn:TotalSeats>1</autn:TotalSeats>
          <autn:SeatsInUse>0</autn:SeatsInUse>
        </autn:Product>
        <autn:Product>
          <autn:ProductType>IDOLSERVER</autn:ProductType>
          <autn:TotalSeats>4</autn:TotalSeats>
          <autn:SeatsInUse>0</autn:SeatsInUse>
        </autn:Product>
      </LICENSEINFO>
    </LicenseDiSH>
  </responsedata>
</autnresponse>
```

Revoke a Client License

After you set up licensing, you can revoke licenses at any time if, for example, you want to reallocate them to different clients or if you want to change a client IP address.

To revoke a license

1. Stop the Autonomy solution that uses the license.
2. Execute the following action from a command prompt.

InstallDir/ExecutableName[.exe] –revokelicense –configfile *cfgFilename*

This action returns the license to the license server.

You can send the `LicenseInfo` action from a Web browser to the running DiSH server to check for free licenses. In this sample output from the action, one IDOL server license is available for allocation to a client.

```
<autn:Product>
   <autn:ProductType>IDOLSERVER</autn:ProductType>
   <autn:Client>
      <autn:IP>192.123.51.23</autn:IP>
      <autn:ServicePort>1823</autn:ServicePort>
      <autn:IssueDate>1063192283</autn:IssueDate>
      <autn:IssueDateText>10/09/2003 12:11:23</autn:IssueDateText>
   </autn:Client>
      <autn:TotalSeats>2</autn:TotalSeats>
      <autn:SeatsInUse>1</autn:SeatsInUse>
```

```
</autn:Product>
```

Related Topics

- “Display License Information” on page 92

Forcibly Revoke Licenses from Inaccessible Clients

If a client machine becomes inaccessible, you can revoke a license on it by sending the next ACI action to the DiSH server. The `AdminRevokeLicense` action frees the license from the inaccessible machine.

NOTE Call the `AdminRevokeLicense` function only for inaccessible client machines. Otherwise, the module shuts down and becomes inaccessible.

```
http://DiSHhost:port/action=AdminRevokeLicense
&ClientProductType=productType&ClientIP=clientHost &ClientServicePort=clientPort
```

where,

`productType`	is the product type of the Autonomy solution whose license you want to revoke from the inaccessible client.
`clientHost`	is the IP address of the inaccessible client machine.
`clientPort`	is the port by which service actions are sent to the Autonomy module on the inaccessible client (specified by the `ServicePort` parameter in the module configuration file's `[Service]` section).

Troubleshoot License Errors

Table 1 contains explanations for typical licensing-related error messages.

Table 1 License-related error messages

Error message	Explanation
Error: Failed to update license from the license server. Your license cache details do not match the current service configuration. Shutting the service down.	The configuration of the service has been altered. Verify that the service port and IP address have not changed since the service started.
Error: License for ProductName is invalid. Exiting.	The license returned from the DiSH server is invalid. Ensure that the license has not expired.
Error: Failed to connect to license server using cached licensed details.	Cannot communicate with the DiSH server. The product still runs for a limited period, however you should verify whether your DiSH server is still available.
Error: Failed to connect to license server. Error code is SERVICE: *ErrorCode*	Failed to retrieve a license from the DiSH server or from the backup cache. Ensure that your DiSH server can be contacted.
Error: Failed to decrypt license keys. Please contact Autonomy support. Error code is SERVICE:*ErrorCode*	Provide Autonomy Support with the exact error message and your license file.
Error: Failed to update the license from the license server. Shutting down	Failed to retrieve a license from the DiSH server or from the backup cache. Ensure that your DiSH server can be contacted.
Error: Your license keys are invalid. Please contact Autonomy support. Error code is SERVICE:*ErrorCode*	Your license keys appear to be out of sync. Provide Autonomy Support with the exact error message and your license file.
Failed to revoke license: No license to revoke from server.	The DiSH server cannot find a license to revoke.
Failed to revoke license from server LicenseServer Host:*LicenseServerPort***. Error code is** *ErrorCode*	Failed to revoke a license from the DiSH server. Provide Autonomy Support with the exact error message.
Failed to revoke license from server. An instance of this application is already running. Please stop the other instance first.	You cannot revoke a license from a running service. Stop the service and try again.
Failed to revoke license. Error code is SERVICE:*ErrorCode*	Failed to revoke a license from the DiSH server. Provide Autonomy Support with the exact error message.

Table 1 License-related error messages (continued)

Error message	Explanation
Your license keys are invalid. Please contact Autonomy Support. Error code is ACISERVER:*ErrorCode*	Failed to retrieve a license from the DiSH server. Provide Autonomy Support with the exact error message and your license file.
Your product ID does not match the generated ID.	Your installation appears to be out of sync. Forcibly revoke the license from the DiSH server and rename the license and uid directories.
Your product ID does not match this configuration.	The service port for the module or the IP address for the machine appears to have changed. Check your configuration file.

CHAPTER 6

Run IDOL

When you have installed IDOL, you are ready to run it. This section describes the basics of using IDOL and verifying that it is running correctly.

Start and Stop IDOL

This section describes how to start and stop your IDOL service.

Start IDOL

Use this procedure to start IDOL.

NOTE The IDOL installer for Windows starts IDOL automatically when it has been installed. This procedure is for starting a server that is not running.

To start IDOL

1. Start the DiSH licensing server by doing one of the following options:
   - Double-click the *Installation*`DiSH.exe` file in your installation directory (Windows).
   - Use the start script (UNIX).
   - Start the DiSH Service from a system dialog box (Windows).

   To start the DiSH Service from a system dialog box:

   a. Display the Windows **Services** dialog box.

   b. Select the ***Installation*DiSH** service, and click **Start** to start IDOL.

   c. Click **Close** to close the **Services** dialog box.
2. Start IDOL by doing one of the following options:
   - Double-click the *Installation*`IDOL.exe` file in your installation directory (Windows).
   - Use the start script (for UNIX).
   - Start the IDOL Service from a system dialog box (Windows).

   To start the IDOL Service from a system dialog box:

   a. Display the Windows **Services** dialog box.

   b. Select the ***Installation*IDOL** service, and click **Start** to start IDOL.

   c. Click **Close** to close the **Services** dialog box.

Stop IDOL

You can stop IDOL from running using:

- the stop script (for UNIX)
- services (for NT):

  a. Display the Windows **Services** dialog box.

  b. Select the ***Installation*IDOL** service, and click **Stop** to stop IDOL.

  c. Click **Close** to close the **Services** dialog box.
- the service port:

  Send the following action to the IDOL service port (you need to have specified a service port in the IDOL configuration file):

  `http://`*`IDOLhost:servicePort`*`/action=stop`

where *IDOLhost* is the name or IP address of the host on which IDOL is running, and *servicePort* is the IDOL service port (which is specified in the [Service] section of the IDOL configuration file).

NOTE It can take a short while for the Content process to stop after you have sent the stop action to IDOL.

Send Actions to IDOL

You query IDOL by sending actions from your Web browser. The general syntax of these actions is:

http://*IDOLhost*:*port*/action=*action*&*requiredParams*&*optionalParams*

where,

IDOLhost	is the IP address or name of the machine on which IDOL is installed.
port	is the ACI port by which actions are sent to IDOL (set by the Port parameter in the IDOL configuration file [Server] section).
action	is the name of the action you want IDOL to run (for example, Query).
requiredParams	are the parameters that you must supply for the action you are requesting. (Not all actions have required parameters.)
optionalParams	are parameters that you may supply for the action you are requesting. (Not all actions have optional parameters.)

NOTE Separate individual parameters with an ampersand (&).

Verify IDOL Runs Correctly

When you have installed IDOL and are using it, you can run actions to verify that IDOL is running correctly.

GetRequestLog

Send a `GetRequestLog` action to IDOL to return a log of the requests that have been made to it, including:

- the date and time that a request was made.
- the client IP address that made the request.
- the internal thread that handled the action.

For example:

`http://IDOLhost:port/action=GetRequestLog`

For further details on the `GetRequestLog` action, refer to the *IDOL Server Online Help*.

Alternatively, you can display the IDOL Server Online Help, then click the **request log** link in the top right-hand corner. This action displays the help **Log** page, which contains the log of requests that the `GetRequestLog` action returns.

Related Topics

- "Send Actions to IDOL" on page 99

GetLicenseInfo

You can send a `GetLicenseInfo` action to IDOL to return information on your license. This action allows you to check whether your license is valid, which IDOL operations your license includes, and which actions you can run.

For example:

`http://IDOLhost:port/action=GetLicenseInfo`

The following result indicates that your license is valid:

```
- <autn:license>
  <autn:validlicense>false</autn:validlicense>
  </autn:license>
```

The following result indicates that your license includes the IDOL Agent operation:

```
- <autn:section>
  <autn:name>Agent</autn:name>
  </autn:section>
```

The following result indicates that you are permitted to run `Query` actions:

```
- <autn:section>
  <autn:name>Query</autn:name>
  </autn:section>
```

GetStatus

Use the `GetStatus` action to check whether the IDOL service is running.

For example:

`http://IDOLhost:port/action=GetStatus`

GetVersion

Use the `GetVersion` action to check the version number of an IDOL service.

For example:

```
http://IDOLhost:port/action=GetVersion
```

Display Online Help

You display IDOL server Online Help by sending an action from your Web browser. To display help for IDOL configuration parameters and actions, start IDOL, and enter this action from your Web browser:

`http://IDOLhost:port/action=Help`

where,

IDOLhost	is the IP address or name of the machine on which IDOL is installed.
port	is the ACI port by which you send actions to IDOL (set by the `Port` parameter in the IDOL configuration file `[Server]` section).

For example:

```
http://12.3.4.56:9000/action=help
```

This action uses port `9000` to request online help from IDOL, which is located on the local machine.

For IDOL to display help, the help data file (`help.dat`) must be available in the same directory as the service instance.

NOTE You can also view help without starting IDOL. In the IDOL installation directory, open the `help` directory and click `index.html`.

On the initial online help page, click one of the following options in the navigation panel to display help:

Tab	Description
Actions	Describes the actions you can send to IDOL. Actions allow you to query IDOL, and to instruct it to perform a variety of operations.
Configuration Parameters	Describes the parameters that determine how the IDOL operates. Configuration parameters are set in the IDOL configuration file.
Index Actions	Describes the index actions you send to IDOL. Index actions allow you to index content into IDOL, and to administer the IDOL Data index.
Service Actions	Describes service actions. Service actions allow you to return data about the IDOL service, and to control IDOL.

Related Topics

- “Send Actions to IDOL” on page 99

CHAPTER 7

Configure IDOL Server

This section describes how to configure IDOL server by editing the configuration file.

IDOL Server Configuration File

If you are using the standalone version of the IDOL, you configure IDOL by manually editing the IDOL configuration file.

The IDOL configuration file contains the parameters that determine how IDOL operates. You can modify the configuration parameters to customize IDOL. You can find this file in the following directory:

AutonomyDir\idol\IDOLserver*serviceAlias*idolserver.cfg

where,

AutonomyDir	is the Autonomy installation directory for your IDOL.
serviceAlias	is the installation name specified for the IDOL.

Modify Configuration Parameter Values

You modify IDOL configuration parameters by directly editing the parameters in the configuration file. When setting configuration parameter values, you must use ASCII (the only exception to this is the `PropertyFieldCSVs` parameter, which accepts UTF-8).

IMPORTANT You must stop and restart IDOL for new configuration settings to take effect.

The following section describes how to enter parameter values in the configuration file.

Enter Boolean Values

The following settings for Boolean parameters are interchangeable:

```
TRUE = true = ON = on = Y = y = 1
FALSE = false = OFF = off = N = n =0
```

Enter String Values

Some parameters require string values that contain quotation marks. Escape each quotation mark by inserting a backslash before it.

For example:

```
FIELDSTART0="<font face=\"arial\"size=\"+1\"><b>"
```

Here, the beginning and end of the string are indicated by quotation marks, while all quotation marks that are contained in the string are escaped.

If you want to enter a comma-separated list of strings for a parameter, and one of the strings contains a comma, you must indicate the start and the end of this string with quotation marks.

For example:

```
ParameterName=cat,dog,bird,"wing,beak",turtle
```

If any string in a comma-separated list contains quotation marks, you must put this string into quotation marks and escape each quotation mark in the string by inserting a backslash before it.

For example:

```
ParameterName="<font face=\"arial\"size=\"+1\
"><b>",dog,bird,"wing,beak",turtle
```

Use a Unified Configuration

If you use the Distributed Action Handler (DAH) and Distributed Index Handler (DIH), you can install and operate these components either as standalone components or integrated with IDOL in a unified configuration.

In a standalone component setup, you configure DIH and DAH separately to other IDOL components, using the `DIH.cfg` and `DAH.cfg` configuration files. In a unified setup, you install DIH and DAH as part of an IDOL server and set the configuration options in the `IDOLserver.cfg` configuration file.

For more detail about unified and component setups, see "Types of IDOL Systems" on page 37.

When you use the DAH and DIH in a unified IDOL configuration, you add the parameters that normally appear in the component configuration files into the `IDOLserver.cfg` file.

- Add all configuration options that normally appear in the `[Server]` section of the DAH or DIH configuration files to the `[DistributionSettings]` section of the `IDOLserver.cfg` file.
- Add all other sections directly to the IDOL server configuration file from the DAH or DIH configuration files without changing the section or parameter names.

Refer to the *Distributed Action Handler (DAH) Administration Guide* and *Distributed Index Handler (DIH) Administration Guide* for more information.

CHAPTER 8

IDOL Performance

This section discusses methods you can use to improve the performance of your IDOL system.

Performance Overview

IDOL server performance generally refers to the performance of the following operations:

- **Index Data**. Documents in IDX or XML file format are added to IDOL server.
- **Query IDOL server**. Users send natural language, keyword or Boolean queries to IDOL server, which analyzes the concept of the query and returns documents that are conceptually similar.

To optimize IDOL server performance, you must consider the relationship between these operations. The appropriate setup for your system depends on whether your priority is achieving fast query speeds or making new information available to users as quickly as possible.

Schedule Index and Query Operations

IDOL server provides the fastest query responses when it is not indexing, and indexes fastest when it is not being queried.

One way to improve performance for both indexing and querying is to schedule indexing so that it occurs when the query load is at a minimum. For example, if your users send most queries to IDOL server during office hours, you can schedule indexing tasks to run during the night.

When you consider this approach, you must also consider how important up-to-date information is in your system. In some systems, users must be able to access the latest information immediately. For other systems, weekly updates to IDOL server data are enough.

Use a Component Setup

You can improve the performance of IDOL operations by using a component setup. In this setup, you can place each IDOL server component on its own hardware, maximizing the available resources.

Related Topics

- "Types of IDOL Systems" on page 37.

Optimize IDOL Content

There are many ways to optimize the content that you store in IDOL server. You can improve both index and query performance by using a stop word list, and configuring IDOL to not index numbers.

Use a Stop Word List

A stop word list is a list of common words that do not convey much meaning to sentences. For example, in English, words such as *the* and *that* do not add extra meaning in a sentence.

By default, IDOL server does not index any stop words, and does not use them to search documents. This option reduces the number of terms that IDOL must index. It also means that IDOL server retrieves documents based only on the relevant terms in the query.

To optimize index and query performance, ensure that your stop word list includes all common words for your system.

Index Numbers

You can determine how IDOL treats numbers when indexing documents. By default, IDOL does not index any numbers. This option reduces the number of terms that IDOL must index and search.

You can configure IDOL to index numbers, if you want users to be able to search for numeric terms. You can also configure IDOL to index numbers only if they occur in a mixed alphanumeric word (for example, IPv6).

Optimize the Index Process

The speed of the indexing process is often less critical than the speed of the query process. However, when indexing large amounts of data into IDOL, it is important to improve the efficiency of the process where possible. In addition, the way you configure the indexing process can affect the efficiency of the query process.

Delayed Synchronization

The indexing process has the following stages:

1. IDOL creates a representation of the new data in the index cache.
2. IDOL synchronizes the cache with data that it currently contains, and stores the new data on disk, removing it from the index cache.

Delayed synchronization allows you to select how IDOL synchronizes the index cache with the IDOL data. This process is useful in systems where you schedule indexing tasks at times when IDOL is also handling queries.

The `DelayedSync` parameter in the `[Server]` section of IDOL configuration file allows you to specify whether the indexing process uses delayed synchronization.

If the `DelayedSync` parameter is not set, IDOL synchronizes the cache as soon as the representation of data is made. New data is available to the user (as query results) quickly. Use this setting in systems where up-to-date data is the priority.

Synchronization uses resources that IDOL could otherwise use for querying. Delayed synchronization reduces this effect by collecting multiple data representations in the index cache and then synchronizing them all with IDOL data at the same time. This process is useful in systems where query speed is more important than having the most up-to-date data.

NOTE Autonomy recommends using delayed synchronization if you index a large number of small files (files that are smaller than 100MB).

To configure delayed synchronization

1. Open the IDOL server configuration file in a text editor.
2. In the `[Server]` section, set the `DelayedSync` parameter to **`true`**.
3. In the `[Server]` section, set the `MaxSyncDelay` parameter to the maximum length of time between synchronization operations.
4. In the `[IndexCache]` section, set the `IndexCacheMaxSize` parameter to the maximum size that the IDOL server index cache can grow to before synchronizing.

NOTE When the index cache reaches this maximum size, IDOL server synchronizes the cache, but data is not available for searching until the `MaxSyncDelay` has passed, or you send a `DRESYNC` index action.

5. Save and close the configuration file.
6. Restart IDOL server for your changes to take effect.

Distribute IDOL Server Data Across Multiple Disks

If your IDOL data becomes too big to store on one volume (as the stored terms, references, content, and so on increase in size), you can store the data files across multiple disk partitions.

To distribute IDOL data

1. Open the IDOL server configuration file in a text editor.
2. Find the `[Paths]` section, or create one if it does not exist.
3. Set the path parameters to the full paths to the directories where you want to store the corresponding part of the index. The following paths have the largest impact on performance:

Parameter	Description
`DynTermPath`	The directory that contains conceptual data.
`NodeTablePath`	The directory that contains content and structured data.
`StatusPath`	The directory that contains status files, and data that is streamed over the network before indexing.

   For a full list of available parameters, refer to the *IDOL server Online Help*.
4. Save and close the configuration file.

5. Restart IDOL server for your changes to take effect.

For example:

```
[Paths]
DyntermPath=C:\autonomy\idolserver\dynterm
NodetablePath=D:\autonomy\idolserver\nodetable
StatusPath=E:\autonomy\idolserver\status
RefIndexPath=F:\autonomy\idolserver\refindex
MainPath=G:\autonomy\idolserver\main
TagPath=H:\autonomy\idolserver\tagindex
```

Related Topics

- “Display Online Help” on page 101

Optimize Query Operations

Query operations include any action that requests data from IDOL server, for example:

- `Query`
- `Suggest`
- `SuggestOnText`
- `GetQueryTagValues`

Query speed can be particularly important because users require a real-time response to any query. This section describes how to optimize the performance of query operations.

The IDOL server `Query` action allows you to search for documents using a number of restrictions and criteria. The main types of query are:

- `Text`. The query contains a natural language expression, which can also contain Boolean or Proximity operators. IDOL server finds documents that contain concepts that are similar to the given text, and that match any additional operations.
- `FieldText`. The query contains restrictions that specify values that must occur in specified fields. For example, you can restrict a query to documents that contain the term *Dog* in the *Animal* field.

Simple `Text` queries are generally quicker than `FieldText` queries.

Optimize IDOL Fields

You can optimize `FieldText` queries by configuring field processes. These processes define how IDOL stores fields, to optimize the retrieval of the data that the field contains. For example, if the field contains a number, configure the field as a numeric field to allow IDOL server to quickly process and retrieve values for that field.

Autonomy recommends that you use field processes to identify types of fields appropriately. For details of how to set up field processes, refer to the *IDOL Server Administration Guide*.

Table 2 lists the field properties that optimize `FieldText` specifiers, and the specifiers that they optimize.

Table 2 Optimized Field Specifiers and Field Properties

Field Property	Optimized Field Specifiers	
`MatchType`	`BIASVAL`	`NOTMATCH`
	`EMPTY`	`NOTSTRING`
	`EXISTS`	`NOTWILD`
	`MATCH`	`STRING`
	`MATCHALL`	`WILD`
`NumericType`	`BIAS`	`EXISTS`
	`EMPTY`	`GREATER`
	`EQUAL`	`LESS`
	`EQUALALL`	`NOTEQUAL`
	`EQUALCOVER`	`NRANGE`
`NumericDateType`	`BIASDATE`	`LTNOW`
	`GTNOW`	`RANGE`
`CountType`	`EQUALCOVER`	`MATCHCOVER`
`SortType`	`ARANGE`	
`BitFieldType`	`BITSET`	

Index Fields

IDOL server processes index fields linguistically. It removes stop words, and stems each term before storing the terms. This process allows IDOL to return documents that match a conceptual query or contain keyword search terms.

Define fields that contain document content as index fields. For example, the document title and body. When performing `Text` queries, IDOL checks Index fields for matching terms and concepts.

Autonomy recommends that you do not store URLs or content that you are unlikely to query as Index fields. Autonomy also recommends that you use Match fields, rather than index fields) for fields where you query only the whole value of the field.

Match Fields

IDOL server stores the value of match fields in a fast look-up structure in memory. You can define fields as match field when you frequently want to retrieve documents using the whole value of this field. For example, you might define the *Author* field of a document as a Match field so that users can search for the author.

Numeric Fields

IDOL server stores the value of numeric fields in a fast look-up structure in memory. You can use numeric fields to allow users to search for values or ranges of values. For example, you might make a *Price* field a numeric field so that users can restrict results to products within a price range.

Numeric Date Fields

IDOL server stores the value of numeric date fields in a fast look-up table in memory. You can use numeric date fields to allow users to search for a date or range of dates. For example, you might make a *Date* field a numeric date field so that users can restrict results to those between two dates.

Count Fields

IDOL server stores the number of occurrences of count fields. You can use count fields to allow users to search for more than one string in a given field.

Sort Fields

You can use sort fields to allow users to search for values in an alphabetical range. For example, you might make the *Author* field a sort field so that users can search for books alphabetically by author.

Sort fields also optimize the `Sort` action parameter when sorting on a the value of a field.

Bit Fields

Typically, bit fields contain information about the sets that a document belongs to. You can use bit fields to allow users to search for documents that only occur in a particular set. For example, you might make a *Workbook* field a bit field so that users can search for documents that occur in a particular workbook.

Parametric Fields

You can use parametric fields to allow users to restrict a search to certain parameters. For example, you might make a *Model* field a parametric field so that users can restrict a search to products of a particular model.

You might also want to make a field both `ParametricType` and `MatchType` to allow you to list the available values and then restrict by those values.

NOTE You can configure a field as simultaneously `MatchType` and `ParametricType` only if you set `ParametricNumericMapping` to **`false`**.

Field Check Fields

IDOL server stores a checksum hash of the value of a field check fields. This option allows especially fast retrieval when you restrict results to the exact value of the field. For example, you might make a *Category* field a field check field so that you can restrict a search to a particular category.

Field Check fields are also useful for the `Combine` operation. In a `Query` action, you can combine results that have the same value of the field check field. IDOL then returns only one document for this field check value.

NOTE Each document that you index into IDOL server must contain only one `FieldCheckType` field.

Hardware Considerations

To maximize IDOL performance, ensure that IDOL server has adequate resources assigned:

- Ensure that IDOL server has enough memory.
- Monitor disk usage to check input and output.

Related Topics

- “Recommended Hardware Specifications” on page 72

Optimize Distributed Systems

Distributed IDOL systems use a Distributed Action Handler (DAH) and Distributed Index Handler (DIH) to distribute actions and index actions respectively. In these systems, you can use one of two modes:

- **Mirror mode** uses several identical IDOL servers to process client queries. You can use this system when you have large numbers of queries, to minimize the query load on individual child servers.
- **Non-mirror mode** distributes all data across several IDOL servers, which can improve the speed of indexing for the system, and increases the amount of data that IDOL server contains. You can use this system when you have a large amount of data to index, or to minimize the index load on individual servers. As servers might respond more slowly while they are indexing data, this method can also improve query response speed.

In both modes, you can consider other modes in the DIH and DAH to distribute ACI and index actions most efficiently.

Related Topics

- "Types of IDOL Systems" on page 37

DAH Performance

The DAH distributes actions between child servers. It also performs some additional processing on actions. For example, in mirror mode it can apply templates, and recognize and distribute administrative actions to all child servers.

Depending on your system, you can use DAH modes to reduce the amount of additional processing that DAH performs, which can improve the performance of the system.

Fast Mirror Mode

In fast mirror mode, DAH distributes actions between mirrored child servers but does not perform any additional processing to actions. You can use this option to increase the performance of your IDOL system in mirror mode when you use DAH only for requesting information from child servers.

When running the DAH in fast mirror mode:

- You cannot send state-changing actions to the DAH (for example, actions including the `Delete`, `State` and `StoreState` action parameters).
- You cannot request additional processing on actions (for example, actions that use the `Template`, `EncryptResponse` or `Output` action parameters).

- You cannot use document IDs in action parameters.
- You cannot use distributed actions (actions to send to all child servers) or asynchronous actions.

Simple Combinator Mode

In simple combinator mode, DAH does not use virtual databases to distribute and combine actions in non-mirror mode. DAH forwards the `DatabaseMatch` action parameter to child servers, and combines results from all child servers.

This option reduces the amount of processing that DAH must perform to distribute and combine virtual database.

DIH Performance

The DIH distributes index actions to its child servers.

In standard mirror and non-mirror mode, DIH forwards the IDX or XML index file to all its child servers, with instructions to the child servers about which documents it must index. You can use different modes in the DIH to reduce the amount of data that the DIH, or the child servers, must process.

Preserve DREADD

The DIH configuration parameter `PreserveDREADD` allows it to distribute index actions more quickly.

By default, DIH accepts an IDX or XML file input and converts the original `DREADD` index action into a `DREADDDATA` index action, which contains all the IDX or XML data. If you set `PreserveDREADD` to **`true`** in the configuration file, DIH forwards the `DREADD` action directly. This option reduces the network load.

To use this parameter, each child IDOL server must be able to access the original IDX or XML file.

You can use `PreserveDREADD` only in mirror mode, simple non-mirror mode or `DistributeOnBatch` mode. For example, you cannot use `PreserveDREADD` if you have set `DistributeByReference` to **`true`**. In advanced distribution modes, DIH must parse the IDX or XML file to distribute the documents correctly.

Distribute On Batch

In distribute on batch mode, DIH indexes data to alternating child servers. This method is an efficient way of distributing the data to child servers in non-mirror mode. You can use this method if:

- you index a steady stream of IDX or XML files, with similar numbers of documents. This mode might not be appropriate if you infrequently index very large IDX or XML files.
- you do not need to remove duplicate documents.

Advanced Distribution Modes

DIH has several distribution modes that determine how it distributes documents between child servers in non-mirror mode. Unlike standard non-mirror mode, these methods send each child server only the documents that it must index. This option reduces the amount of data that each child server receives.

NOTE If you use these methods, the number of DIH child servers is fixed. To add child servers, you must clear all engines, add the new servers, and then re-index using the DIH.

- **Distribute by reference**. DIH distributes documents to child servers based on the document reference. This method ensures that you can prevent duplicate documents that have the same reference.
- **Distribute by date**. DIH distributes documents to child servers according to the document date. You configure a date range for each child server.
- **Distribute by fields or field values**. DIH distributes documents to child servers based on the value of a certain field.
  - In distribute by fields mode, DIH sends documents with the same value of the specified field to the same child server. DIH determines which child server receives a given field value.
  - In distribute by field values mode, you can specify which child servers receive documents that contain certain field values.

Round Robin Mode

The Round-Robin indexing mode rotates indexing between several child IDOL servers, so that indexing and querying are performed by different child servers at different times.

IDOL Server provides the fastest queries when it is not indexing, and indexes fastest when it is not being queried. When you configure indexing for round-robin mode, DIH suspends query handling for a specific child server. It then indexes only into this child server, which has optimal indexing performance.

After the specified time period, DIH starts indexing to a different child engine and makes the previous engine available for querying.

CHAPTER 9

Tutorial: Index Data into IDOL

This chapter describes how to configure the IDOL server index and index data.

Overview

The IDOL data index is a representation of all the data that you can search for in IDOL server. When IDOL server indexes data, it processes the terms and fields in the new documents. This process allows it to retrieve the documents quickly when you send queries to IDOL server.

Before you index documents into IDOL, you must configure the data index to process documents according to your requirements. You can then create and index documents, either by using an Autonomy connector, or by creating an IDX or XML file and indexing them manually.

Related Topics

- "Configure the IDOL Server Index" on page 120
- "Create Documents" on page 131
- "Index Documents" on page 131

Configure the IDOL Server Index

Before you begin to index documents into IDOL server, you must configure the data index to process the documents correctly.

You can configure:

- IDOL databases to store your data.
- field processes to correctly identify and process the fields in the documents you index.
- language types to correctly process different languages and encodings.

Related Topicst

- "Configure Databases" on page 120
- "Configure Fields" on page 121
- "Configure Languages" on page 126
- "Configure Index Tasks" on page 129

Configure Databases

IDOL server can store documents in different databases. You can configure up to 65534 databases in IDOL server.

Indexing data into different databases allows you to restrict queries to data in a particular database. This can improve the performance of your queries.

You can add IDOL server databases by editing the configuration file or by sending an index action. For details of adding a database by editing the configuration file, refer to the *IDOL Server Administration Guide* and the *IDOL Server Online Help*.

To add IDOL server databases

- Send a DRECREATEDBASE action (case sensitive) from your Web browser:

  `http://IDOLhost:indexPort/DRECREATEDBASE?DREdbname=databaseName`

where,

IDOLhost	is the IP address or host name of the machine on which IDOL is installed.
indexPort	is the IDOL index port (specified as `IndexPort` in the IDOL configuration file `[Server]` section).
databaseName	is the name of the database that you want to create.

For example:

```
http://12.3.4.56:20001/DRECREATEDBASE?DREdbname=Archive
```

This action uses port 20001 to create a new database named `Archive` on the IDOL machine with the IP address `12.3.4.56`.

You can also add other index action parameters to this action, for example to make the new database internal. For details, refer to the *IDOL Server Online Help*.

Configure Fields

The documents in IDOL server store data in fields. These fields can contain many different types of information. For example, one field might contain the bulk of the document content, and another field might contain only the name of the author.

When you index data into IDOL server, it is important to set up field processes so that IDOL server treats each field correctly.

Choose Field Properties

Before you set up fields in IDOL server, you must consider the fields in the documents that you want to index.

There are a large number of different field properties that you can apply to IDOL fields. Fields can contain information about the document, or values that you want to retrieve in queries.

For a complete list of the types of fields that you can store in IDOL server, refer to the *IDOL Server Administration Guide* and *IDOL Server Online Help*.

Fields that contain information about the document

`ReferenceType`	Reference fields contain a unique document reference, which you can use to remove duplicate documents, and to retrieve a specific document. Each document must contain at least one reference field. IDOL server also uses the reference field to populate the `autn:reference` metadata field.
`DateType`	Date fields contain the document date, which IDOL server uses to populate the `autn:date` metadata field. If a document does not have a date field, IDOL uses the date that the document was indexed.
`TitleType`	Title fields contain the document title, which IDOL server uses to populate the `autn:title` metadata field.
`DatabaseType`	Database fields contain the IDOL database that IDOL server must index the document into. If the document does not contain a database field, you must specify the database in the index action.
`LanguageType`	Language fields contain the language type of the document, which IDOL server uses to find the appropriate language configuration. IDOL server also uses this field to populate the `autn:languagetype` metadata field.
`SecurityType`	Security fields contain the security type of the document, which IDOL server uses to index the document according to the specified security configuration.
`ACLType`	ACL fields contain document access control lists (ACLs), which determine the access restrictions for that document.
`ExpireDateType`	Expire date fields contain the date that the document expires. On this date, IDOL processes the document according to the expiry rules for the database, for example to delete the document or move it to an archive database. IDOL server also uses this field to populate the `autn:expiredate` metadata field.
`SectionBreakType`	Section break fields contain the section number for a document section, which IDOL server uses to populate the `autn:section` metadata field.
`ParametricType`	Parametric fields contain values that you want to use to restrict queries. In a parametric query, you can return all values that occur in a certain parametric field in all documents.

`ParametricRangeType`	Parametric fields contain values that you want IDOL server to dynamically analyze to generate numeric ranges in parametric queries.

Fields that contain document content

`Index`	Index fields contain document content. IDOL processes these fields linguistically and stores terms to allow fast data retrieval. Typically, you store the document content and title field as index fields. Do not store data as index fields if: ■ you do not need to query the field contents. ■ you query only the whole value of the fields. It is more efficient to store these fields as `MatchType` fields and query the fields using a `FieldText` specifier such as `MATCH`.
`SourceType`	Source fields contain document content that IDOL server uses to create document summaries and suggest conceptually similar documents. If you do not configure source fields, IDOL server uses the configured Index fields.
`LangDetectType`	Language Detection fields contain document content that IDOL server uses to automatically detect the document language. If you do not configure a language detection fields, IDOL server uses the configured source fields.
`HighlightType`	Highlight fields contain document content that IDOL server can highlight when you send the `Highlight` action (or a query action with the `Highlight` parameter set to **`true`**).

Field properties that optimize FieldText query operators

`MatchType`	See “Match Fields” on page 113
`NumericType`	See “Numeric Fields” on page 113
`NumericDateType`	See “Numeric Date Fields” on page 113
`SortType`	See “Sort Fields” on page 113

Field properties that determine how to display fields

`HiddenType`	Hidden fields do not appear
`PrintType`	Print fields are displayed in a `Query` action when the `Print` parameter is set to **`Fields`** (the default value).

Set Up Field Processes

To identify properties for different fields, you must define field processes. In a field process, you define:

- the set of fields that the field process applies to.
- the property that applies to this process.

You can have multiple field processes that share the same property. You must then create a configuration section for each property that you use, and define the field properties.

NOTE Use the following formats to identify fields:

- `/FieldName` to match root-level fields
- `*/FieldName` to match all fields except root-level
- */Path/FieldName* to match fields that the specified path points to.

Field names must not contain spaces nor accents, and they must not start with a number. For IDX documents, IDOL server converts these text elements to underscores (_) when it indexes the fields. You must also change any queries that reference these field names to use the modified field name.

To apply processes to fields or documents that contain specific fields

1. Open the IDOL server configuration file in a text editor.
2. In the `[FieldProcessing]` section, list the processes to apply to fields. For example:

```
[FieldProcessing]
0=IndexFields
1=DateFields
2=DatabaseFields
3=SetReferenceFields
```

3. Create a configuration section for each process listed.
   - Set `PropertyFieldCSVs` to a comma-separated list of fields that this process applies to.
   - Set `Property` to the name of the property configuration section.

NOTE Each property must have a unique configuration section name.

- (Optional) Set `PropertyMatch` to a comma-separated list of values that the field must have to be processed. For example, you can use this parameter to set up a process that identifies security or language fields.
- (Optional) Set `PropertyNegativeFieldCSVs` to a comma-separated list of fields that this process does not apply to. For example, if you use a wildcard value in `PropertyFieldCSVs`, you can define any exceptions in `PropertyNegativeFieldCSVs`.

For example:

```
[IndexFields]
Property=Index
PropertyFieldCSVs=*/DRECONTENT,*/DRETITLE

[DateFields]
Property=Date
PropertyFieldCSVs=*/DREDATE,*/harvest_time

[DatabaseFields]
Property=Database
PropertyFieldCSVs=*/DREDBNAME,*/Database*
PropertyNegativeFieldCSVs=*/DatabaseNumber

[SetReferenceFields]
Property=Reference
PropertyFieldCSVs=*/DREREFERENCE,*/DRETITLE
```

4. Create a section for each property. Specify appropriate configuration parameters for each property. These configuration parameters define the processes that apply to all the fields (or all documents that contain the fields) that you previously associated with the processes.

   For example:

```
[Index]
Index=TRUE

[Date]
DateType=TRUE

[Database]
DatabaseType=TRUE

[Reference]
ReferenceType=TRUE
TrimSpaces=TRUE
```

NOTE For some properties, you must re-index your content if you want to change the property value after you have indexed content into IDOL server.

For details, refer to the *IDOL Server Online Help*.

5. Save and close the configuration file.
6. Restart IDOL server for your changes to take effect.

Related Topics

- "Display Online Help" on page 44

Configure Languages

IDOL server can process documents in multiple languages and encodings. For each language that you want to use, you must define the language types in the IDOL server configuration file. You must also configure IDOL server to classify documents, either by automatically detecting the language and encoding, or by reading the language type from a field.

Define Language Types

To run IDOL in multiple languages, specify the language types you want IDOL to process. A language type is a combination of the language and encoding.

NOTE You must specify languages and language types before you index data into IDOL.

To define language types

1. Open the IDOL configuration file in a text editor.
2. Find the `[LanguageTypes]` section. List the languages that you want IDOL to process. You must use ASCII characters to specify the language names.

   For example:

   ```
   [LanguageTypes]
   0=English
   1=Afrikaans
   2=General
   ```

3. For each language, create a configuration section that matches the name you defined in the `[LanguageTypes]` section.
4. In this section, specify appropriate settings that determine how IDOL handles this language. For details on the configuration parameters you can use, refer to the *IDOL Server Online Help*.
5. For each section, set the `Encodings` parameter to a list of the encodings and corresponding language types used by the language. List each encoding and language in the format *`encoding`*`:`*`languagetype`*. Separate multiple language types with commas.

For example:

```
[english]
Encodings=ASCII:englishASCII,UTF8:englishUTF8
Stoplist=english.dat
IndexNumbers=1

[afrikaans]
Encodings=ASCII:afrikaansASCII,UTF8:afrikaansUTF8
IndexNumbers=1

[general]
Encodings=UTF8:generalUTF8,ASCII:generalASCII,CYRILLIC:generalC
YRILLIC
IndexNumbers=1
```

6. Save and close the configuration file.
7. Restart IDOL server for your changes to take effect.

   You can now configure IDOL to associate the language types you defined with documents.

Associate Language Types with Documents

After you define all the language types you want IDOL to process, set up a field process that allows IDOL to associate these language types with documents.

The way you configure the field process depends on the documents that you want to index:

- If all the documents contains a field that exactly specifies the language type, configure a field process to define this field as a `LanguageType` field. The language types that appear in this field must exactly match the language types that you define in the `[LanguageTypes]` configuration section. See "Configure Fields" on page 121.
- If the documents contain a field that specifies the language, but does not exactly specify the language type, you can configure field processes to detect the language from this field data.

To configure IDOL to detect the language type from field data

1. Open the IDOL server configuration file in a text editor.
2. In the `[FieldProcessing]` section, define a field process for each language that you want to detect.

   For example:

```
[FieldProcessing]
0=DetectArabic
1=DetectEnglish
```

```
2=DetectFrench
```

3. Create a configuration section with the same name as each of the field processes you defined in the `[FieldProcessing]` section.
4. In this section:
   - Set `Property` to the name of the property for the specified language type.
   - Set `PropertyFieldCSVs` to a comma-separated list of fields that can contain the language data.
   - Set `PropertyMatch` to a comma-separated list of values that this field might contain to identify the specified language type.

   For example:

   ```
   [DetectArabic]
   Property=SetArabicProperty
   PropertyFieldCSVs=*/DRELANGUAGETYPE,*/LANG
   PropertyMatch=arabic

   [DetectEnglish]
   Property=SetEnglishProperty
   PropertyFieldCSVs=*/DRELANGUAGETYPE,*/LANG
   PropertyMatch=*eng*,uk,*british

   [DetectFrench]
   Property=SetFrenchProperty
   PropertyFieldCSVs=*/DRELANGUAGETYPE,*/LANG
   PropertyMatch=*fre*,fran*
   ```
5. Create a configuration section for each property that you define in the field processing sections.
6. In the property configuration section, set the `LanguageType` parameter to the language type to use to define documents that match this property (that is, that contain a field with a matching value for the field process). This language type must match one of the language types you configure in the `[LanguageTypes]` configuration section.

   For example:

   ```
   [SetArabicProperty]
   LanguageType=Arabic

   [SetEnglishProperty]
   LanguageType=English

   [SetFrenchProperty]
   LanguageType=French
   ```
7. Save and close the configuration file.
8. Restart IDOL server for your changes to take effect.

Configure Index Tasks

IDOL can process documents that it receives (for example from an Autonomy connector) before it indexes them. You can configure one or more pre-indexing tasks to manipulate IDOL data before you index it. For example, you can perform an ACI action to categorize the incoming document. You can then add the category information to an IDOL field.

Available Tasks

IDOL can perform the following basic pre-indexing tasks:

ACI	to run an action.
Alert	to alert users to new documents that IDOL receives if these documents are similar to agents that the users own.
Cat	to categorize incoming documents.
Eduction	to extract information embedded in unstructured data and store it in structured fields (refer to the *Eduction User Guide*).
FieldOp	to modify the content of fields, or add new fields to documents.
FileWriter	to write incoming documents to disk.
HTTP	to send an HTTP call out to a Web interface (for example, you can connect to a third-party Web application to store data on a legacy SQL database).
OCR	to evaluate the quality of files produced as a result of optical character recognition, and to route the files according to their quality.

In addition, IDOL can perform the following related tasks:

Route	to specify conditions that determine which task IDOL performs next. Use this task when you want to use different tasks for different kinds of documents.
Index	to index data that was processed by tasks into IDOL.
Add2Replace	to convert a `DREADD` index action into a `DREREPLACE` index action.
Lua	to run tasks written in the embedded scripting language Lua.

For more information about the available index tasks, refer to the *IDOL Server Administration Guide*, and the *IDOL Server Online Help*.

Set up Index Tasks

Pre-indexing tasks use the Index Tasks IDOL server component. You can configure index tasks either in a unified IDOL server configuration file, or in the standalone Index Tasks configuration file.

To set up pre-indexing tasks

1. Open the IDOL configuration file in a text editor.
2. Create an `[IndexTasks]` configuration section.
3. Set `StartTask` to the name of the first task that you want IDOL to run on incoming data. For example:

   ```
   StartTask=MyACITask
   ```

4. Create a configuration section with the same name as this task. For example:

   ```
   [MyACITask]
   ```

5. Create a configuration section for all the tasks that you want IDOL server to run. Each task must have a unique name.
6. For each task:
   a. Set the `Module` parameter to the type of task. For example:

      ```
      Module=ACI
      ```

      For a full list of available modules, refer to the *IDOL Server Online Help*.
   b. Add any optional and required parameters for the task. Refer to the *IDOL Server Online Help* for a complete list of available parameters.
   c. Set `NextTask` to the name of the task that IDOL must run after the current task. You do not need to add the `NextTask` parameter for the last task in the sequence. For example:

      ```
      NextTask=MyOCRTask
      ```

   d. Set `OnFailureTask` to the name of the task that IDOL must run next if the current task fails. For example:

      ```
      OnFailureTask=MyFileWriterTask
      ```

NOTE Remember to configure any tasks that you use in the `OnFailureTask` configuration parameter.

7. When you have finished adding tasks, save and close the configuration file.
8. Restart IDOL server for your changes to take effect.

Create Documents

If your data is not in XML format, you must first import it. You can import data using one of two methods.

- **Import with a connector**. The Autonomy connectors (for example, File System Connector, HTTP Connector, Database Connector, and so on) retrieve documents from different repositories and import them into IDX or XML file format. For details of how to configure a connector to import data and index it into IDOL, refer to the appropriate Connector Administration Guide.
- **Import manually**. You can create a text file in either XML or IDX format, which contains the information that you want to index into your IDOL in specific IDOL fields. For details of how to create a manual IDX document, refer to the *IDOL Server Administration Guide*.

Index Documents

You can index only files in XML or IDX format into IDOL. If the data that you want to index into IDOL is in XML format, you can index it directly into IDOL, without having to first *import* it (convert its content and metadata to IDX).

Autonomy connectors use the `DREADD` and `DREADDDATA` index actions to index data into IDOL. You also can use these actions to directly index data into IDOL.

Use DREADD to Index IDX and XML Files Directly

The `DREADD` index action (case sensitive) directly indexes an IDX or XML file that is located on the same machine as the IDOL. For example:

`http://IDOLhost:indexPort/DREADD?filename&optionalParams`

where,

`IDOLhost`	is the IP address or host name of the machine on which IDOL is installed.
`indexPort`	is the indexing port of the IDOL (specified in the `IndexPort` parameter in the `[Server]` section of the IDOL configuration file.
`filename`	is the filename or location of the IDX or XML file to index.
`optionalParams`	are any optional index action parameters that you want to add. **NOTE** you must set the `DREDbName` parameter if your IDOL configuration does not contain a field process to read the database from a field.

For a full list of available parameters for the DREADD action, refer to the *IDOL Server Online Help*.

Use DREADDDATA to Index Data Over a Socket

The DREADDDATA index action (case sensitive) allows you to directly index data over a socket into IDOL.

NOTE This index action requires a POST request method. For more information, refer to the *IDOL Server Administration Guide*.

For example:

```
http://IDOLhost:indexPort/
DREADDDATA?OoptionalParamsData#DREENDDATAKillDuplicatesOption\n\n
```

where,

IDOLhost	is the IP address or host name of the machine on which IDOL is installed.
indexPort	is the indexing port of the IDOL (specified in the IndexPort parameter in the [Server] section of the IDOL configuration file.
optionalParams	are any optional index action parameters. DREADDDATA accepts the same optional parameters as the DREADD index action, except for *KillDuplicatesOption*.
Data	is the content of the IDX or XML document to index. You must add #DREENDDATA to the end of your data. You can use gzipped documents, but #DREENDDATA must not be compressed. This parameter is required.
KillDuplicatesOption	is the option to use for removing duplicate documents.

For details of all the parameters available in the DREADDDATA index action, refer to the *IDOL Server Online Help*.

Glossary

A

ACI (Autonomy Content Infrastructure)	A technology layer that automates operations on unstructured information for cross enterprise applications, thus enabling an automated and compatible business-to-business, peer-to-peer infrastructure. The ACI allows enterprise applications to understand and process content that exists in unstructured formats, such as e-mail, Web pages, office documents, and Lotus Notes.
Agent index	An index in IDOL server that stores agents and profiles.
agent	A process that searches for information about a specific topic. An administrator can create agents for users or allow users to create their own agents.

C

Category index	An IDOL server index that stores categories.
cluster	A hierarchically agglomerated collection of data that has been extracted from snapshots. Each cluster represents a concept area that contains a set of items, which share common properties. Clustering data allows you to make trends and developments in data visible.
community	All the people in a user's network neighborhood. It allows a user to find other people in the community who have been looking at similar documents or have agents that are similar to the user's agents.

connector	An Autonomy fetching solution (for example HTTP Connector, Oracle Connector, File System Connector and so on) that allows you to retrieve information from any type of local or remote repository (for example, a database or a Web site). It imports the fetched documents into IDX or XML file format and indexes them into IDOL server from where you can retrieve them (for example by sending queries to IDOL server).

D

Data index	An IDOL server index that stores content data. You can customize how data is stored in the Data index by configuring appropriate settings in the IDOL server configuration file.
database	An IDOL server data pool that stores indexed information. The administrator can set up one or more databases, and specifies how data is fed to the databases. By default IDOL server contains the databases Profile, Agent, Activated, Deactivated, News and Archive.

I

IAS (Intellectual Asset Protection System)	An integrated security solution to protect your data. At the front end, authentication checks that users are allowed to access the system on which result data is displayed. At the back end, entitlement checking and authentication combine to ensure that query results include only documents that the user has permission to see, from repositories that the user has permission to access.
IDOL server	The Autonomy Intelligent Data Operating Layer (IDOL) server, which integrates unstructured, semi-structured and structured information from multiple repositories through an understanding of the content, delivering a real time environment in which operations across applications and content are automated, removing all the manual processes involved in getting the right information to the right people at the right time.
IDX	A structured file format that can be indexed into IDOL server. You can use a connector to import files into this format or you can manually create IDX files.

indexing	The process of storing data in IDOL server. Data can be stored in different field types (such as, index, numeric and ordinary fields). It is important to store data in appropriate field types to ensure optimized performance.

P

profile	Information about a user that is based on the concepts in documents that the user reads. Every time users open a document their profile updates. This feature allows you to alert users to new documents that they are interested in (according to their profiles).

Q

query	A string that you submit to IDOL server, which analyzes the concept of the query and returns documents that are conceptually similar to it. You can submit queries to IDOL server to perform several kinds of search, such as natural-language, Boolean, bracketed Boolean, and keyword.

S

stemming	The process of extracting the morphological root of a word. In languages some words have a common morphological root. Autonomy provides stemming algorithms that reduce words to this form. This is useful because it allows concepts to be matched regardless of the grammatical use of words. In English for example, the words "help", "helpful", "helping" and "helped" can all be stripped down to their stem "help" without significant loss of meaning. Autonomy provides as standard a set of stemming algorithms for the most commonly used languages. Stemming is applied after stopwords have been discarded both at index time (when content is stored in IDOL server) and at query time (query text is stopped and stemmed before it is matched).
stopword list	(Also called **stoplist**). A list (located in the IDOL server langfiles directory) that contains common words (stopwords) that are not stored in IDOL server. Words as, for example, "the" or "a" are used too frequently to carry any significance and IDOL server does not require them to understand the concept of text.

stopping	The process of removing the words listed in the stopword list from documents before storing them in IDOL server and from query text before matching against IDOL server content.
stopword	A word that appears in a **stopword list**.

T

taxonomy	An automatically created hierarchical structure of clusters or other information. A taxonomy provides you with an overview of the 'information' landscape and an insight into specific areas of the information.
term	The basic entity that IDOL server stores (for example, a word in a document after IDOL server applies stemming).

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X