Black Pearl Knowledge Broker

User Guide

Version 2.1

Contents

2.1 Introduction

Part I • Foundations

Chapter 1 **Transforming Data Into Knowledge • 3**

Chapter 2 **Developing a Knowledge Base • 9**

Part II • Getting Started

Chapter 3 **Installing and Configuring Knowledge Broker • 15**

Appendix D Writing XQueries • 499

Glossary • 515

Index • 521

2.1 Introduction

Welcome to the *Knowledge Broker 2 User Guide,* which provides information for understanding and using the Black Pearl™ Knowledge Broker™. Knowledge Broker is an all-Java, net market platform that applies business rules to data captured from multiple sources—existing enterprise databases and the Internet—to create real-time recommendations and alerts. These recommendations take into account buyer/seller goals, preferences, behaviors, and market context. The result is more accurate and timely recommendations that both increase the likelihood a transaction will occur, and enhance long-term customer satisfaction. With Knowledge Broker you can:

- Use point-to-point, asynchronous communications to access and process disparate data located in JDBC-compliant datasources and XML files, including Web, relational database, and flat files.
- Generate real-time recommendations that are easily accessible through any standard Web browser or Java application

What's New in Version 2.1

New features in version 2.1 include the following:

Improved Ontology and Editor. The Ontology can now embed a more diverse range of objects, and the editor has improved functionality over previous versions.

New Client-Server Architecture. Knowledge Broker uses an enhanced Java 2 Enterprise Edition (J2EE) architecture, with support for XML, XQuery, and other advanced data technologies..

Heterogeneous Joins. Knowledge Broker can federate across persistent and event-based datasources, and perform logical joins and mappings to provide a consistent, unified view of customer data.

Application/Agent Editor. Knowledge Broker now supports agents that enable users to collect and modify data using triggers or schedules, modify and transform the data, then publish the results using triggers or schedules to a wide variety of persistent and event-based datasources. The graphical user interface (GUI) approach enables business users to create and modify complex control flow logic without necessarily possessing programming and coding skills.

Interaction Editor. The new Interaction Editor enables advanced Knowledge Broker users with XQuery programming experience to directly create powerful data source manipulation commands.

New Rule Editor. The new Rule Editor is optimized to provide complete access to the powerful capabilities of Knowledge Broker's lightening-fast RETE inferencing engine.

New Knowledge Discovery Center. The new Knowledge Discovery Center features a modular interface that will allow rapid extension and deployment of industry-standard data mining and statistical inferencing tools. It also features a powerful user-defined sampling interface for dealing with large volumes of data. The generated models can now be easily integrated within the Ontology using the GUI.

Java Server Pages Support. Knowledge Broker's JSP Application Program Interface (API) support enables application developers to quickly and easily create and develop web-based interfaces to the run-time version of Knowledge Broker.

Java Exits. Knowledge Broker's Java Exits API enables developers and advanced users with coding experience to plug-in extra functionality within Knowledge Broker using programmatic Java code.

How This Guide Is Organized

Part I ***Foundations***

Chapter 1 *Transforming Data Into Knowledge* provides an introduction to the human process of transforming data into knowledge.

Chapter 2 *Developing a Knowledge Base* provides an introduction to using Knowledge Broker to transform data into knowledge.

Part II ***Getting Started***

Chapter 3 *Installing and Configuring Knowledge Broker* describes how to download, install, configure, and uninstall Knowledge Broker.

Chapter 4 *Using Knowledge Broker* describes basic procedures for using Knowledge Broker.

Part III ***Procedures***

Chapter 5 *Defining a Knowledge Base* provides step-by-step instructions for defining a knowledge base that contains all your ontologies, business rules, agents, and applications relevant to your business requirements.

Chapter 6 *Defining an Ontology* provides step-by-step instructions for defining an ontology containing business concepts that link to your data source and is relevant to your business requirements.

Chapter 7 *Making Connections and Mappings* provides step-by-step instructions for defining a connections to external data sources and mappings that link these connections to business concepts.

Chapter 8 *Creating Interactions* demonstrates how to embed user-defined XQueries into Knowledge Broker's ontology. These programmatic queries provide an enhanced degree of data flexibility for advanced users.

Chapter 9 *Creating Agents and Applications* describes how using Knowledge Broker's Agents and Applications enables Business Users to convert their static business concepts and models into active business processes.

Chapter 10 *Creating Rules* provides step-by-step instructions for defining Rules. Rules are logical statements that describe how an object (person, place, thing, or event) will behave in a particular situation. They provide discernment and selection for Knowledge Broker.

Chapter 11 *Discovering Knowledge* provides step-by-step instructions for using the Knowledge Discovery Center to create Data Sets and train Models that can be embedded within the Ontology to enable real-time analytics.

Part IV ***Tutorials***

Chapter 12 *Creating a Case Base* demonstrates how to create a case base from a data source.

Part V ***Appendices***

Appendix A *JSP Interface* describes how application programmers can create and develop web-based interfaces to the run-time version of Knowledge Broker.

Appendix B — *Java Exits* describes how advanced users can provide extra functionality within Knowledge Broker using programmatic Java code.

Appendix C — *XQuery Keywords* describes some of the reserved words in XQuery.

Appendix D — *Writing XQueries* details how to create and search XML Schema using XQuery.

Glossary — Provides a glossary of key concepts relating to Knowledge Broker.

Index — Provides an index of key concepts within this *User Guide*.

Audience

This guide is intended for people who will use Knowledge Broker to perform a variety of tasks:

User	Tasks	Important Chapters
System Administrators	Download, install, and monitor Knowledge Broker	• Chapter 3, 4, 7
Deployment Engineers	Build a knowledge management system	• Chapters 1, 2, 4, 5, and 8; • Appendices A, B, C
Business Users	Define ontologies and applications	• Chapters 1, 2, 5, 6, 9, 10; • Appendices A, B, C
End-users	View real-time recommendations	• Chapters 1 and 2

Document Conventions

This guide uses a variety of formats to identify different types of information.

Convention	Function
`courier`	Identifies syntax statements, on-screen computer text, and path, file, drive, directory, database, and table names.
`<courier>`	Identifies variable names.
`bold courier`	Identifies text you must type.
italics	Identifies document and chapter titles, special words or phrases used for the first time, and words of emphasis.
underline	Identifies URLs, domain names, and email addresses.
Initial Caps	Identifies Window, menu, command, button, option, tab, keyboard, and product-specific names.
ALL CAPS	Identifies acronyms and abbreviations.
[ ]	Identifies an optional item in syntax statements.
{ }	Identifies an optional item that can be repeated as necessary within a syntax statement.
>	Identifies a separation between a menu and an option.
|	Identifies a separation between items in a list of unique keywords when you may only specify one keyword.

Special Message Conventions

❗ Identifies information that will help prevent equipment failure or loss of data.

💡 Identifies information of importance or special interest, including Notes and Tips.

Menu Conventions

This guide uses the **Menu > Option** convention. For example, “**Click Format > Style**” is a shorthand instruction for “Click the Format menu, then select the Style option.”

Mouse Conventions

To select something, place the on-screen pointer or cursor on the item and click the left mouse button.

To view an **Options** menu, place the on-screen pointer or cursor on an item and click the right mouse button (or left mouse button if using a left-handed mouse). If a menu is available, it will open. (Clicking the right mouse or left mouse button is refered to in this guide as *option-click*.)

When the term *click the mouse on...*is used without qualification, it means to place the on-screen pointer or cursor on an item and click the left mouse button.

To drag something, click the mouse on it and drag the pointer to a different location before releasing the mouse button.

When selecting items from a list using the mouse, you can sometimes select more than one item by holding down the **Shift** or **Control** key while clicking the mouse.

To select a contiguous block of items, click on one item, hold the **Shift** key down, and click on a second item. All items between the two will be selected when multiple selection is enabled.

To select items from different locations when multiple selection is enabled, hold the **Control** key down. Each selected item will remain selected until you complete the action or click the mouse without holding the **Control** key down.

Additional Help

For additional information or advice, contact:

Contact Information

Application	Online Help and Readme text file
Phone	(415) 357-8300
Facsimile	(415) 357-8399
Internet	http://www.blackpearl.com
Email	sales@blackpearl.com techsupport@blackpearl.com techpubs@blackpearl.com
Postal	Black Pearl, Inc. 400 Second Street, Suite 450 San Francisco, CA 94107

Part

I

Foundations

Knowledge Broker is an all-Java, net market platform that transforms data and information into knowledge. It lets you intelligently apply business rules and predictive models to data captured from multiple sources—existing enterprise databases and the Internet—to generate real-time recommendations. These recommendations, which incorporate buyer/seller goals, preferences, behaviors, and market context, both increase the likelihood a transaction will occur, and enhance long-term customer satisfaction.

But what does this mean? What are data, information, and knowledge? What are business rules, predictive models, and real-time recommendations?

Part I: Foundations answers those questions and provides a foundation for understanding Knowledge Broker.

Chapter

1

Transforming Data Into Knowledge

For nearly 3,000 years, questions about human learning and knowledge have been a source of study, discourse, and debate for writers, philosophers, and theologians. Within the last century, social and cognitive scientists, neuro-physiologists, physicists, computer scientists, and engineers became interested in the subject.

It is not within the scope of this guide to present all the discourse on human learning and knowledge. Instead, this chapter will focus on how humans transform data into knowledge, viewed from a perspective relevant to understanding Knowledge Broker.

This chapter is deliberately constructed in a linear fashion. Each section builds upon the ideas presented in the previous sections. It is recommended that you read this chapter in sequential order, and before reading Chapter 2.

What is a Data Element?

A *data element* is a word, number, measurement, or fact that represents or refers to an object. That object can be any person, place, thing, thought, idea, or event that exists (or is believed to exist) in space and/or time. A data element is only a pointer to the object and not the object itself, and as such, is without any intrinsic or inherent meaning.

People, however, naturally apply meaning to data elements. For example, to most English-speaking people, *May* is the name for a month in a calendar year. It comes after the month of April, and before the month of June. It has 31 days. In the Northern Hemisphere, May is the last full month in the spring season. May is also the name for a woman. And, if the word is written as *may,* then it is a verb denoting the right to do something. In other words, the data element *May* has meaning because we understand it in relationship (*context)* to the data elements *name*, *month*, *calendar*, *year*, and *action*.

To keep track of and organize these relationships, we develop an abstract image (*concept)* about the who, what, when, and where of those relationships. Thus, *May* is a data element when viewed as a discrete, stand-alone object. It is a concept when viewed as being in relationship with other data elements or concepts.

Concepts, however, develop over time and as a result of our dynamic interactions with the world and other people. Our experiences—and the learning that is a consequence of those experiences—help us to contextualize and conceptualize the world. For example, a child encounters a bright, flickering object. Someone says the word “candle.” This scene is repeated several times until the day arrives when the child sees the object and says “candle.” The child experiences something that exists in space and time (the bright, flickering object), relates the object to the sound “candle” (another object that exists in space and time), creates a mental image of the connection between the objects, and learns the concept “candle.” This process is repeated hundreds of times each day, with many different objects, until the child develops an extensive vocabulary of concepts and contexts that describe the world.

It is these concepts and contexts that enable the child to function within the world, because that world now has meaning—as defined by those concepts and contexts. These concepts and contexts, which are built upon data elements, are the foundation for information.

What is Information?

Information is the association of concepts and contexts whose meanings are understood. For example, “May has 31 days” is information if you or another person understands that:

- The data element “May” is transformed into the concept “May” when understood in relationship with name, months, calendar, and year.
- The data element “31” is transformed into the concept “31” when understood in relationship with numbers.
- The data element “days” is transformed into the concept “days” when understood in relationship with measurements of time.

Information, which can be communicated through spoken and written words, pictures, symbols, movements, touch, scents, and other methods, enables you to learn, create more and more complex concepts, contexts, and information, and develop critical thinking. For example, you learn that the bright object is a candle when you decide one day to touch the brightness. Instead of obtaining the object of desire, you scream out in pain. Your mother or father says “no, hot.” You have no idea what “hot” means, but do know you don’t like the feeling in your fingers! You associate the sound “hot” with the unpleasant sensation in your fingers and conceptualize that “Candles are hot. No!” What you’ve accomplished is to relate two discrete concepts (candle, hot) with an experience (pain in your fingers) to create a new set of information (don’t touch candles because they are hot).

This process of using information to learn, create ever more complex concepts, contexts, and information, and develop critical thinking, is the basis for knowledge.

What is Knowledge?

Knowledge is four interrelated things:

- The understanding of patterns of information that consistently and completely repeat themselves over time (*pattern knowledge*). For example, you are *pattern knowledgeable* if you understand that May is the fifth month of the calendar year, always has 31 days, your mother's birthday is on May 23, she lives in Florida, and she always expects you to join her for a birthday dinner.
- The recognition of the implications of those patterns on future behavior (*implication knowledge*). For example, you are *implication knowledgeable* if, on April 15, you understand that there are exactly thirty-eight days before your mother's birthday, you live in California, and you must arrange to fly to Florida for her birthday dinner.
- The capacity to make plans and recommendations based on those implications (*strategy knowledge*). For example, you are *strategy knowledgeable* if you understand that you must purchase an airline ticket by April 23 if you want to take advantage of the 30-day advance fare rates.
- The ability to act on those plans (*action knowledge*). For example, you are *action knowledgeable* if you actually purchase the ticket by April 23.

Each level of knowledge is gained through learning, and learning can only take place through interaction, whether that interaction is:

- Sensorial (learned through seeing, touching, tasting, smelling, or hearing something)
- Experimental (learned by testing a hypothesis to determine whether or not something is true)
- Informational (learned through conversation, reading, or social/educational norms about something)
- Mental (learned through internal, abstract conceptualizations about something)

Daily interactions afford you with opportunities to learn about interconnected webs of patterns, implications, strategies, and actions. If you learn to recognize and understand the patterns, you are pattern knowledgeable. If you learn to recognize and understand the implications, you are both pattern and implication knowledgeable. And so on. The more levels of knowledge you gain, the deeper your understanding. You gain more levels of knowledge, and the you are better able to orient yourself in a particular environment, overcome difficulties, make decisions, and solve problems.

But knowledge is not a static, one-time event. It is both dynamic and iterative. It is *dynamic* if it makes use of existing data elements, concepts, contexts, information, and knowledge to create additional knowledge. For example, if you know that all mammals breathe, that humans are mammals, and that John is a human, then you can say, quite knowledgeably, that John breathes, even though no one explicitly told you that John breathes. It is *iterative* if it repeats the learning process for a particular concept, context, or information in order to increase your understanding. For example, you may first learn the concept "red" by seeing something red and being told "red." Next, you may learn that red is a "color." Later, you may learn that the color red is a "primary color." Still later, you may learn that the primary color red is a part of the "visible spectrum of light." In each case, you are deepening your understanding of the concept "red" through an iterative process.

The dynamic and iterative nature of knowledge is based on two doctrines:

- Association of ideas
- Compositionality

The *association of ideas* is the ability to unite in some manner two or more concepts, based on resemblance and/or contiguity. *Resemblance* states that when two concepts are considered similar, whatever properties associated with one concept are automatically associated with the other concept. Thus, the statement "John looks like a human, therefore he must breathe since all humans are mammals and all mammals breathe" is an example of the association of ideas through resemblance. *Contiguity* states that when concepts are frequently experienced together, they are always mentally associated together. Thus, the statement "Red is a primary color that is a part of the visible spectrum of light" is an example of the association of ideas through contiguity.

Compositionality is the ability to infer knowledge from the meanings, patterns, and relationships of individual concepts or information, based on a system of rules. *Rules* are logical statements that describe how an object (person, place, thing, or event) will behave in a particular situation. There are three basic types of rules:

- *Legal rules,* which are mandated by a federal, state, city, or other warrantable agency (including SEC, OSHA, FCC, EPA, and unions), cannot be broken unless the issuing agency amends the rule. For example, an SEC rule mandates a "quiet period" before a company's initial public offering.
- *Domain-specific rules,* which are defined by a specific business or field of knowledge (including physics, chemistry, and law), can also not be broken unless amended by the issuing domain. For example, a brokerage house may stipulate that stockbrokers employed with the firm for less than six months may not offer speculative stock advice.
- *Common sense rules,* which include opinions, hunches, and best practices, can be broken if circumstances warrant. For example, a brokerage house may institute a "best practices" policy whereby speculative stocks will not be recommended to a person who is 70 years old and living on a fixed income.

Thus, "You touch the candle flame with your fingers. Your fingers are burned. Do not touch a burning candle," is an example of knowledge inferred through compositionality. It is based on two types of rules: domain-specific and common-sense. The domain-specific rules states that fire is hot and will burn other objects. The common-sense rule states that if you touch a candle flame, you will burn your fingers.

Knowledge gained, whether through association of ideas or compositionality, is implicit or explicit. *Implicit knowledge* is a set of personal experiences, know-how, and mental images that is rarely (if ever) communicated with others. *Explicit knowledge* is a set of codified experience and know-how that is communicated through formal language, including data, concepts, contexts, information, and rules.

The collection of implicit and/or explicit knowledge is a *knowledge base,* which is the foundation for logical predictions and actions.

Chapter

2

Developing a Knowledge Base

Whereas a human knowledge base can include both implicit and explicit knowledge, a computerized knowledge base must be explicit. Knowledge Broker enables you to create an explicit knowledge base that includes information about datasources, concepts, contexts, and rules. It then uses that knowledge base to discern patterns in your raw data, understand the implications of those patterns, make real-time recommendations, and dynamically and iteratively refine and deepen the knowledge base.

This chapter provides a conceptual overview of Knowledge Broker objects and processes that enable you to define and refine a knowledge base, including:

- Ontologies
- Actions

What is an Ontology?

An *ontology* is a collection of concepts and contexts that provides a common vocabulary used for defining rules, analyzing information patterns, recognizing implications, and making real-time recommendations. Knowledge Broker's ontology consists of five main components:

- Root Business Concepts
- Derived Root Business Concepts
- Derived Business Concepts
- Connections
- Mappings

The concepts and concept relations defined within the ontology are the means for "teaching" Knowledge Broker about your business—as it is contained within your Web, relational database, and flat file datasources—and communicating with your datasources to discover knowledge and generate real-time recommendations.

You can use Knowledge Broker's Ontology Editor to easily define an ontology linked to your datasources and relevant to your business requirements.

Active Knowledge

To act on knowledge to create effects and new knowledge is a key characteristic of an intelligent system such as Knowledge Broker. The Agent/Application Editor enables you to process ontology components and produce output based on specific conditions.

Agents

Agents are self-contained blocks of Knowledge Broker agent language commands. Agents can be set listen for a keyword trigger, to activate during specific time periods, or to wait for an Application trigger.

Applications

Applications tie together agents and can output Knowledge Broker recommendations to a variety of external systems and feeds. Applications use a subset of the agent language commands.

What is a Knowledge Base?

A *knowledge base* is the collection of ontology concepts, agents, and applications related to a particular industry or business domain. A knowledge base contains all the information concerning a specific business model and provides a handy repository for your business knowledge.

Part II

Getting Started

Knowledge Broker is an easy-to-use application that includes self-explanatory installation and use procedures. However, a quick introduction to the basic procedures required to install and use Knowledge Broker will increase your familiarity with the application.

Part II: Getting Started provides you with information for downloading, installing, configuring, and using Knowledge Broker.

Chapter

3

Installing and Configuring Knowledge Broker

Knowledge Broker can be either downloaded from the Black Pearl website or obtained on a CD-ROM. It is bundled in a Java Edition InstallAnywhere application, and is designed for quick and easy installation.

This chapter provides step-by-step instructions for installing and uninstalling Knowledge Broker on a standalone, client, or server machine using either the Windows NT 4.0 or Solaris 8 operating systems. Knowledge Broker's standalone installation is a self-contained single-machine client-server option. Distributed installations of Knowledge Broker must be deployed as an Enterprise JavaBean (EJB) within the WebLogic 6.1 application server. This chapter also includes information about system and configuration requirements for Windows NT, Solaris, and EJB application server environments.

System Requirements

Microsoft Windows

To use Knowledge Broker successfully, your Windows NT machine requires the following:

Table 3-1. Windows NT Requirements

Function	Description
Operating System	• **Server:** • Windows NT Server 4.0, Service Pack 4 or higher; • Windows 2000 Server, Advanced Server, or Datacenter Server with Service Pack 2 • **Client:** • Windows NT Workstation 4.0, Service Pack 4 or higher. • Windows 2000 Professional, Server, Advanced Server, or Datacenter Server with Service Pack 2
CPU	266 MHz Pentium II-class CPU (or greater)
Disk Space	Minimum 500 MB
Memory	Minimum 64 MB RAM
Database Engine	JDBC 2.0-compliant
Web Browser	Internet Explorer 4.0 (or higher), or Netscape 4.7 (or higher)
Network	TCP/IP
Java Environment	Java Runtime Environment (JRE) 1.3

Sun Solaris

Your Solaris system requirements are these:

Table 3-2. Sun Solaris Requirements

Function	Description
Operating System	Solaris 8 Version 5.8 locale: EN_US ISO 8859-1, 64-bit enabled .
CPU	230 MHz single-processor SPARC (minimum)
Disk Space	Minimum 200 MB
Memory	Minimum 128 MB RAM
Database Engine	Oracle 8i
Web Browser	Internet Explorer for Unix 5 SP1 or Netscape 4.7 (or higher)
Network	TCP/IP
Java Environment	• Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1-b24) • Java HotSpot(TM) Client VM (build 1.3.1-b24, mixed mode)

If individual Knowledge Broker component or third-party application requirements differ or exceed these baseline requirements, this is noted in the text.

Downloading Knowledge Broker

Knowledge Broker resides on a secure directory on the Black Pearl website. Contact your Black Pearl sales representative to obtain a User Name and Password, both of which are required to access the directory.

1. Establish an Internet connection.
2. Type **`http://www.blackpearl.com/download`** into your web browser.
3. Type the User Name and Password assigned by your Black Pearl sales representative.
4. For Windows, select the `broker.exe`. For Solaris, select the `broker.bin` file.
5. Save the appropriate installer file to your local machine.
6. Exit the Black Pearl website after downloading the appropriate files.

Types of Installations

There are three types of installations:

- Server
- Client
- Standalone

The Server installation consists of server and client components that are installed on the same machine. The Client installation consists of client components that must be connected to a remote server. The Standalone installation is a minimal installation with limited functionality, intended for quick deployment and simple testing.

Server Installation

The *Server* installation comprises Server and Client components that are installed on the same machine. The graphical user interface-based (GUI) Client installs, but communication between Server and Client uses HTTP on the local host provided by your WebLogic deployment.

You must first install Knowledge Broker's Server and Client components using Knowledge Broker's InstallAnywhere package. Then you must deploy Knowledge Broker within an Enterprise Java Bean (EJB) framework.

You must select an Application for the Knowledge Broker installation to be configured. The three available applications are: TIB/RV, WebLogic, and WebSphere. If you have an application installed and are using Windows, the installation locates the correct directory in the Windows Registry and connects to Knowledge Broker. If you have no application server you must either install standalone (which comes with its own mini application server), or just the client (because you intend to attach to a server located on another machine).

Integrating Knowledge Broker within a J2EE Framework

Knowledge Broker supports installation within J2EE application servers that support Enterprise Java Beans (EJB). Application servers encapsulate much of the business logic within enterprise systems by providing a component abstraction layer. They handle database access, transaction coordination, and provide an integrated component framework for distributed applications along with administrative and security functions.

Enterprise JavaBeans (EJB) defines a server-side technology model for developing, integrating, and deploying Java components and applications in a scalable, secure, and transactional fashion. For further information, refer to: http://www.javasoft.com/products/ejb/.

By deploying Knowledge Broker within an EJB application server, you gain access to many enterprise, messaging, and database access functions while improving Knowledge Broker's interoperability across heterogeneous networks. `KBRuntime.war` is the application server component of Knowledge Broker.

Knowledge Broker currently supports deployment within these J2EE application servers:

- BEA WebLogic Server 6.1
- IBM WebSphere Application Server Advanced Edition V4.0
- BEA WebLogic Server 5.1 (deprecated, see the `ReadMe` file)

Installing Knowledge Broker Server

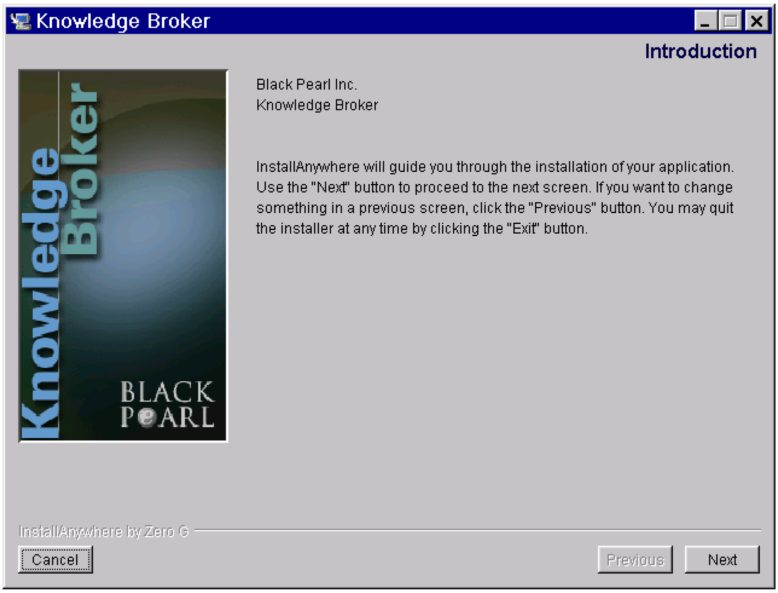
Knowledge Broker's installation process enables you to select from a set of options. After you select the appropriate options, installation proceeds automatically.

1 Complete one of the following options:

   a If you downloaded Knowledge Broker, open the directory containing the application file and double-click `broker.exe`. The InstallAnywhere application automatically runs and Knowledge Broker's Introduction dialog displays.

b If you received Knowledge Broker on a CD-ROM, insert the CD into the appropriate CD-ROM drive. The InstallAnywhere application runs and Knowledge Broker's Introduction dialog displays.

Figure 3-1. Knowledge Broker - Server Installation

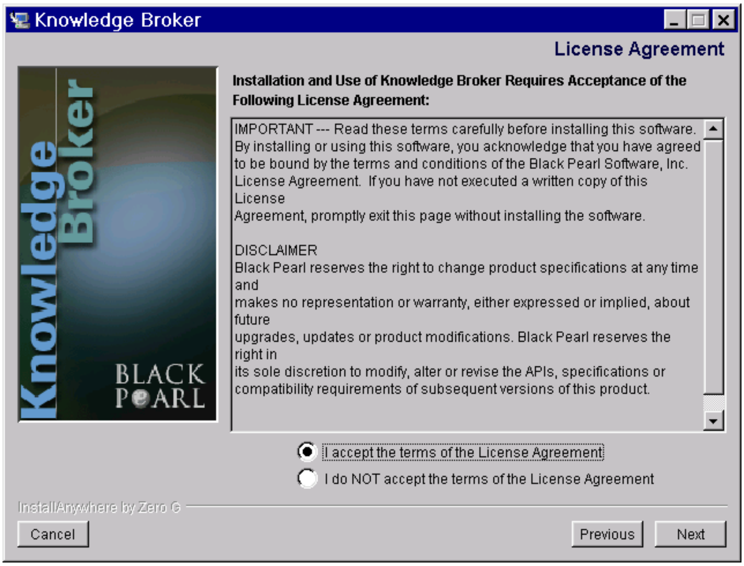


Black Pearl Inc.
Knowledge Broker

InstallAnywhere will guide you through the installation of your application. Use the "Next" button to proceed to the next screen. If you want to change something in a previous screen, click the "Previous" button. You may quit the installer at any time by clicking the "Exit" button.

2 Click Next. The License Agreement dialog displays.

Figure 3-2. License Agreement - Server Installation



Installation and Use of Knowledge Broker Requires Acceptance of the Following License Agreement:

IMPORTANT --- Read these terms carefully before installing this software. By installing or using this software, you acknowledge that you have agreed to be bound by the terms and conditions of the Black Pearl Software, Inc. License Agreement. If you have not executed a written copy of this License
Agreement, promptly exit this page without installing the software.

DISCLAIMER
Black Pearl reserves the right to change product specifications at any time and
makes no representation or warranty, either expressed or implied, about future
upgrades, updates or product modifications. Black Pearl reserves the right in
its sole discretion to modify, alter or revise the APIs, specifications or compatibility requirements of subsequent versions of this product.

● I accept the terms of the License Agreement
○ I do NOT accept the terms of the License Agreement

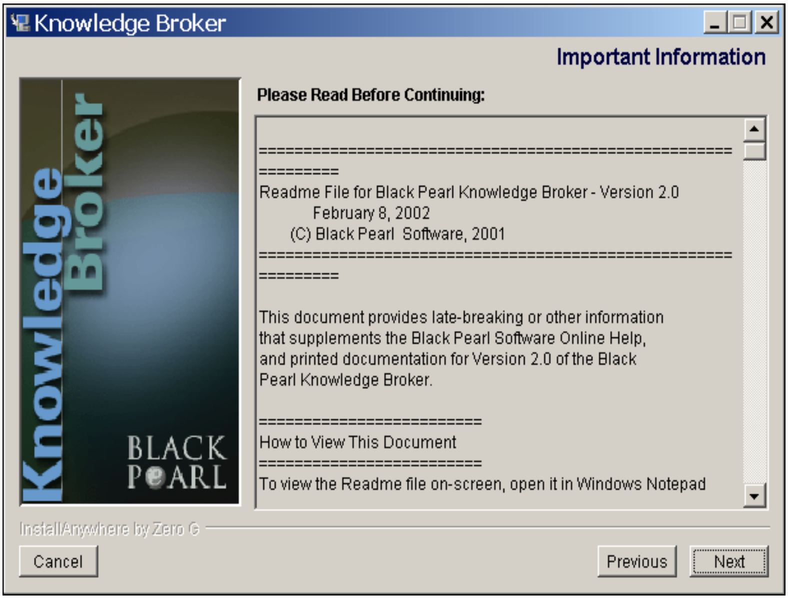
3 Select one of the following options:

a If you do not accept Knowledge Broker's license agreement, click Exit. Click Yes when the Confirm Exit dialog displays.

b If you accept Knowledge Broker's license agreement, select Yes.

4 Click Next. The Important Information dialog displays.

Figure 3-3. Important Information - Server Installation



Please Read Before Continuing:

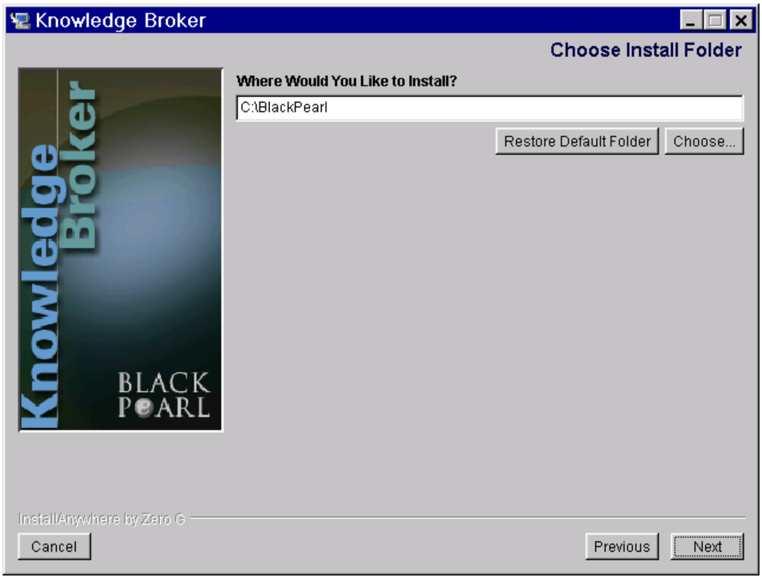
```
======================================================
=========
Readme File for Black Pearl Knowledge Broker - Version 2.0
         February 8, 2002
      (C) Black Pearl  Software, 2001
======================================================
=========

This document provides late-breaking or other information
that supplements the Black Pearl Software Online Help,
and printed documentation for Version 2.0 of the Black
Pearl Knowledge Broker.

=========================
How to View This Document
=========================
To view the Readme file on-screen, open it in Windows Notepad
```

5 (Optional) Scroll down the Important Information dialog to review the Readme file prior to installation of Knowledge Broker.

After installing Knowledge Broker, you may access the Readme file by opening file in a standard text editor such as Notepad.

6 Click Next. The Choose Install Folder dialog displays.

Figure 3-4. Choose Install Folder - Server Installation



Choose Install Folder

Where Would You Like to Install?

C:\BlackPearl

7 Complete one of the following options:

a Accept the default installation settings. By default, all Knowledge Broker files install to the `C:\BlackPearl\` folder (Windows) or the `/usr/blackpearl` directory (Solaris).

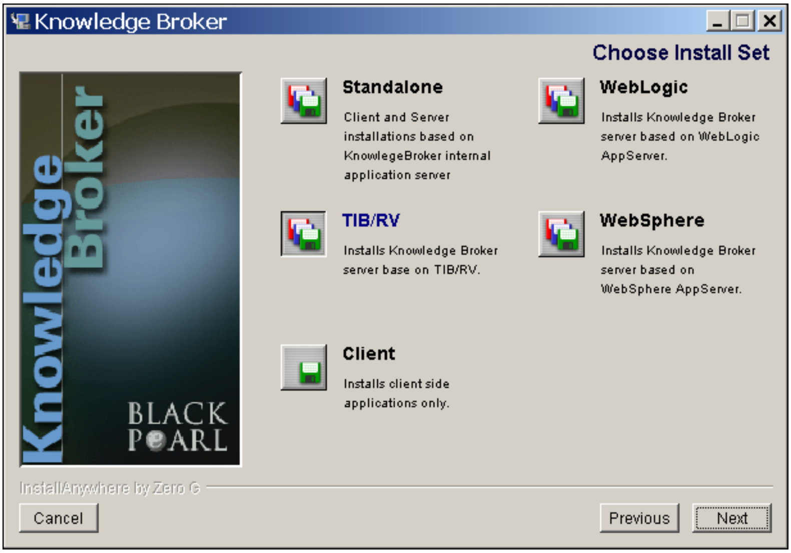
b Type the exact path to designate a destination folder. By default, all Knowledge Broker's files install to this folder.

8 Click Choose and use the file selection dialog to select a drive and folder. By default, all Knowledge Broker's files install to this folder.

The installation file name and parent directories must be alphanumeric and must not contain any spaces.

c Click Next. The Choose Install Set dialog displays.

Figure 3-5. Choose Install Set - Server Installation



Knowledge Broker

Choose Install Set

Standalone
Client and Server installations based on KnowlegeBroker internal application server

WebLogic
Installs Knowledge Broker server based on WebLogic AppServer.

TIB/RV
Installs Knowledge Broker server base on TIB/RV.

WebSphere
Installs Knowledge Broker server based on WebSphere AppServer.

Client
Installs client side applications only.

You must now select an Application for the Knowledge Broker installation to be configured. Select one of the following application server options:

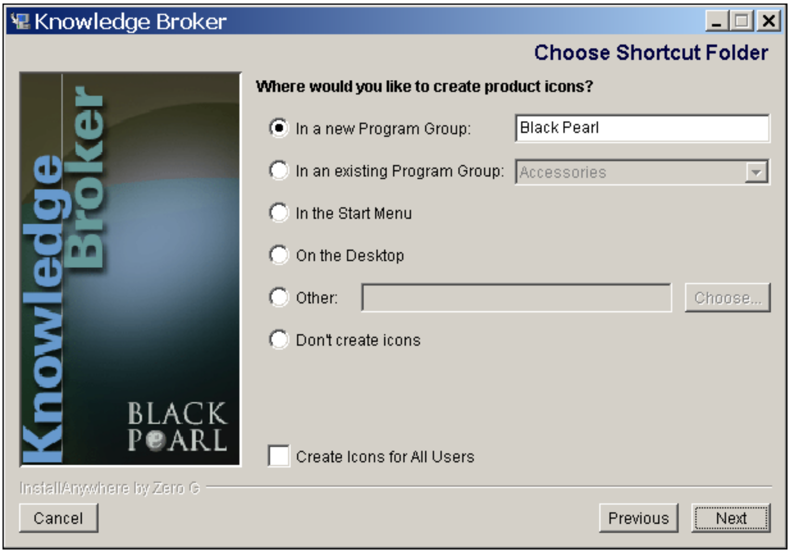
- TIB/RV
- WebLogic
- WebSphere

TIB/RV Application Server

1 Select TIB/RV from the Choose Install Set dialog.

2 Click Next. Knowledge Broker installs to your server machine. The Choose Shortcut Folder Window displays.

Figure 3-6. Choose Shortcut - TIB/RV Server Installation



Knowledge Broker

Choose Shortcut Folder

Where would you like to create product icons?

● In a new Program Group: Black Pearl

○ In an existing Program Group: Accessories

○ In the Start Menu

○ On the Desktop

○ Other:

○ Don't create icons

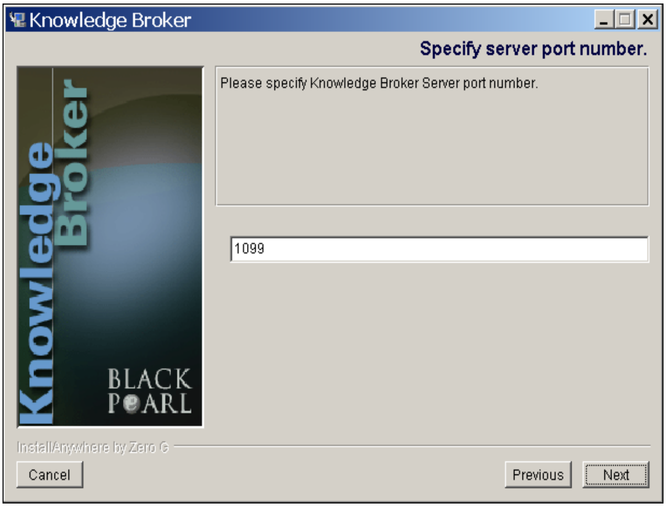
□ Create Icons for All Users

3 Select the appropriate option and click Next. You must now tell the Black Pearl Client installation process where to find the Server installation components.

See the Readme for up-to-date information about your installation choices.

4 The Specify Server Port Number dialog displays. Specify Server Port Number.

Figure 3-7. Specify Server Port Number - TIB/RV Server Installation



Knowledge Broker

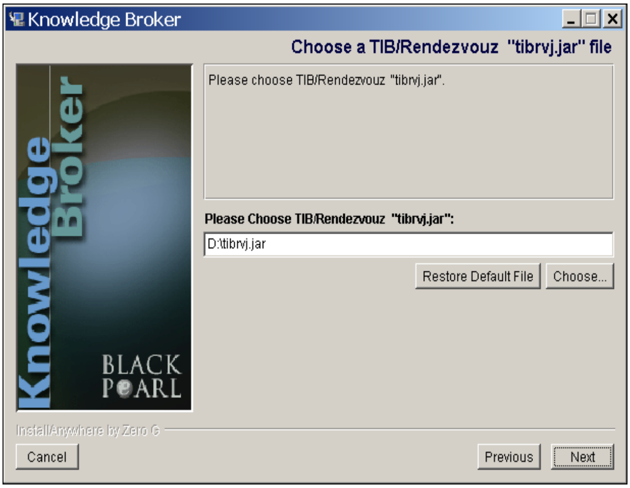
Specify server port number.

Please specify Knowledge Broker Server port number.

1099

5 Click Next. The Choose a TIB/Rendezvous "tibrvj.jar" file dialog displays. .

Figure 3-8. Choose a TIB/Rendezvous tibrvj.jar file - TIB/RV Server Installation



Knowledge Broker

Choose a TIB/Rendezvouz "tibrvj.jar" file

Please choose TIB/Rendezvouz "tibrvj.jar".

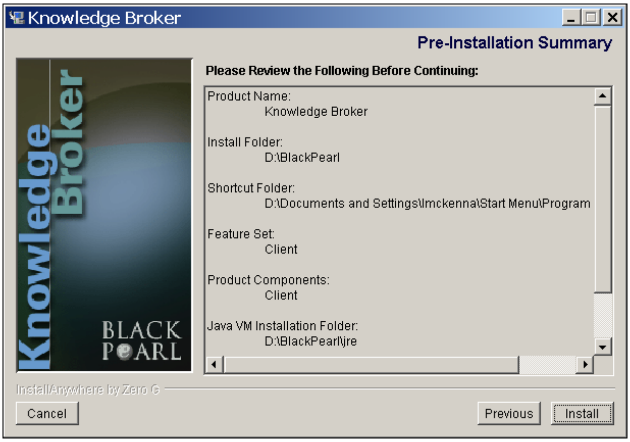
Please Choose TIB/Rendezvouz "tibrvj.jar": D:\tibrvj.jar

Complete one of the following options:

- **a** Accept the default installation settings. By default, all Knowledge Broker files install to the `C:\BlackPearl\` folder (Windows) or the `/usr/blackpearl` directory (Solaris).
- **b** Type the exact path to designate a destination folder. By default, all Knowledge Broker's files install to this folder.

6 Click Next. Knowledge Broker begins the installation of the TIB/RV application server. The Pre-Installation Summary displays.

Figure 3-9. Pre-Installation Summary - TIB/RV Server Installation



Knowledge Broker

Pre-Installation Summary

Please Review the Following Before Continuing:

Product Name:
Knowledge Broker

Install Folder:
D:\BlackPearl

Shortcut Folder:
D:\Documents and Settings\lmckenna\Start Menu\Program

Feature Set:
Client

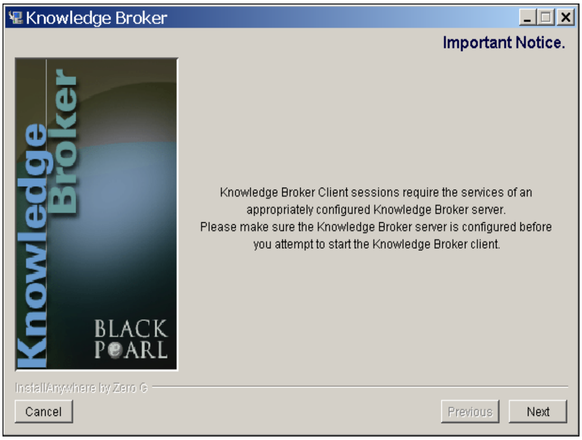
Product Components:
Client

Java VM Installation Folder:
D:\BlackPearl\jre

7 Click Install. The Knowledge Broker is installed.

8 Click Next. An Important notice displays.

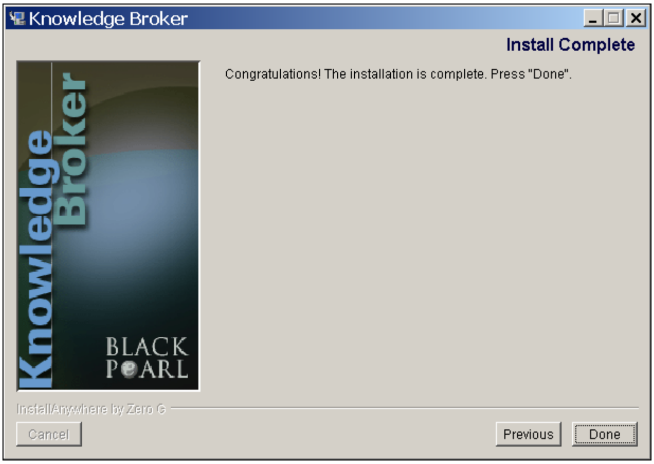
Figure 3-10. Important Notice - TIB/RV Server Installation



Knowledge Broker Client sessions require the services of an appropriately configured Knowledge Broker server.
Please make sure the Knowledge Broker server is configured before you attempt to start the Knowledge Broker client.

9 Click Next. The Install Complete dialog displays.

Figure 3-11. Install Complete - TIB/RV Server Installation



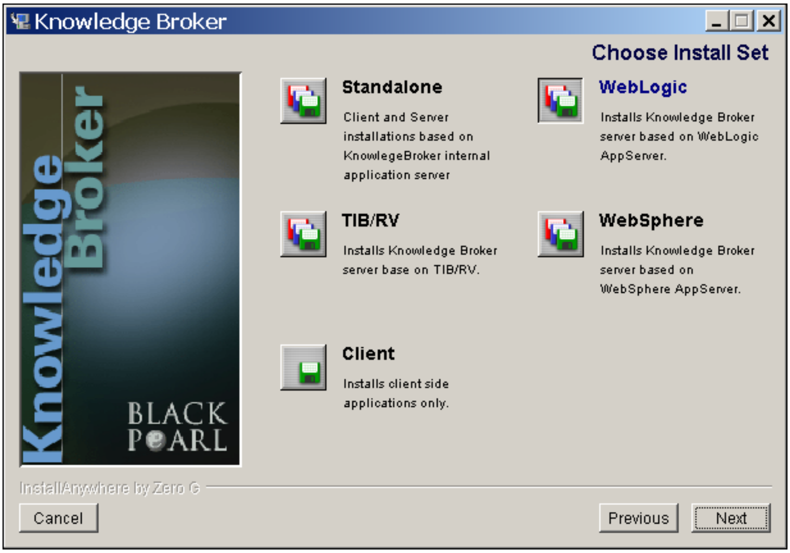
Congratulations! The installation is complete. Press "Done".

10 Click Done. You have now installed Knowledge Broker.

WebLogic Application Server

1 Select WebLogic from the Choose Install Set dialog.

Figure 3-12. Choose Application Server - WebLogic Server Installation



Knowledge Broker

Choose Install Set

Standalone
Client and Server installations based on KnowlegeBroker internal application server

WebLogic
Installs Knowledge Broker server based on WebLogic AppServer.

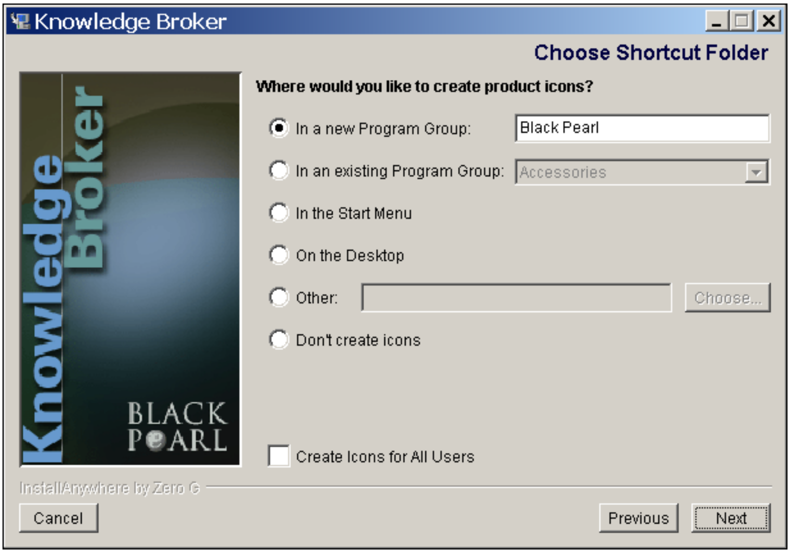
TIB/RV
Installs Knowledge Broker server base on TIB/RV.

WebSphere
Installs Knowledge Broker server based on WebSphere AppServer.

Client
Installs client side applications only.

2 Click Next. Knowledge Broker installs to your server machine. The Choose Shortcut Folder Window displays.

Figure 3-13. Choose Shortcut - WebLogic Server Installation



Knowledge Broker

Choose Shortcut Folder

Where would you like to create product icons?

● In a new Program Group: Black Pearl

○ In an existing Program Group: Accessories

○ In the Start Menu

○ On the Desktop

○ Other:

○ Don't create icons

□ Create Icons for All Users

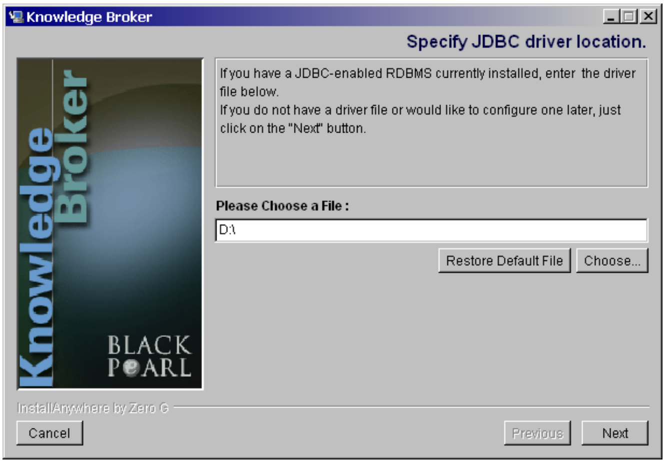
3 Select the appropriate option and click Next. You must now tell the Black Pearl Client installation process where to find the Server installation components.

See the Readme for up-to-date information about your installation choices.

You can connect to one or more JDBC-compliant datasources (for more information, see *Configuring JDBC datasource Driver Locations on page 71*). Note that within Microsoft Windows, you may instead use the Microsoft ODBC bridge mechanism (see *Configuring ODBC datasource Drivers on page 69*) and click Next to skip this section.

The Specify JDBC Driver Location dialog displays, allowing you select a suitable JDBC driver.

Figure 3-14. Specify JDBC Driver Location - WebLogic Server Installation



Knowledge Broker

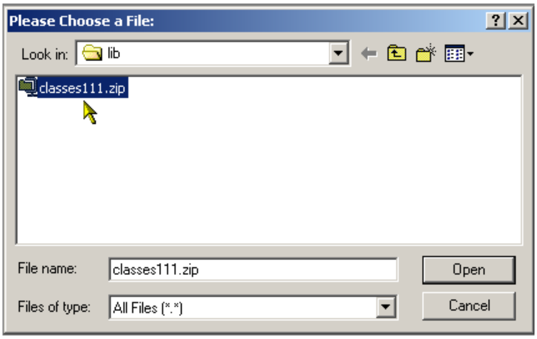
Specify JDBC driver location.

If you have a JDBC-enabled RDBMS currently installed, enter the driver file below.
If you do not have a driver file or would like to configure one later, just click on the "Next" button.

Please Choose a File : D:\

4 You can enter the file location, or click Choose to use the standard Windows file browser dialog to identify the file that contains a suitable JDBC driver:

Figure 3-15. File Locator - WebLogic Server Installation



Please Choose a File:

Look in: lib

classes111.zip

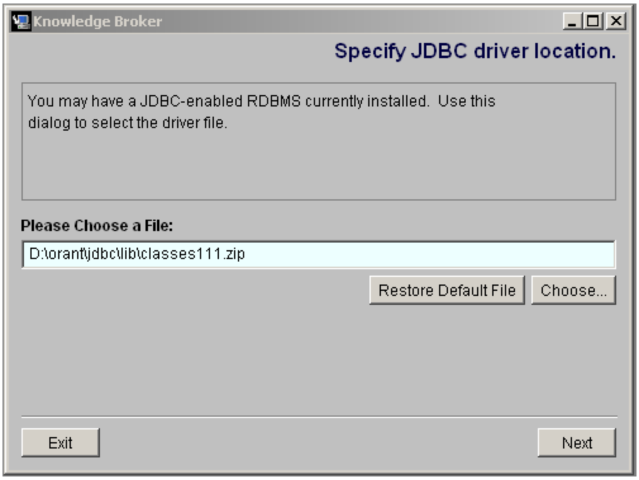
File name: classes111.zip

Files of type: All Files (*.*)

The name of the `.jar` or `.zip` file required depends on which database you wish to access. Your network administrator or DBA (Database Administrator) can help you locate suitable files that contain JDBC drivers.

5 Click Open to select the JDBC driver file. The Specify JDBC Driver Location dialog displays again, this time with the file location entered. Click Next to continue.

Figure 3-16. Specify JDBC Driver Location 2 - WebLogic Server Installation



Specify JDBC driver location.

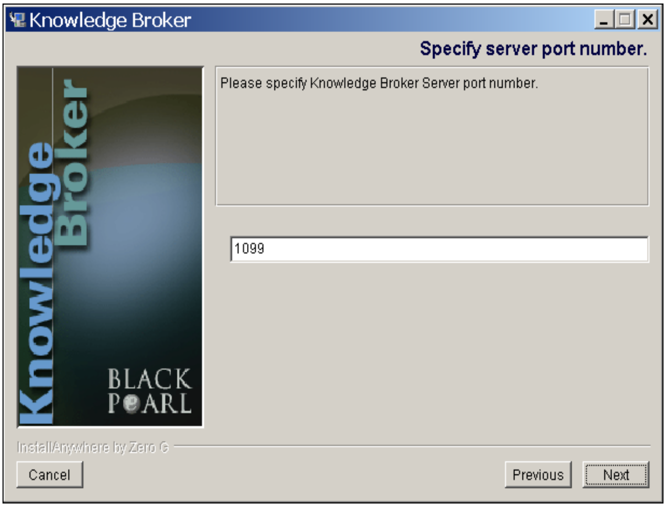
You may have a JDBC-enabled RDBMS currently installed. Use this dialog to select the driver file.

Please Choose a File:

D:\orant\jdbc\lib\classes111.zip

6 After all files install, the Installation Complete dialog displays. Click Next. The Specify Server Port Number dialog displays. Specify Server Port Number.

Figure 3-17. Specify Server Port Number - WebLogic Server Installation



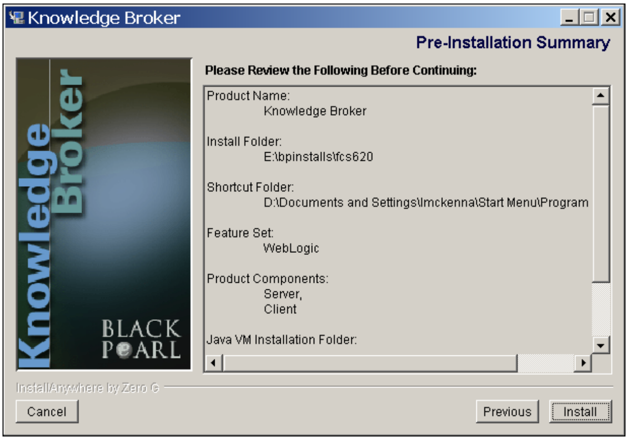
Specify server port number.

Please specify Knowledge Broker Server port number.

1099

7 Click Next. The Pre-Installation Summary displays.

Figure 3-18. Pre-Installation Summary - WebLogic Server Installation



Please Review the Following Before Continuing:

Product Name:
Knowledge Broker

Install Folder:
E:\bpinstalls\fcs620

Shortcut Folder:
D:\Documents and Settings\lmckenna\Start Menu\Program

Feature Set:
WebLogic

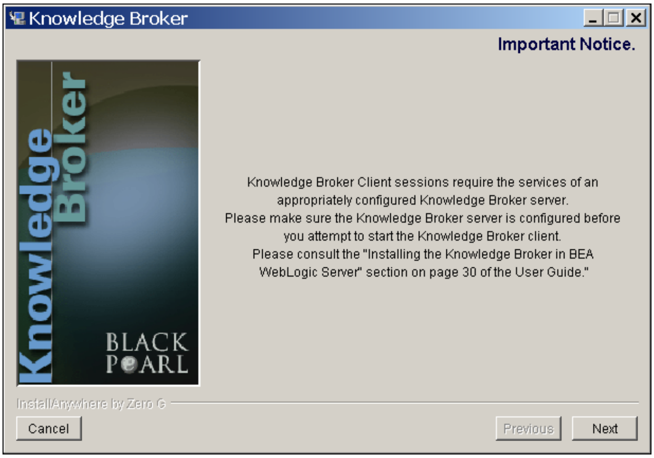
Product Components:
Server,
Client

Java VM Installation Folder:

8 Click Install. The Knowledge Broker is installed.

9 Click Next. An Important notice displays.

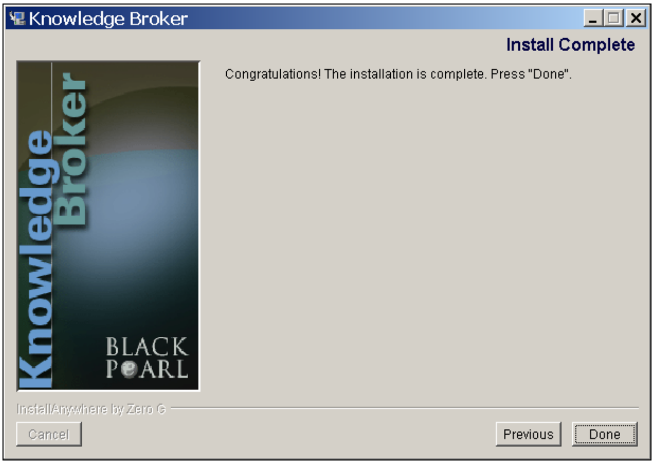
Figure 3-19. Important Notice - WebLogic Server Installation



Knowledge Broker Client sessions require the services of an appropriately configured Knowledge Broker server.
Please make sure the Knowledge Broker server is configured before you attempt to start the Knowledge Broker client.
Please consult the "Installing the Knowledge Broker in BEA WebLogic Server" section on page 30 of the User Guide."

10 Click Next. The Install Complete dialog displays.

Figure 3-20. Install Complete - WebLogic Server Installation



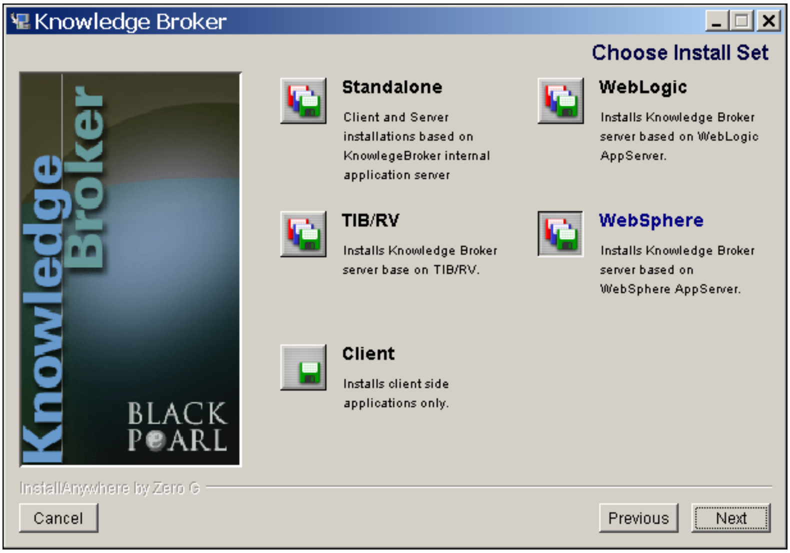
Congratulations! The installation is complete. Press "Done".

11 Click Done. You have now installed Knowledge Broker. Click Done.

WebSphere Application Server

1 Select WebLogic from the Choose Install Set dialog.

Figure 3-21. Choose Application Server - WebSphere Server Installation



Knowledge Broker

Choose Install Set

Standalone
Client and Server installations based on KnowlegeBroker internal application server

WebLogic
Installs Knowledge Broker server based on WebLogic AppServer.

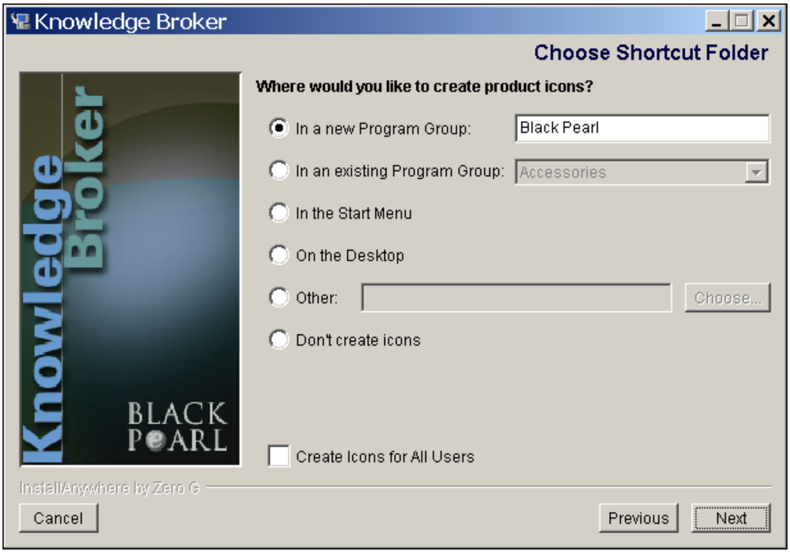
TIB/RV
Installs Knowledge Broker server base on TIB/RV.

WebSphere
Installs Knowledge Broker server based on WebSphere AppServer.

Client
Installs client side applications only.

2 Click Next. Knowledge Broker installs to your server machine. The Choose Shortcut Folder Window displays.

Figure 3-22. Choose Shortcut - WebSphere Server Installation



Choose Shortcut Folder

Where would you like to create product icons?

● In a new Program Group: Black Pearl

○ In an existing Program Group: Accessories

○ In the Start Menu

○ On the Desktop

○ Other:

○ Don't create icons

□ Create Icons for All Users

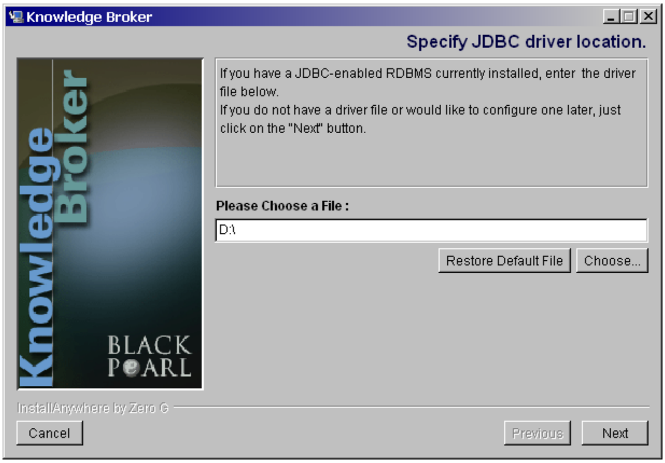
3 Select the appropriate option and click Next. You must now tell the Black Pearl Client installation process where to find the Server installation components.

See the Readme for up-to-date information about your installation choices.

You can connect to one or more JDBC-compliant datasources (for more information, see *Configuring JDBC datasource Driver Locations on page 71*). Note that within Microsoft Windows, you may instead use the Microsoft ODBC bridge mechanism (see *Configuring ODBC datasource Drivers on page 69*) and click Next to skip this section.

The Specify JDBC Driver Location dialog displays, allowing you select a suitable JDBC driver.

Figure 3-23. Specify JDBC Driver Location - WebSphere Server Installation



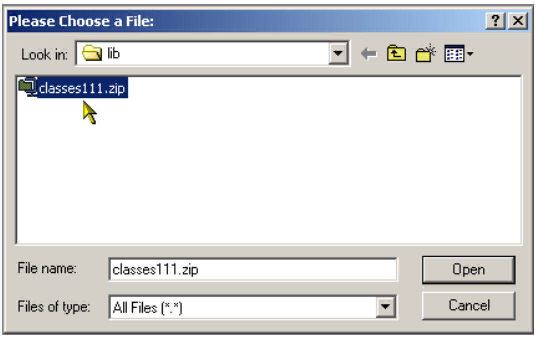
Specify JDBC driver location.

If you have a JDBC-enabled RDBMS currently installed, enter the driver file below.
If you do not have a driver file or would like to configure one later, just click on the "Next" button.

Please Choose a File : D:\

4 You can enter the file location, or click Choose to use the standard Windows file browser dialog to identify the file that contains a suitable JDBC driver:

Figure 3-24. File Locator - WebSphere Server Installation



Look in: lib

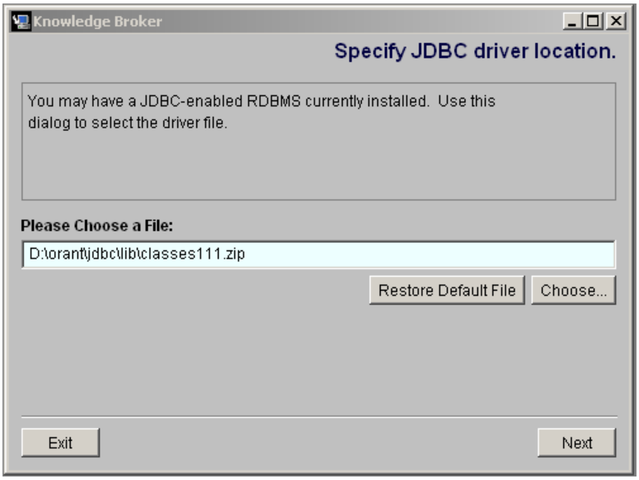
File name: classes111.zip

Files of type: All Files (*.*)

The name of the `.jar` or `.zip` file required depends on which database you wish to access. Your network administrator or DBA (Database Administrator) can help you locate suitable files that contain JDBC drivers.

5 Click Open to select the JDBC driver file. The Specify JDBC Driver Location dialog displays again, this time with the file location entered. Click Next to continue.

Figure 3-25. Specify JDBC Driver Location 2 - WebSphere Server Installation



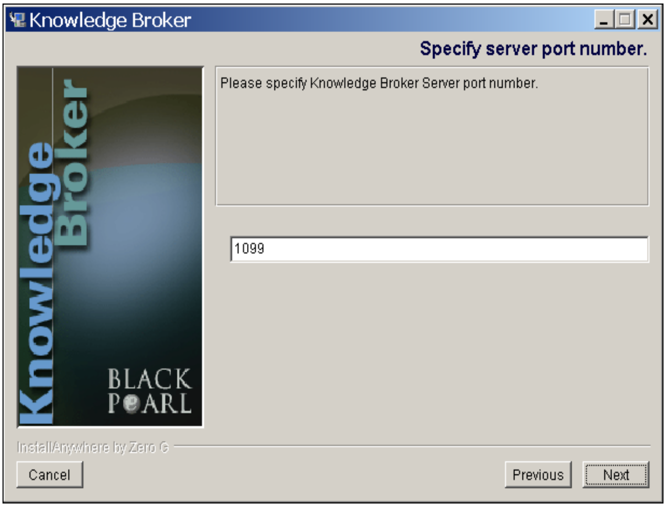
Specify JDBC driver location.

You may have a JDBC-enabled RDBMS currently installed. Use this dialog to select the driver file.

Please Choose a File: D:\orant\jdbc\lib\classes111.zip

6 After all files install, the Installation Complete dialog displays. Click Next. The Specify Server Port Number dialog displays. Specify Server Port Number.

Figure 3-26. Specify Server Port Number - WebSphere Server Installation



Specify server port number.

Please specify Knowledge Broker Server port number.

1099

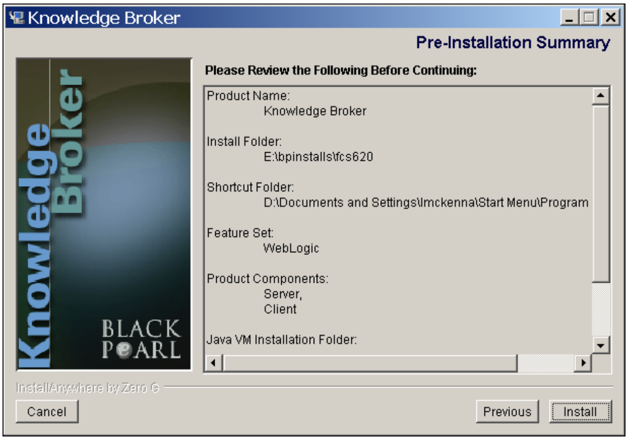
7 Click Next. The Choose WebSphere App Server Root Folder dialog displays. Complete one of the following options:

a Accept the default installation settings. By default, all Knowledge Broker files install to the `C:\BlackPearl\` folder (Windows) or the `/usr/blackpearl` directory (Solaris).

b Type the exact path to designate a destination folder. By default, all Knowledge Broker's files install to this folder.

8 Click Next. The Pre-Installation Summary displays.

Figure 3-27. Pre-Installation Summary - WebSphere Server Installation



Pre-Installation Summary

Please Review the Following Before Continuing:

Product Name:
Knowledge Broker

Install Folder:
E:\bpinstalls\fcs620

Shortcut Folder:
D:\Documents and Settings\lmckenna\Start Menu\Program

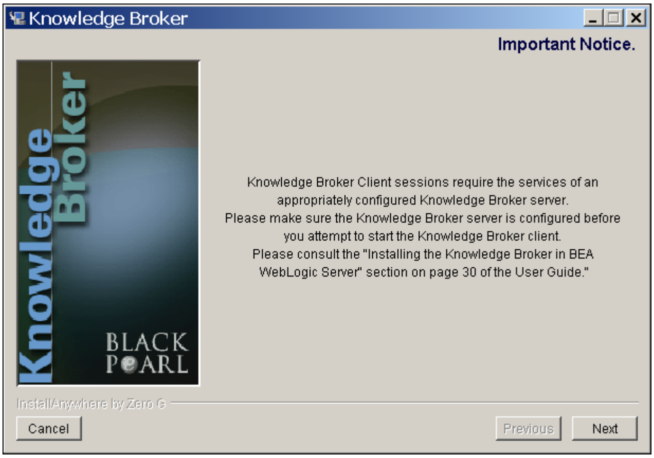
Feature Set:
WebLogic

Product Components:
Server,
Client

Java VM Installation Folder:

9 Click Install. The Knowledge Broker is installed. An Important notice displays.

Figure 3-28. Important Notice - WebSphere Server Installation

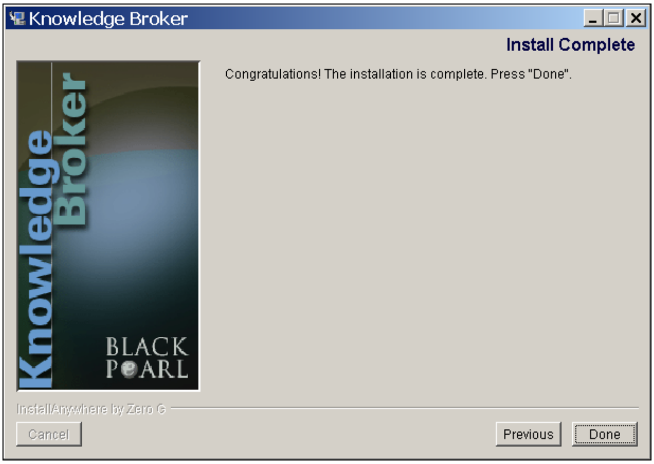


Important Notice.

Knowledge Broker Client sessions require the services of an appropriately configured Knowledge Broker server.
Please make sure the Knowledge Broker server is configured before you attempt to start the Knowledge Broker client.
Please consult the "Installing the Knowledge Broker in BEA WebLogic Server" section on page 30 of the User Guide."

10 Click Next. The Install Complete dialog displays.

Figure 3-29. Install Complete - WebSphere Server Installation



Install Complete

Congratulations! The installation is complete. Press "Done".

11 Click Done. You have now installed Knowledge Broker. Click Done.

Deploying Knowledge Broker in BEA WebLogic 6.1 Server

Knowledge Broker supports installation within the BEA WebLogic Server version 6.1 according to the EJB 1.1 specification. You should refer to the WebLogic Server support pages (http://www.weblogic.com/docs61/) to ensure your system exceeds the minimum hardware and software installation requirements.

To deploy Knowledge Broker as an EJB within WebLogic Server EJB, you must install the `knowledgebroker.ear` and `KBRuntime.war` files. These contain deployment descriptors and classes in accordance with the EJB 1.1 specification.

1 Edit the WebLogic `startWebLogic.cmd` (Windows) or `StartWebLogic.ksh` (Solaris) file to include references to Black Pearl's Java classes. The commands you should include are (modified to suit your own installation configuration):

```
set blackpearl_root=d:\[KB_installed_home]\
set BPPATH=%blackpearl_root%


set BPPATH=%BPPATH%;%blackpearl_root%lib\core.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\stub.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\advisor.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\commandhandler.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\container.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\calculator.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\blackpearl_javax.jar

set BPPATH=%BPPATH%;%blackpearl_root%lib\j2ee.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\Jaxen.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\fscontext.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\VisualNumerics.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\xerces.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\xercesSamples.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\log4j.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\jtidy.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\jdom.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\dom4j-full.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\sfc.jar
set BPPATH=%BPPATH%;%blackpearl_root%lib\knowledgebroker.ear
set BPPATH=%BPPATH%;%blackpearl_root%lib\KBRuntime.war

set CLASSPATH=.;.\lib\weblogic_sp.jar;.\lib\weblogic.jar;%BPPATH%
```

2 Start Weblogic by running the WebLogic `startWebLogic.cmd` (Windows) or `StartWebLogic.ksh` (Solaris).

3 Start an Internet browser window. Type this URL:

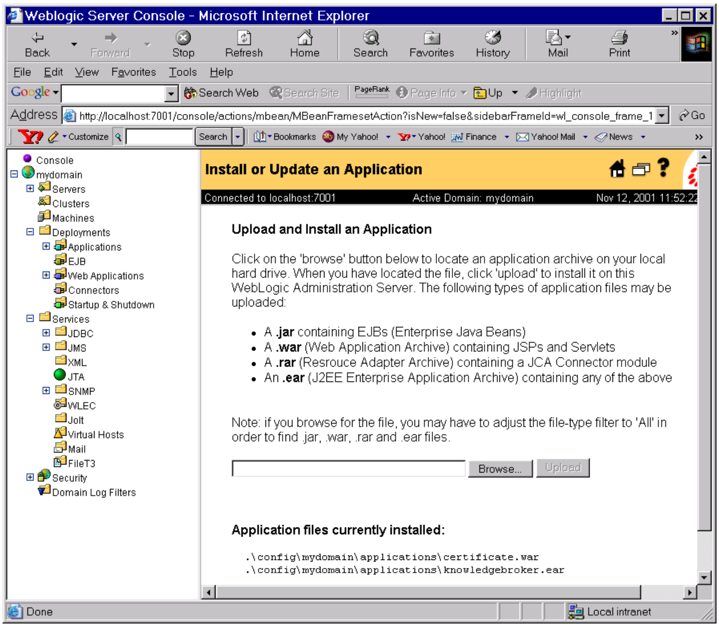
```
http://localhost:7001/console
```

The WebLogic server and console displays.

4 On the left-hand panel of the WebLogic console, navigate to **`My Domain→Deployments→Applications`**.

5 Select Install a New Application. The Install or Update an Application window displays.

Figure 3-30. WebLogic Install or Update an Application - WebLogic Server Installation



http://localhost:7001/console/actions/mbean/MBeanFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_1

Install or Update an Application

Connected to localhost:7001 Active Domain: mydomain Nov 12, 2001 11:52:22

Upload and Install an Application

Click on the 'browse' button below to locate an application archive on your local hard drive. When you have located the file, click 'upload' to install it on this WebLogic Administration Server. The following types of application files may be uploaded:

- A **.jar** containing EJBs (Enterprise Java Beans)
- A **.war** (Web Application Archive) containing JSPs and Servlets
- A **.rar** (Resrouce Adapter Archive) containing a JCA Connector module
- An **.ear** (J2EE Enterprise Application Archive) containing any of the above

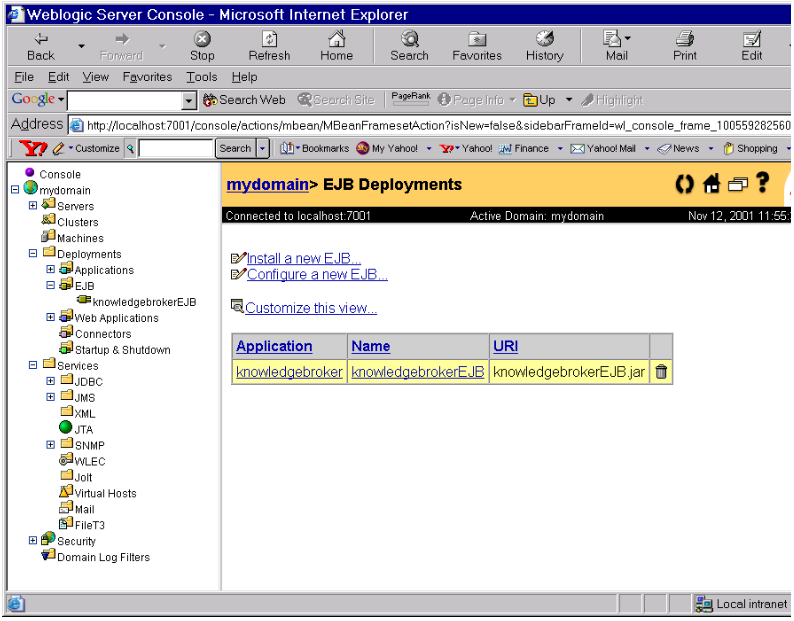
Note: if you browse for the file, you may have to adjust the file-type filter to 'All' in order to find .jar, .war, .rar and .ear files.

Application files currently installed:

```
.\config\mydomain\applications\certificate.war
.\config\mydomain\applications\knowledgebroker.ear
```

6 In the right-hand panel, click Browse to navigate to the directory where you have installed Knowledge Broker. Select `knowledgebroker.ear` and click Upload. Knowledge Broker displays in Weblogic.

Figure 3-31. WebLogic mydomain⟶EJB Deployments - WebLogic Server Installation



http://localhost:7001/console/actions/mbean/MBeanFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_100559282560

mydomain> EJB Deployments

Connected to localhost:7001 Active Domain: mydomain Nov 12, 2001 11:55:

Application	Name	URI	
knowledgebroker	knowledgebrokerEJB	knowledgebrokerEJB.jar	🗑

The following directory contains example files suitable for modification to suit your local environment:

```
[InstallDirectory]/appservers/weblogic
```

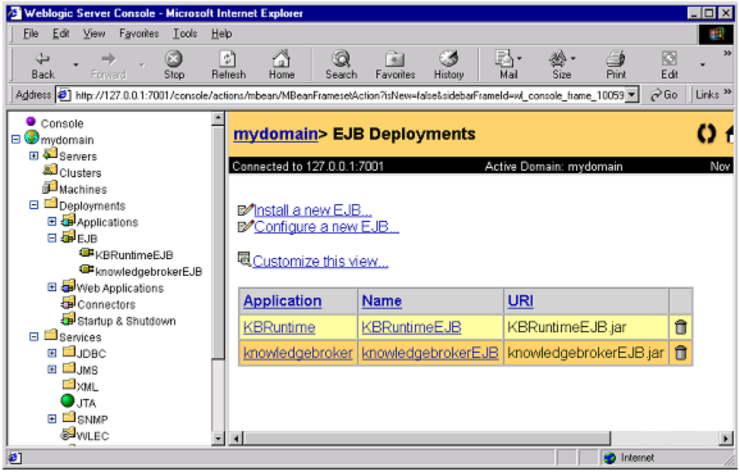
Now you must deploy `KBRuntime.war file`.

7 Repeat above steps 4-6.

An EAR file is a J2EE-standard method of application packaging. The KBRuntime.war packages the necessary components to provide runtime web functionality (that is, HTTP support) for Knowledge Broker within Weblogic.

8 Knowledge Broker files display within WebLogic.

Figure 3-32. WebLogic mydomain→EJB Deployments Display - WebLogic Server Installation


Weblogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print Edit
Address http://127.0.0.1:7001/console/actions/mbean/MBeanFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_10059
Go Links
Console
mydomain
Servers
Clusters
Machines
Deployments
Applications
EJB
KBRuntimeEJB
knowledgebrokerEJB
Web Applications
Connectors
Startup & Shutdown
Services
JDBC
JMS
XML
JTA
SNMP
WLEC
Internet

mydomain> EJB Deployments

Connected to 127.0.0.1:7001 Active Domain: mydomain Nov

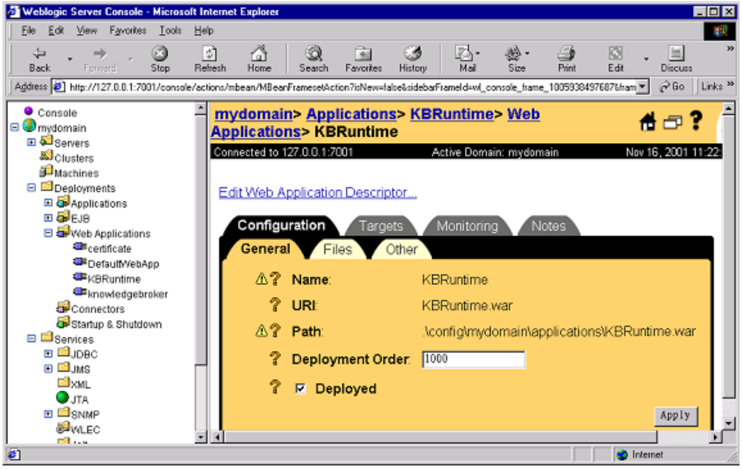
Install a new EJB...
Configure a new EJB...

Customize this view...

Application	Name	URI	
KBRuntime	KBRuntimeEJB	KBRuntimeEJB.jar	🗑
knowledgebroker	knowledgebrokerEJB	knowledgebrokerEJB.jar	🗑

9 Within the left-hand panel of the WebLogic console, navigate to **`My Domain→Deployments→Web Applications→KBRuntime`**.

Figure 3-33. WebLogic KBRuntime- WebLogic Server Installation


Weblogic Server Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Size Print Edit Discuss
Address http://127.0.0.1:7001/console/actions/mbean/MBeanFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_1005938497687&fram
Go Links
Console
mydomain
Servers
Clusters
Machines
Deployments
Applications
EJB
Web Applications
certificate
DefaultWebApp
KBRuntime
knowledgebroker
Connectors
Startup & Shutdown
Services
JDBC
JMS
XML
JTA
SNMP
WLEC
Internet

mydomain> Applications> KBRuntime> Web Applications> KBRuntime

Connected to 127.0.0.1:7001 Active Domain: mydomain Nov 16, 2001 11:22

Edit Web Application Descriptor...

Configuration
Targets
Monitoring
Notes
General
Files
Other

Name: KBRuntime

URI: KBRuntime.war

Path: .\config\mydomain\applications\KBRuntime.war

Deployment Order: 1000

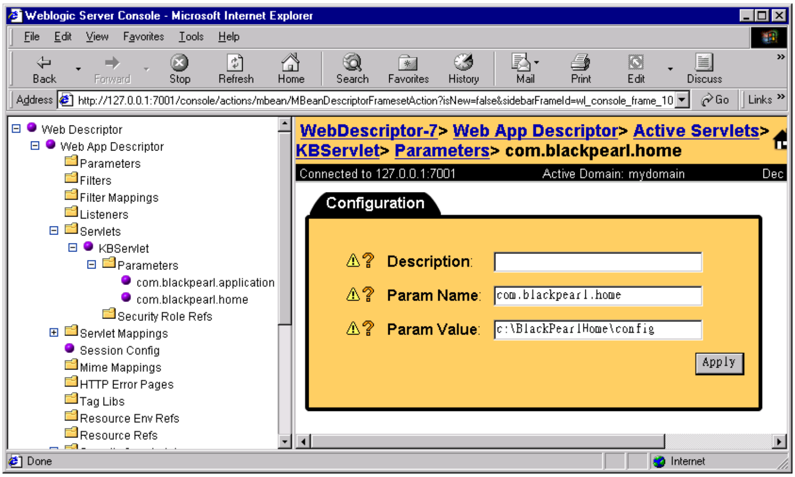
☑ Deployed

10 Within the right-hand panel, select Edit Web Application Descriptor.

11 On the left-hand panel of the WebLogic console, navigate to
`Web Descriptor→Web App Descriptor`
`→Servlets→KBServlet→Parameters→com.blackpearl.home`.

12 Type the location of the `KBhome` directory in the **Param Value** field.

Figure 3-34. WebLogic com.blackpearl.com - WebLogic Server Installation



http://127.0.0.1:7001/console/actions/mbean/MBeanDescriptorFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_10

WebDescriptor-7> Web App Descriptor> Active Servlets> KBServlet> Parameters> com.blackpearl.home

Connected to 127.0.0.1:7001 Active Domain: mydomain Dec

Description:

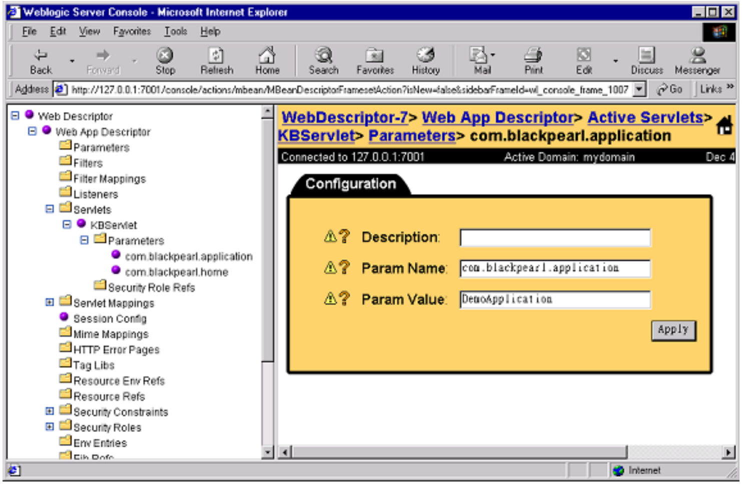
Param Name: com.blackpearl.home

Param Value: c:\BlackPearlHome\config

13 Click Apply.

14 On the left-hand panel of the WebLogic console, navigate to **`Web Descriptor→Web App Descriptor →Servlets→KBServlet→Parameters→com.blackpearl.application`**.

Figure 3-35. WebLogic com.blackpearl.application - WebLogic Server Installation



http://127.0.0.1:7001/console/actions/mbean/MBeanDescriptorFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_1007

WebDescriptor-7> Web App Descriptor> Active Servlets> KBServlet> Parameters> com.blackpearl.application

Connected to 127.0.0.1:7001 Active Domain: mydomain Dec 4

Description:

Param Name: com.blackpearl.application

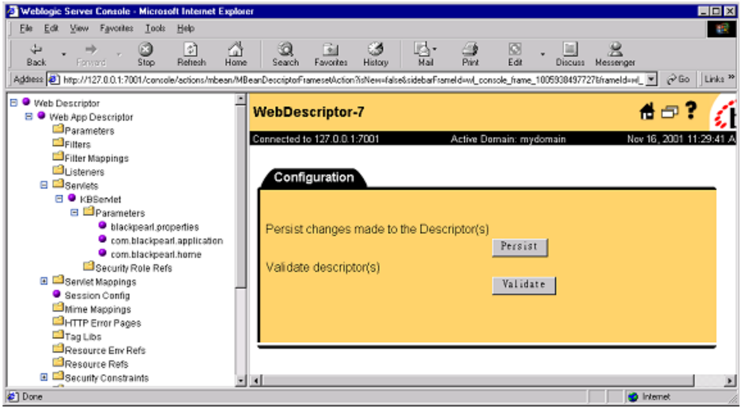
Param Value: DemoApplication

15 Type the name of the KnowledgeBase.

16 Click Apply.

17 On the left-hand panel of the WebLogic console, navigate to `Web App Descriptor.`

Figure 3-36. WebLogic Web Descriptor - WebLogic Server Installation



Weblogic Server Console - Microsoft Internet Explorer

Address http://127.0.0.1:7001/console/actions/mbean/MBeanDescriptorFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_1005938497727&frameId=wl_

WebDescriptor-7

Connected to 127.0.0.1:7001 Active Domain: mydomain Nov 16, 2001 11:29:41 A

Configuration

Persist changes made to the Descriptor(s)

Validate descriptor(s)

18 Click Persist. You have now set the values for Knowledge Broker initialization properties.

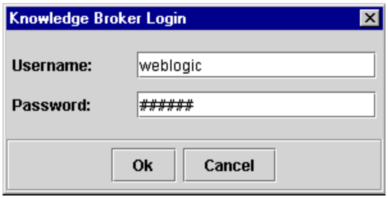
19 On the left-hand panel of the WebLogic console, navigate to **`My Domain→Deployments→Web Applications→Knowledgebroker.`** To edit the parameters repeat steps 10-18 for the Knowledgebroker.ear.

There is a problem with persisting the application descriptor. You must wait before closing the WebLogic windows.

Using Login Credentials for Users

1 Start Knowledge Broker Client. Knowledge Broker's Login window displays.

Figure 3-37. WebLogic Knowledge Broker Login



Knowledge Broker Login

Username: weblogic

Password: ######

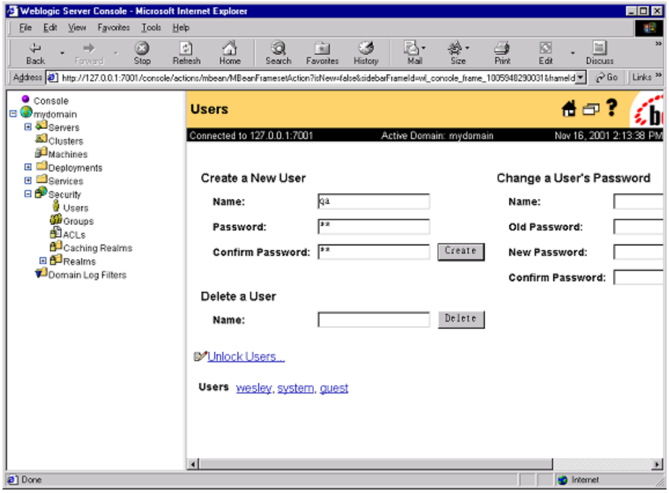
2 Type Username and Password.

3 Click Ok.

Creating Additional Users and Credentials

1 To create an additional user name, within the left-hand panel, navigate to **`My Domain`→`Security`→`Users`.**

Figure 3-38. WebLogic Users



Address http://127.0.0.1:7001/console/actions/mbean/MBeanFramesetAction?isNew=false&sidebarFrameId=wl_console_frame_1005948290031&frameId

Users

Connected to 127.0.0.1:7001 Active Domain: mydomain Nov 16, 2001 2:13:38 PM

Create a New User

Name: qa

Password: **

Confirm Password: **

Change a User's Password

Name:

Old Password:

New Password:

Confirm Password:

Delete a User

Name:

Unlock Users...

Users wesley, system, guest

2 Type the desired Username and Password. Confirm Password and click Create.

Deleting Users and Credentials

1 To delete a user name type the user name you want to delete in the Delete a User Name field and click Delete.

Modifying Users and Credentials

1 To change a User's Password type the Name, Old Password, New Password in the appropriate fields. Type the New Password in the Confirm Password field to confirm the new password.

Starting the Client

1 Execute the batch file `[KB_Install_Directory]\bin\client.bat` (Windows) or `[KB_Install_Directory]/client.ksh` (Solaris). Knowledge Broker's Login window displays.

2 Type your WebLogic user name and password.

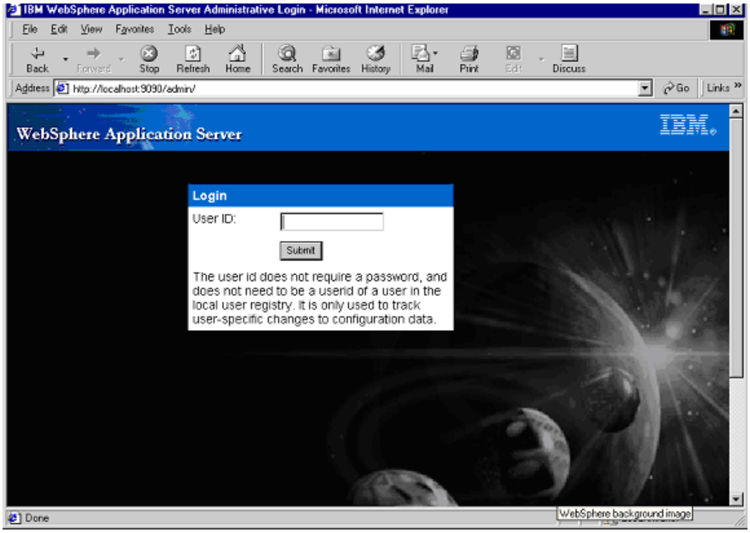
3 Click OK to continue.

Deploying Knowledge Broker in WebSphere

1 Start the WebSphere server. Start the Admin Console by typing the following url into a web browser: `http://localhost:9080/admin`

The WebSphere Application Server dialog displays.

Figure 3-39. WebSphere Application Server- WebSphere Server Installation


IBM WebSphere Application Server Administrative Login - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Discuss
Address
http://localhost:9090/admin/
Go
Links
WebSphere Application Server
IBM
Done
WebSphere background image

Login

User ID:

Submit

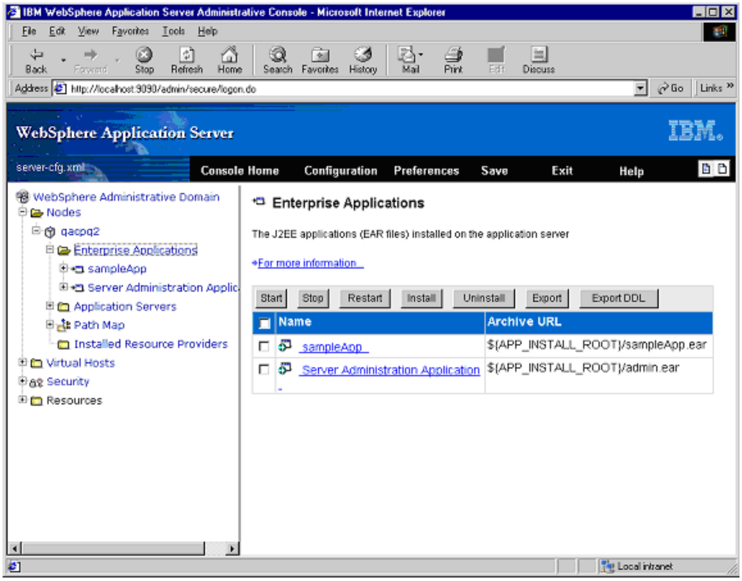
The user id does not require a password, and does not need to be a userid of a user in the local user registry. It is only used to track user-specific changes to configuration data.

2 Type a user id in the User ID field. The user is does not require a password, and does not need to be a userid of a user in the local user registry. It is only used to track user-specific changes to configuration data.

3 Click Submit. The Enterprise Applications window displays.

4 On the left-hand panel of the WebLogic console, navigate to **`Nodes`**→**`Wqacpq2`**→ **`Enterprise Applications`**.

Figure 3-40. Enterprise Applications - WebSphere Server Installation


IBM WebSphere Application Server Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:9090/admin/secure/logon.do
Go Links
WebSphere Application Server
IBM.
server-cfg.xml
Console Home Configuration Preferences Save Exit Help
WebSphere Administrative Domain
Nodes
qacpq2
Enterprise Applications
sampleApp
Server Administration Applic
Application Servers
Path Map
Installed Resource Providers
Virtual Hosts
Security
Resources
Start Stop Restart Install Uninstall Export Export DDL
Local intranet

Enterprise Applications

The J2EE applications (EAR files) installed on the application server

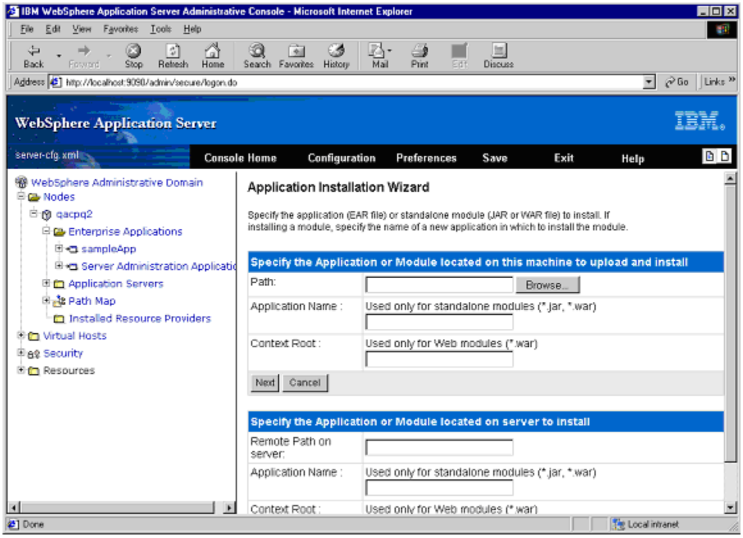
For more information

	Name	Archive URL
□	sampleApp	${APP_INSTALL_ROOT}/sampleApp.ear
□	Server Administration Application	${APP_INSTALL_ROOT}/admin.ear

5 Click Install.

6 The Application Installation Wizard Displays. Click Browser to select `KnowledgeBrokerWS.ear` [WS_HOME]\AppServer\installableApps.

Figure 3-41. Application Installation Wizard - WebSphere Server Installation


IBM WebSphere Application Server Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:9090/admin/secure/logon.do
Go Links »
WebSphere Application Server
IBM.
server-cfg.xml
Console Home Configuration Preferences Save Exit Help
WebSphere Administrative Domain
Nodes
qacpq2
Enterprise Applications
sampleApp
Server Administration Applicatio
Application Servers
Path Map
Installed Resource Providers
Virtual Hosts
Security
Resources
Browse...
Next Cancel
Done
Local intranet

Application Installation Wizard

Specify the application (EAR file) or standalone module (JAR or WAR file) to install. If installing a module, specify the name of a new application in which to install the module.

Specify the Application or Module located on this machine to upload and install

Path:

Application Name : Used only for standalone modules (*.jar, *.war)

Context Root : Used only for Web modules (*.war)

Specify the Application or Module located on server to install

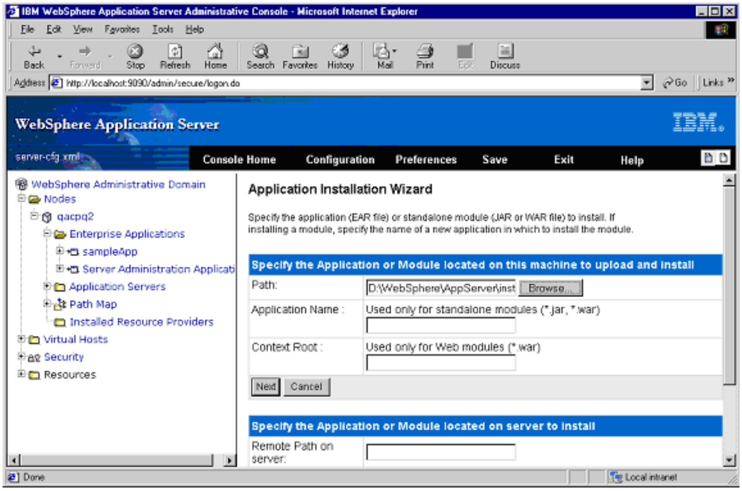
Remote Path on server:

Application Name : Used only for standalone modules (*.jar, *.war)

Context Root : Used only for Web modules (*.war)

7 Click Next. The Application Installation Wizard displays again.

Figure 3-42. Application Installation Wizard2 - WebSphere Server Installation


IBM WebSphere Application Server Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:9090/admin/secure/logon.do
Go Links
WebSphere Application Server
IBM.
server-cfg.xml
Console Home Configuration Preferences Save Exit Help
WebSphere Administrative Domain
Nodes
qacpq2
Enterprise Applications
sampleApp
Server Administration Applicati
Application Servers
Path Map
Installed Resource Providers
Virtual Hosts
Security
Resources
Done
Local intranet

Application Installation Wizard

Specify the application (EAR file) or standalone module (JAR or WAR file) to install. If installing a module, specify the name of a new application in which to install the module.

Specify the Application or Module located on this machine to upload and install

Path: D:\WebSphere\AppServer\inst Browse...

Application Name : Used only for standalone modules (*.jar, *.war)

Context Root : Used only for Web modules (*.war)

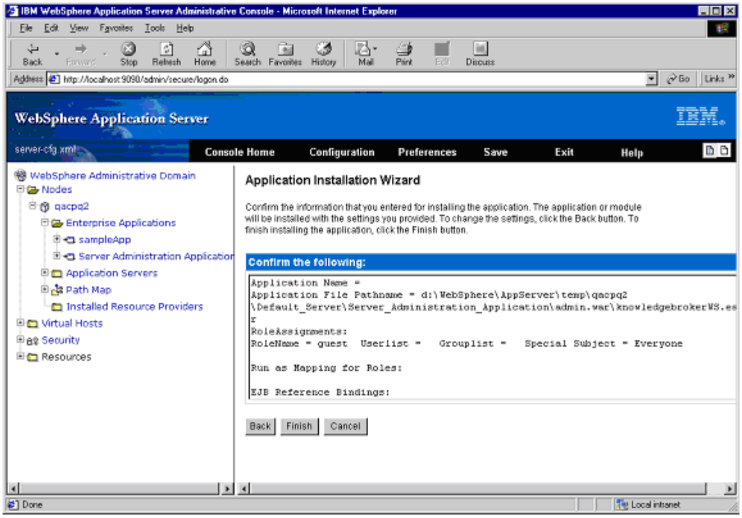
Next Cancel

Specify the Application or Module located on server to install

Remote Path on server:

8 Click Next through the Application Installation Window until the Confirm the Following window displays.

Figure 3-43. Application Installation Wizard Confirmation - WebSphere Server Installation


IBM WebSphere Application Server Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Forward
Stop
Refresh
Home
Search
Favorites
History
Mail
Print
Edit
Discuss
Address http://localhost:9090/admin/secure/logon.do
Go
Links
WebSphere Application Server
IBM
server-cfg.xml
Console Home
Configuration
Preferences
Save
Exit
Help
WebSphere Administrative Domain
Nodes
qacpq2
Enterprise Applications
sampleApp
Server Administration Applicatior
Application Servers
Path Map
Installed Resource Providers
Virtual Hosts
Security
Resources
Back
Finish
Cancel
Done
Local intranet

Application Installation Wizard

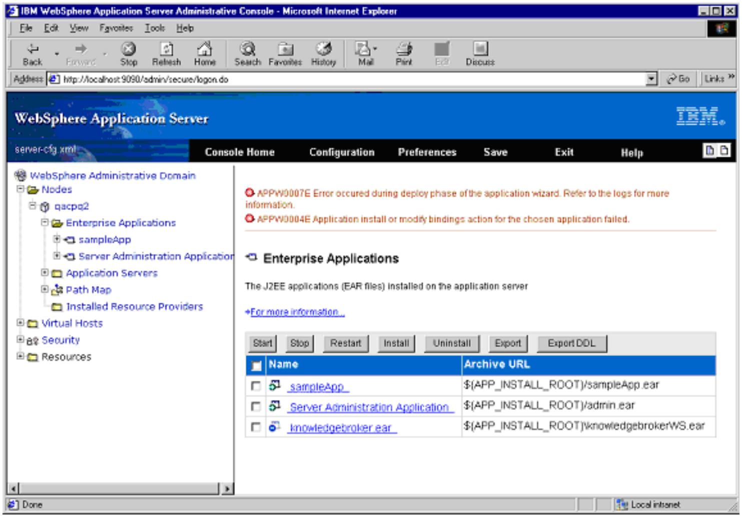
Confirm the information that you entered for installing the application. The application or module will be installed with the settings you provided. To change the settings, click the Back button. To finish installing the application, click the Finish button.

Confirm the following:

```
Application Name =
Application File Pathname = d:\WebSphere\AppServer\temp\qacpq2
\Default_Server\Server_Administration_Application\admin.war\knowledgebrokerWS.ea
r
RoleAssignments:
RoleName = guest  Userlist =   Grouplist =   Special Subject = Everyone

Run as Mapping for Roles:

EJB Reference Bindings:
```

9 Click Finish. The Enterprise Application window displays.

Figure 3-44. Enterprise Applications - WebSphere Server Installation


IBM WebSphere Application Server Administrative Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http://localhost:9090/admin/secure/logon.do
Go Links
WebSphere Application Server
IBM.
server-cfg.xml
Console Home Configuration Preferences Save Exit Help
WebSphere Administrative Domain
Nodes
qacpq2
Enterprise Applications
sampleApp
Server Administration Applicatior
Application Servers
Path Map
Installed Resource Providers
Virtual Hosts
Security
Resources
Done
Local intranet

APPW0007E Error occured during deploy phase of the application wizard. Refer to the logs for more information.

APPW0004E Application install or modify bindings action for the chosen application failed.

Enterprise Applications

The J2EE applications (EAR files) installed on the application server

For more information...

Start Stop Restart Install Uninstall Export Export DDL

☐	Name	Archive URL
☐	sampleApp	${APP_INSTALL_ROOT}/sampleApp.ear
☐	Server Administration Application	${APP_INSTALL_ROOT}/admin.ear
☐	knowledgebroker.ear	${APP_INSTALL_ROOT}\knowledgebrokerWS.ear

Ignore the error messages.

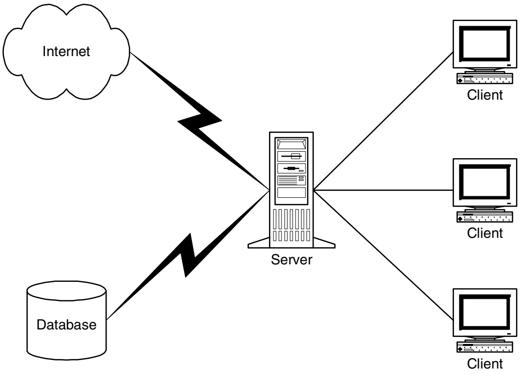
You have just deployed the `knowledgebrokerWS.ear`.

10 Save changes.

Client Installation

The Client option installs all Server libraries. During installation, you must enter a URL to locate the WebLogic Server where Knowledge Broker Server has been deployed. All the XML model and configuration files are stored remotely on the Server. Although Client users can add, modify, and delete datasources, there are multi-user restrictions concerning this functionality.

Figure 3-45. Client Installation



Server Root Directory

The client installation requires the root directory of your Blackpearl server installation. When prompted during the client installation for the server installation location, either choose or type the directory path. You should include only the path to the top-level server installation. For more information on locating the server directory, see the information about identifying the *Standalone Installation on page 62*.

Installation Sequence
Install the server first, followed by the client. For convenience, within Microsoft Windows, you can map the server's installation directory to a network directory and note its name.

Installing Knowledge Broker Client

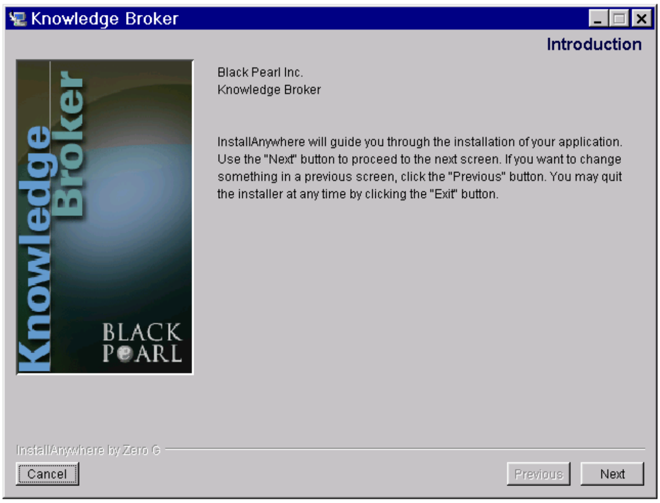
Knowledge Broker's installation process enables you to select from a set of options. After you select the appropriate options, installation proceeds automatically.

1 Complete one of the following options:

a If you downloaded Knowledge Broker, open the directory containing the application file and double-click `broker.exe`. The InstallAnywhere application will automatically run and Knowledge Broker's Introduction dialog displays.

b If you received Knowledge Broker on a CD-ROM, insert the CD into the appropriate CD-ROM drive. The InstallAnywhere application will run and Knowledge Broker's Introduction dialog displays.

Figure 3-46. Knowledge Broker Introduction - Client Installation

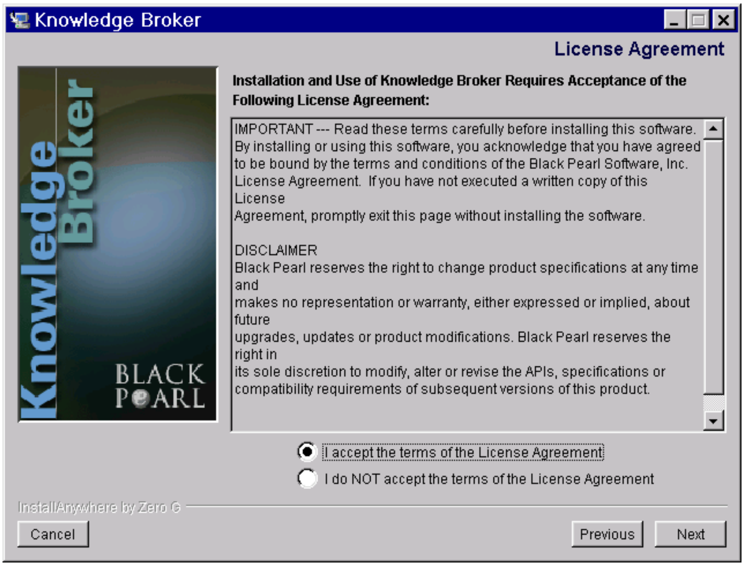


Black Pearl Inc.
Knowledge Broker

InstallAnywhere will guide you through the installation of your application. Use the "Next" button to proceed to the next screen. If you want to change something in a previous screen, click the "Previous" button. You may quit the installer at any time by clicking the "Exit" button.

2 Click Next. The License Agreement dialog displays.

Figure 3-47. License Agreement - Client Installation



Installation and Use of Knowledge Broker Requires Acceptance of the Following License Agreement:

IMPORTANT --- Read these terms carefully before installing this software. By installing or using this software, you acknowledge that you have agreed to be bound by the terms and conditions of the Black Pearl Software, Inc. License Agreement. If you have not executed a written copy of this License
Agreement, promptly exit this page without installing the software.

DISCLAIMER
Black Pearl reserves the right to change product specifications at any time and
makes no representation or warranty, either expressed or implied, about future
upgrades, updates or product modifications. Black Pearl reserves the right in
its sole discretion to modify, alter or revise the APIs, specifications or compatibility requirements of subsequent versions of this product.

● I accept the terms of the License Agreement
○ I do NOT accept the terms of the License Agreement

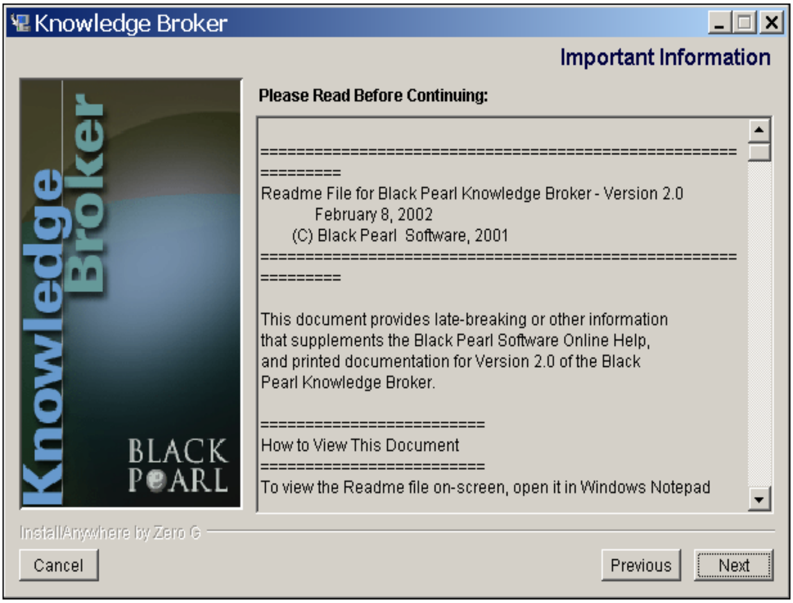
3 Select one of the following options:

   a If you do not accept Knowledge Broker's license agreement, click Exit. Click Yes when the Confirm Exit dialog displays.

   b If you accept Knowledge Broker's license agreement, select Yes.

4 Click Next. The Important Information dialog displays.

Figure 3-48. Important Information - Client Installation



Please Read Before Continuing:

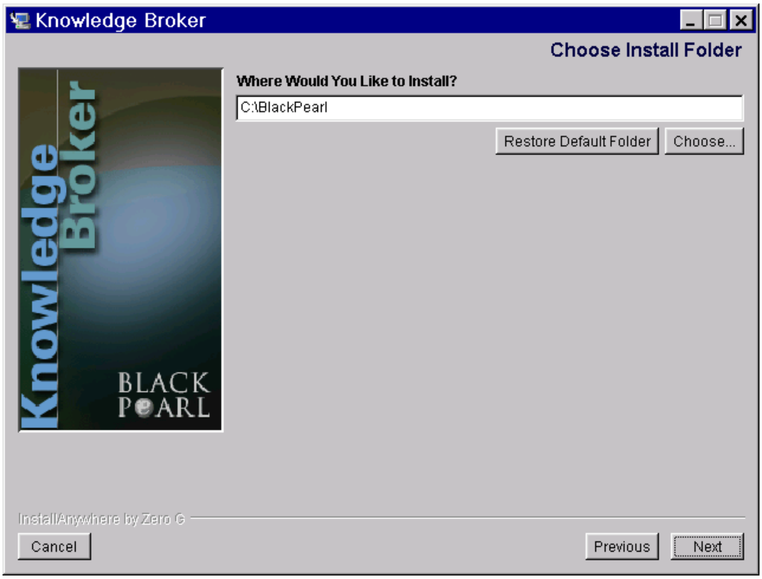
```
=====================================================
=========
Readme File for Black Pearl Knowledge Broker - Version 2.0
         February 8, 2002
     (C) Black Pearl  Software, 2001
=====================================================
=========

This document provides late-breaking or other information
that supplements the Black Pearl Software Online Help,
and printed documentation for Version 2.0 of the Black
Pearl Knowledge Broker.

=========================
How to View This Document
=========================
To view the Readme file on-screen, open it in Windows Notepad
```

5 (Optional) Scroll down the Important Information dialog to review the Readme file prior to installation of Knowledge Broker.

After installing Knowledge Broker, you may access the Readme file by opening file in a standard text editor such as Notepad.

6 Click Next. The Choose Install Folder dialog displays.

Figure 3-49. Choose Install Folder - Client Installation



Where Would You Like to Install?

C:\BlackPearl

7 Complete one of the following options:

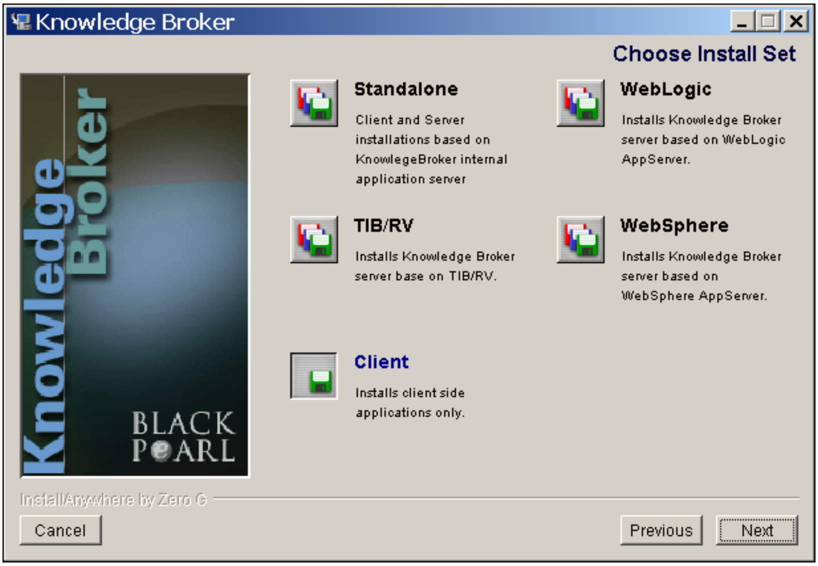
a Accept the default installation settings. By default, all Knowledge Broker files install to the `C:\BlackPearl\` folder (Windows) or the `/usr/blackpearl` directory (Solaris).

b Type the exact path to designate a destination folder. By default, all Knowledge Broker files install to this folder.

c Click Choose and use the file selection dialog to select a drive and folder. By default, all Knowledge Broker files install to this folder.

8 Click Next. The Choose Install Set dialog displays.

Figure 3-50. Choose Install Set - Client Installation



Standalone
Client and Server installations based on KnowlegeBroker internal application server

WebLogic
Installs Knowledge Broker server based on WebLogic AppServer.

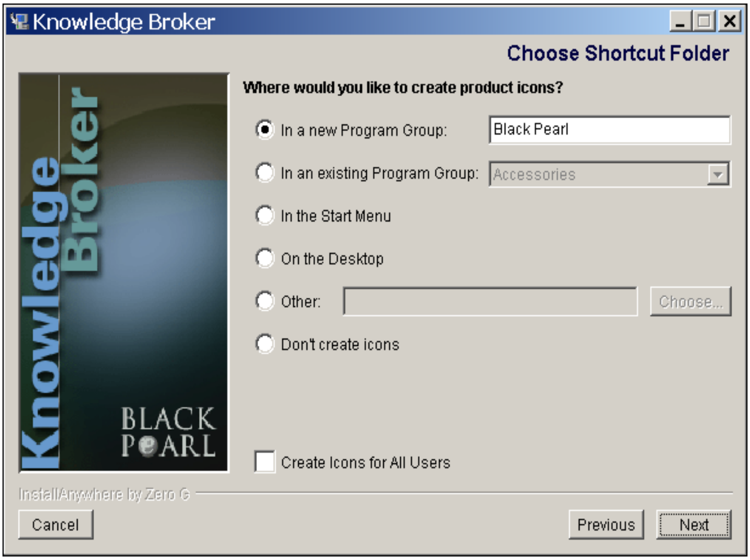
TIB/RV
Installs Knowledge Broker server base on TIB/RV.

WebSphere
Installs Knowledge Broker server based on WebSphere AppServer.

Client
Installs client side applications only.

To install only the Client components, select Client and click Next. Knowledge Broker installs to your client machine. You must now choose a location for the product icon. The Choose Shortcut Folder displays.

Figure 3-51. Choose Shortcut Folder - Client Installation



Knowledge Broker

Choose Shortcut Folder

Where would you like to create product icons?

● In a new Program Group: Black Pearl

○ In an existing Program Group: Accessories

○ In the Start Menu

○ On the Desktop

○ Other: Choose...

○ Don't create icons

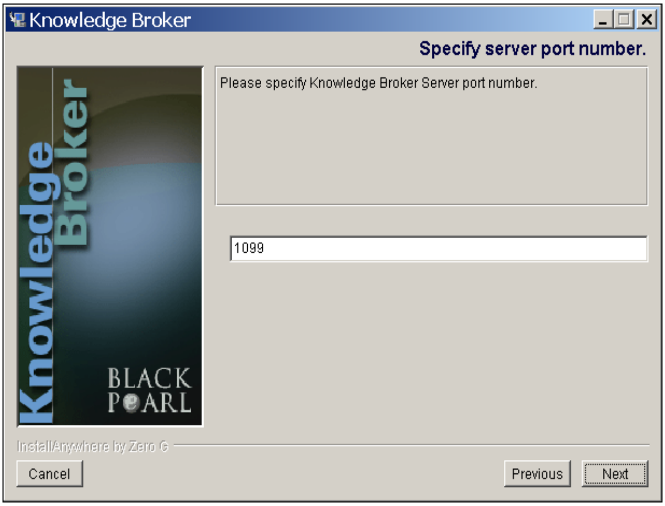
□ Create Icons for All Users

9 Select a location for the Knowledge Broker. Click Next. You must now tell the Black Pearl Client installation process where to find the Server installation components.

See the Readme for up-to-date information about your installation choices.

The Specify Server Port Number displays:

Figure 3-52. Specify Server Port Number - Client Installation



Specify server port number.

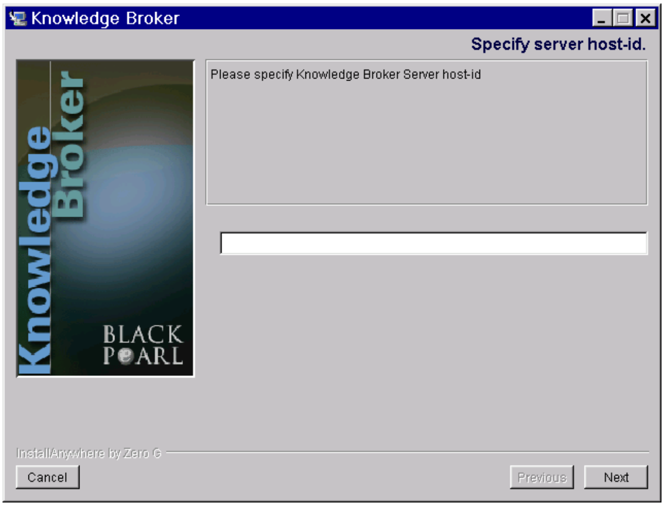
Please specify Knowledge Broker Server port number.

1099

i Specify the Server Port Number.

ii Click Next. The Specify Server Host-id dialog displays:

Figure 3-53. Specify Server Host-id - Client Installation



Specify server host-id.

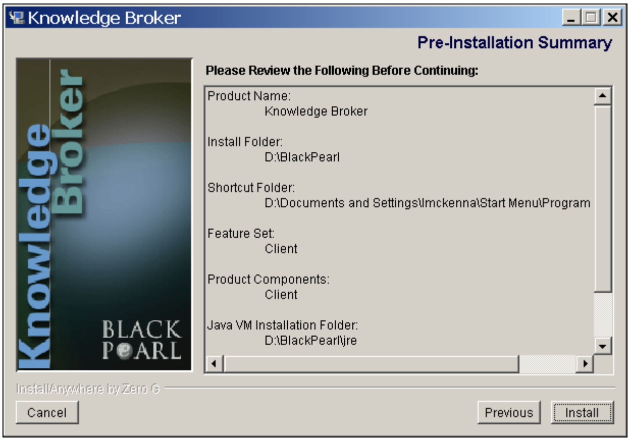
Please specify Knowledge Broker Server host-id

iii Type the name of the location of Knowledge Broker's Server host-id.

iv Click Next.

v Click Next. Knowledge Broker begins the installation of the Client. The Pre-Installation Summary displays.

Figure 3-54. Pre-installation Summary- Client Installation



Pre-Installation Summary

Please Review the Following Before Continuing:

Product Name:
Knowledge Broker

Install Folder:
D:\BlackPearl

Shortcut Folder:
D:\Documents and Settings\lmckenna\Start Menu\Program

Feature Set:
Client

Product Components:
Client

Java VM Installation Folder:
D:\BlackPearl\jre

10 Complete one of the following options:

a If you do not accept the Pre-Installation Summary, click Previous and correct the necessary information.

b If you accept the Pre-Installation Summary, click Install. Knowledge Broker Client is installed. Click Done.

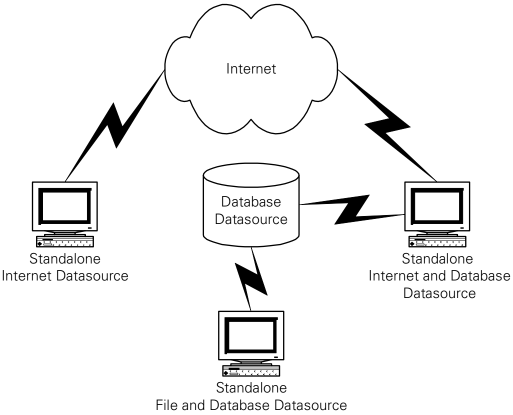
You have now installed Knowledge Broker Client.

Standalone Installation

The Standalone installation is a minimal installation. It is useful for quick prototyping or development. Selecting the Standalone option installs both the Server and Client components on a single machine. The Standalone user can access all Datasource Connection, Ontology Editor, Application/Agent Editor, Rule Editor, and Knowledge Discovery Center functionality. Because it doesn't leverage WebLogic functionality, as compared to the Server installation, the Standalone installation doesn't scale.

This option installs a minimal client and server configuration where all communication is done through Remote Method Invocation (RMI). In this case, `server.bat` (`server.ksh` in Solaris) invokes a local RMI handler, as well as Knowledge Broker Server functionality. Use `client.bat` (`client.ksh` in Solaris) to invoke Knowledge Broker Client functionality.

Figure 3-55. Standalone Installation


Internet
Database Datasource
Standalone Internet Datasource
Standalone Internet and Database Datasource
Standalone File and Database Datasource

Installing Knowledge Broker Standalone

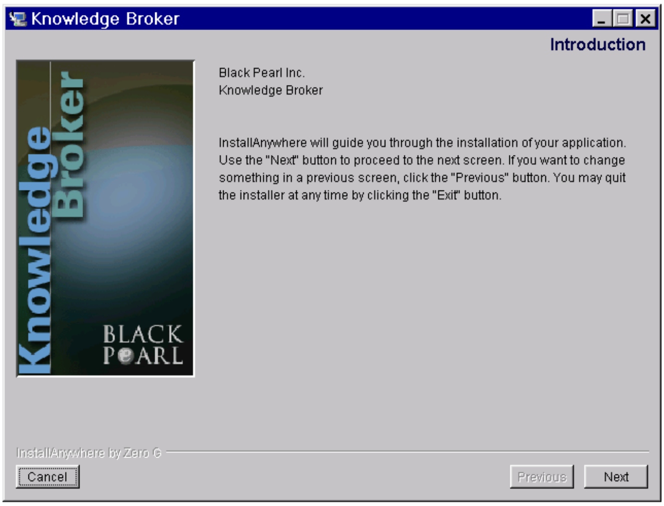
Knowledge Broker's installation process enables you to select from a set of options. After you select the appropriate options, installation proceeds automatically.

1 Complete one of the following options:

a If you downloaded Knowledge Broker, open the directory containing the application file and double-click `broker.exe`. The InstallAnywhere application will automatically run and Knowledge Broker's Introduction dialog displays.

b If you received Knowledge Broker on a CD-ROM, insert the CD into the appropriate CD-ROM drive. The InstallAnywhere application will run and Knowledge Broker's Introduction dialog displays.

Figure 3-56. Knowledge Broker Introduction - Standalone Installation

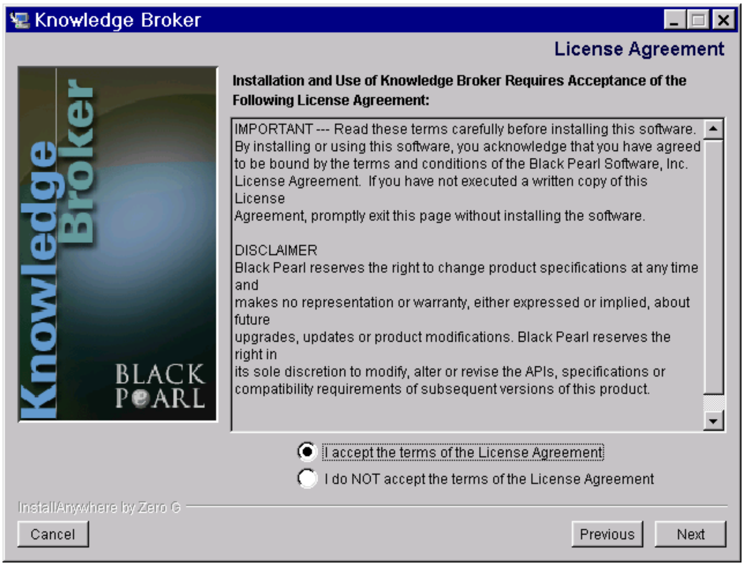


Black Pearl Inc.
Knowledge Broker

InstallAnywhere will guide you through the installation of your application. Use the "Next" button to proceed to the next screen. If you want to change something in a previous screen, click the "Previous" button. You may quit the installer at any time by clicking the "Exit" button.

2 Click Next. The License Agreement dialog displays.

Figure 3-57. License Agreement - Standalone Installation



Installation and Use of Knowledge Broker Requires Acceptance of the Following License Agreement:

IMPORTANT --- Read these terms carefully before installing this software. By installing or using this software, you acknowledge that you have agreed to be bound by the terms and conditions of the Black Pearl Software, Inc. License Agreement. If you have not executed a written copy of this License
Agreement, promptly exit this page without installing the software.

DISCLAIMER
Black Pearl reserves the right to change product specifications at any time and
makes no representation or warranty, either expressed or implied, about future
upgrades, updates or product modifications. Black Pearl reserves the right in
its sole discretion to modify, alter or revise the APIs, specifications or compatibility requirements of subsequent versions of this product.

● I accept the terms of the License Agreement

○ I do NOT accept the terms of the License Agreement

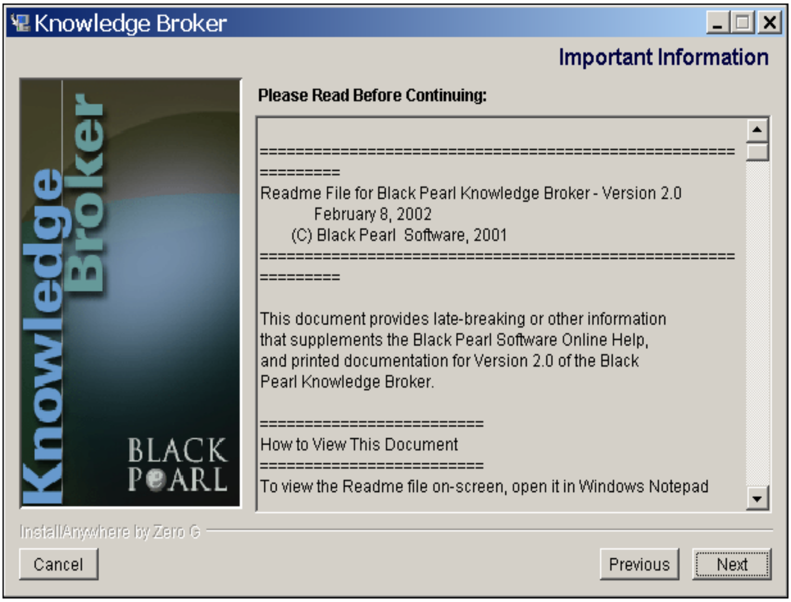
3 Select one of the following options:

   a If you do not accept Knowledge Broker's license agreement, click Exit. Click Yes when the Confirm Exit dialog displays.

   b If you accept Knowledge Broker's license agreement, select Yes.

4 Click Next. The Important Information dialog displays.

Figure 3-58. Important Information - Standalone Installation



Please Read Before Continuing:

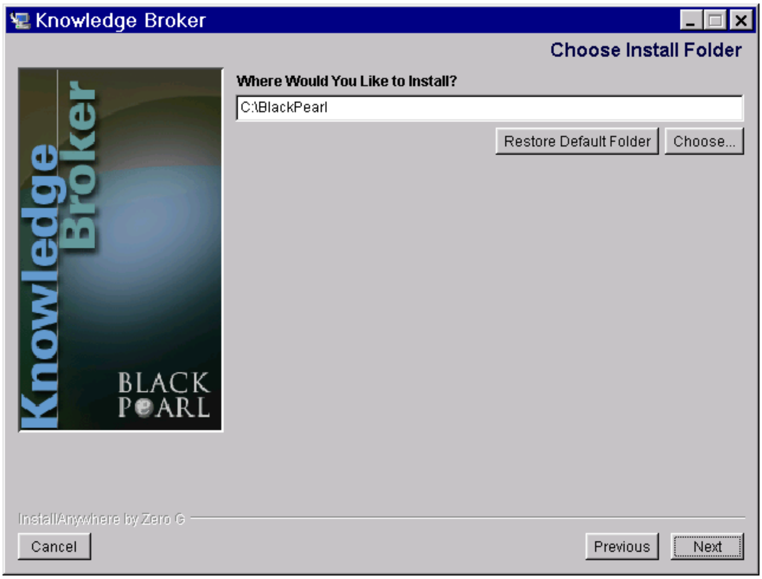
```
============================================================
==========
Readme File for Black Pearl Knowledge Broker - Version 2.0
          February 8, 2002
      (C) Black Pearl  Software, 2001
============================================================
==========

This document provides late-breaking or other information
that supplements the Black Pearl Software Online Help,
and printed documentation for Version 2.0 of the Black
Pearl Knowledge Broker.

============================
How to View This Document
============================
To view the Readme file on-screen, open it in Windows Notepad
```

5 (Optional) Scroll down the Important Information dialog to review the Readme file prior to installation of Knowledge Broker.

After installing Knowledge Broker, you may access the Readme file by opening file in a standard text editor such as Notepad.

6 Click Next. The Choose Install Folder dialog displays.

Figure 3-59. Choose Install Folder - Standalone Installation



Where Would You Like to Install?

C:\BlackPearl

7 Complete one of the following options:

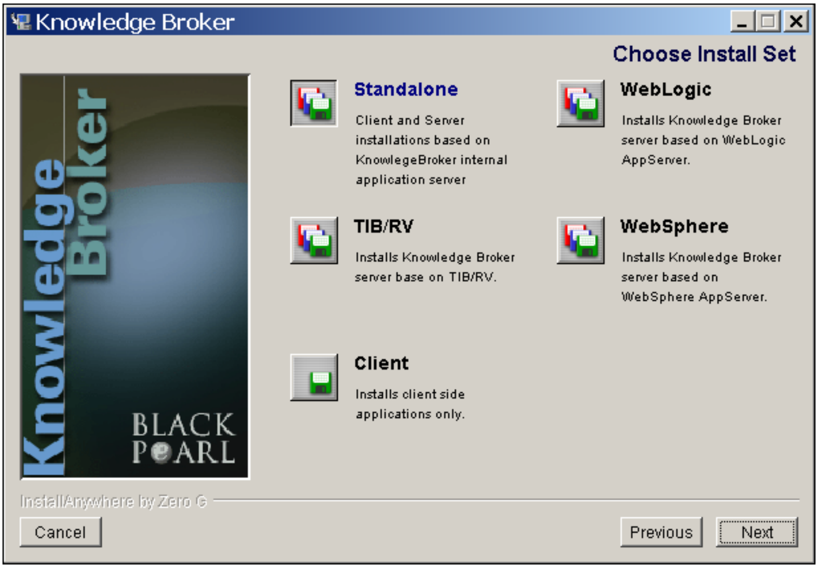
a Accept the default installation settings. By default, all Knowledge Broker's files install to the `C:\BlackPearl\` folder (Windows) or the `/usr/blackpearl` directory (Solaris).

b Type the exact path to designate a destination folder. By default, all Knowledge Broker's files install to this folder.

c Click Choose and use the file selection dialog to select a drive and folder. By default, all Knowledge Broker's files install to this folder.

8 Click Next. The Choose Install Set dialog displays.

Figure 3-60. Choose Install Set - Standalone Installation



Choose Install Set

Standalone
Client and Server installations based on KnowlegeBroker internal application server

WebLogic
Installs Knowledge Broker server based on WebLogic AppServer.

TIB/RV
Installs Knowledge Broker server base on TIB/RV.

WebSphere
Installs Knowledge Broker server based on WebSphere AppServer.

Client
Installs client side applications only.

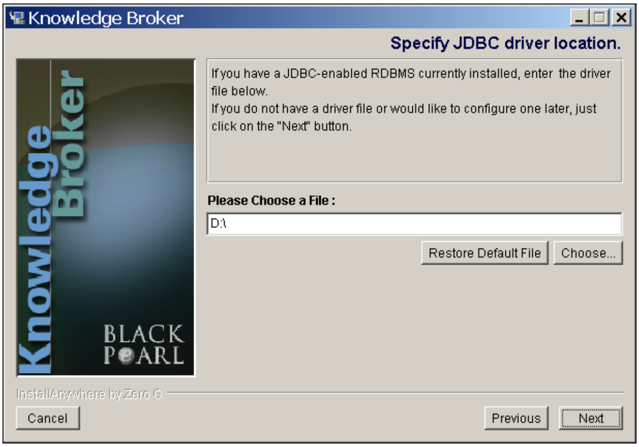
9 To install both the Server and Client components on a single machine, select Standalone and click Next. Knowledge Broker installs to your standalone machine. You must now tell the Black Pearl Client installation process where to find the Server installation components.

See the Readme for up-to-date information about your installation choices.

You can connect to one or more JDBC-compliant datasources (for more information, see *Configuring JDBC datasource Driver Locations on page 71*). Note that within Windows NT, you may instead use the Microsoft ODBC bridge mechanism (see *Configuring ODBC datasource Drivers on page 69*) and click Next to skip this section.

The Specify JDBC Driver Location dialog displays, enabling you to select a suitable JDBC driver.

Figure 3-61. Specify JDBC Driver Location - Standalone Installation

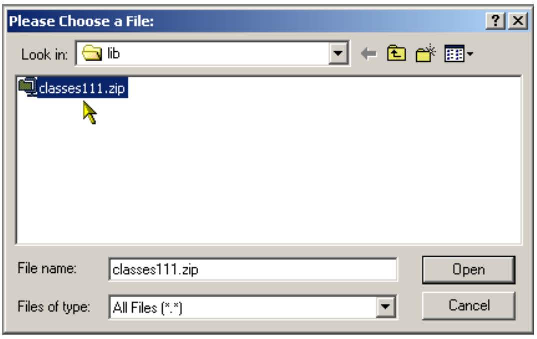


If you have a JDBC-enabled RDBMS currently installed, enter the driver file below.
If you do not have a driver file or would like to configure one later, just click on the "Next" button.

Please Choose a File : D:\

10 You can enter the file location, or click Choose to use the standard Windows file browser dialog to identify the file that contains a suitable JDBC driver:

Figure 3-62. File Locator - Standalone Installation



Look in: lib

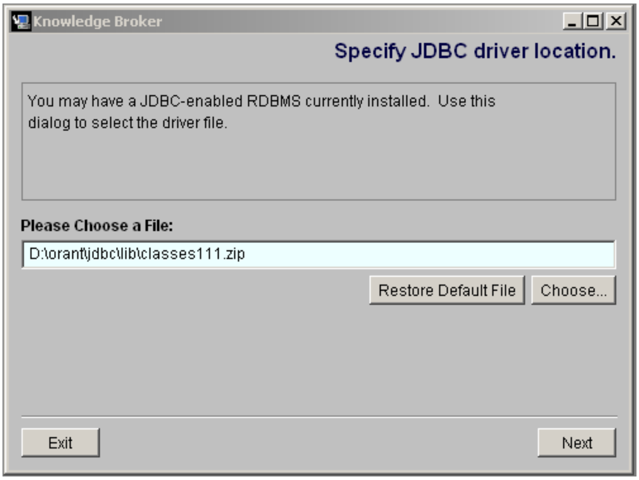
File name: classes111.zip

Files of type: All Files (*.*)

The name of the `.jar` or `.zip` file required depends on which database you wish to access. Your network administrator or DBA (Database Administrator) can help you locate suitable files that contain JDBC drivers.

11 Click Open to select the JDBC driver file. The Specify JDBC Driver Location dialog displays again, this time with the file location entered. Click Next to continue.

Figure 3-63. Specify JDBC Driver Location2 - Standalone Installation



Specify JDBC driver location.

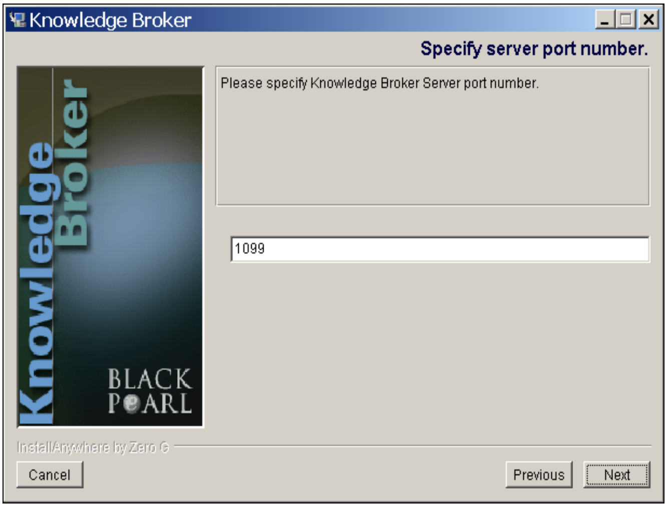
You may have a JDBC-enabled RDBMS currently installed. Use this dialog to select the driver file.

Please Choose a File:

D:\orant\jdbc\lib\classes111.zip

The Specify Server Port Number dialog displays..

Figure 3-64. Specify Server Port Number - Standalone Installation



Specify server port number.

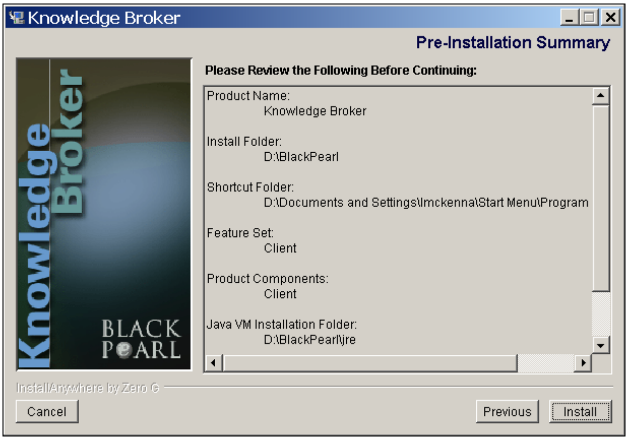
Please specify Knowledge Broker Server port number.

1099

i Specify the Server Port Number.

i Click Next. Knowledge Broker begins the installation of the Standalone version. The Pre-Installation Summary displays.

Figure 3-65. Server Name Completed - Standalone Installation



Knowledge Broker

Pre-Installation Summary

Please Review the Following Before Continuing:

Product Name:
Knowledge Broker

Install Folder:
D:\BlackPearl

Shortcut Folder:
D:\Documents and Settings\lmckenna\Start Menu\Program

Feature Set:
Client

Product Components:
Client

Java VM Installation Folder:
D:\BlackPearl\jre

12 Complete one of the following steps:

a If you do not accept the Pre-Installation Summary, click Previous and correct the necessary information.

b If you accept the Pre-Installation Summary, click Install. Knowledge Broker Standalone version is installed. Click Done.

You have now successfully installed the Standalone version of Knowledge Broker.

Configuring ODBC datasource Drivers

Because it is possible to use Knowledge Broker with a variety of datasource drivers, you must identify the driver before you can use Knowledge Broker. The Microsoft ODBC datasource Administrator provides a convenient single location that aggregates many datasource drivers for easy access. To create and specify system-wide ODBC drivers, follow these steps:

1 Select **Start > Settings > Control Panel > ODBC datasources** (Windows) or **Start > Settings > Control Panel > Administrative Tools > datasources** (**ODBC**) (Windows 2000). The ODBC datasource Administrator displays, listing all currently configured datasource drivers.

2 Click System DSN. The ODBC datasource Administrator/System DSN dialog displays.

3. Click Add. The Create New datasource dialog displays.
4. Select the appropriate datasource driver.
5. Click Finish. The ODBC datasource Administrator dialog re-displays.
6. Click OK. The ODBC Setup dialog displays.
7. Complete each of the Setup fields, including the path to the datasource.

Contact your Database Administrator or read the database documentation for the exact information to enter in the ODBC Setup dialog.

8. Click OK. The System DSN dialog re-displays.
9. Click OK again. The Control Panel re-displays.
10. Exit the Control Panel.

If you change datasource types, you will need to modify the ODBC Setup dialog information. Select **Start > Settings > Control Panel > ODBC datasources**. Click System DSN. Select the driver to be modified. Click Modify. Make the appropriate changes.

Configuring JDBC datasource Driver Locations

Some database vendors provide alternative database access methods in addition to the Microsoft ODBC mechanism. Commonly, they provide Java drivers that use the JDBC mechanism. These drivers are usually installed during a database client install process and frequently reside in `.jar` (a common Java file format) or `.zip` compressed files. Oracle, for example, provides a JDBC driver within an installed file known as `classes111.zip`.

You can select a suitable file during installation of Knowledge Broker (see Step 10 of *Installing Knowledge Broker Standalone on page 62*). Alternatively, you can specify or update JDBC driver locations after installation:

1 Navigate to the directory where you have installed Knowledge Broker.

2 In the `bin` sub-directory, edit the `server.bat` file using a text editor.

3 This batch command indicates the location of the JDBC driver file:

```
if "%JDBC_Drivers%"    == "" set
JDBC_Drivers=D:\orant\jdbc\lib\classes111.zip
```

4 Change this line to reflect the directory path and name where your `.jar` file resides. For example:

```
if "%JDBC_Drivers%"    == "" set
JDBC_Drivers=C:\MyNewRDBMS\lib\jdbc_access.jar
```

5 Save your changes to `server.bat`. When restarted, Knowledge Broker will search for a JDBC driver in the new location.

You can specify multiple file locations where Knowledge Broker will search for suitable JDBC drivers. Use a semicolon character (`;`) to separate the file locations, similar to the `CLASSPATH` format in Java. For example:

```
if "%JDBC_Drivers%"    == "" set
JDBC_Drivers=C:\OracleClient\classes\Oracle.jar;
D:\BlackPearl\lib\DB2.jar;D:\BlackPearl\lib\Sybase.jar
```

Uninstalling in Windows

1 Close all applications.

2 Complete one of the following options:

   a Select **Start > Programs > Black Pearl > Uninstall Knowledge Broker**. Knowledge Broker's uninstall screen displays, followed by a warning to exit all applications. Installed files are removed from your machine.

   b Click **Start > Settings > Control Panel > Add/Remove Programs**.

      i Select Knowledge Broker.

      ii Click Uninstall. Knowledge Broker's uninstall screen displays, followed by a warning to exit all applications. Installed files are removed from your machine.

3 Click OK when the Uninstall Complete dialog displays.

You may need to manually remove some Knowledge Broker files. The Uninstall Complete dialog will indicate if any Knowledge Broker files remain on your machine.

Uninstalling in Solaris

1 Close all Knowledge Broker applications (including any RMIregistry processes).

2 Change directory to Knowledge Broker's installation directory.

   a Run **Uninstall_Knowledge_Broker**

```
sh ./Uninstall_Knowledge_Broker)
```

   Knowledge Broker's uninstall screen displays, followed by a warning to exit all applications. Installed files are removed from your machine.

      i Select Knowledge Broker.

      ii Click Uninstall. Knowledge Broker's uninstall screen displays, followed by a warning to exit all applications. Installed files are removed from your machine.

3 Click OK when the Uninstall Complete dialog displays.

You may need to manually remove some Knowledge Broker files. The Uninstall Complete dialog will indicate if any Knowledge Broker files remain on your machine.

Chapter

4

Using Knowledge Broker

Although Knowledge Broker is designed to be an easy-to-use application, a quick introduction to basic procedures will increase your familiarity with the application.

This chapter provides you with information for basic operations including launching the application, opening an editor, closing an editor, and ending a session.

Starting Knowledge Broker

Deployment Engineers and Business Users start the Client version of Knowledge Broker. This Graphical User Interface (GUI) program enables users to log in to a Knowledge Broker Server to begin sessions and create and edit knowledge bases.

Knowledge Broker Client sessions require the services of an appropriately configured Knowledge Broker Server. Your system administrator can configure your Client to log in to a suitable Knowledge Broker Server.

Beginning a Knowledge Broker Server (Windows)

The Design-Time Server is used for development machines, whereas the Run-Time Server is deployed on production machines. Client sessions of Knowledge Broker ordinarily will log in to Design-Time Servers.

1. Start Knowledge Broker. Select the “Knowledge Broker Design-Time Server” shortcut in your Start Menu. Alternatively, run the `\bin\client.bat` file in your Knowledge Broker installation directory. The Black Pearl CommandHandler displays.
2. The CommandHandler displays the red status light. Knowledge Broker Servers install with the “autostart” option set, so a red light indicates Knowledge Broker is operational but can be stopped. A green status light indicates there was a problem during startup, or that the autostart option is disabled.

Beginning a Knowledge Broker Server (Solaris)

The Design-Time Server is used for development machines, whereas the Run-Time Server is deployed on production machines. Client sessions of Knowledge Broker ordinarily will log in to Design-Time Servers.

1. Start Knowledge Broker. Select the “Knowledge Broker Design-Time Server” shortcut in your Start Menu. Alternatively, run the `blackpearl/bin/server.ksh` file in your Knowledge Broker installation directory. The Black Pearl CommandHandler displays.
2. The CommandHandler displays the green status light. Knowledge Broker Servers install with the “autostart” option set, so a red light indicates Knowledge Broker is operational but can be stopped. A green status light indicates there was a problem during startup, or that the autostart option is disabled.

Starting a Knowledge Broker Server

Select the **CommandHandler > Start** menu option (or click the green light)

Stopping a Knowledge Broker Server

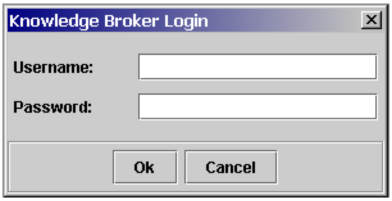
Select the **CommandHandler > Stop** menu option (or click the red light)

Beginning a Knowledge Broker Session (Windows)

To begin a new Knowledge Broker Client session:

1 Start Knowledge Broker. Select the "Knowledge Broker Client" shortcut in your Start Menu. Alternatively, run the `\bin\client.bat` file in your Knowledge Broker installation directory. Knowledge Broker's Login dialog displays.

Figure 4-1. Knowledge Broker Login



Knowledge Broker Login

Username:

Password:

2 Type a suitable Username and Password.

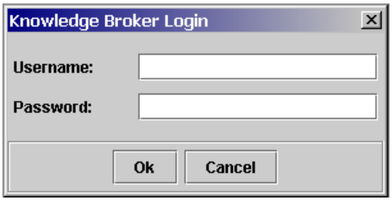
3 Click Ok to open the session.

Beginning a Knowledge Broker Session (Solaris)

To begin a new Knowledge Broker Client session:

1. Run the `/bin/client.ksh` file in your Knowledge Broker installation directory. Knowledge Broker's Login dialog displays:

Figure 4-2. Knowledge Broker Login

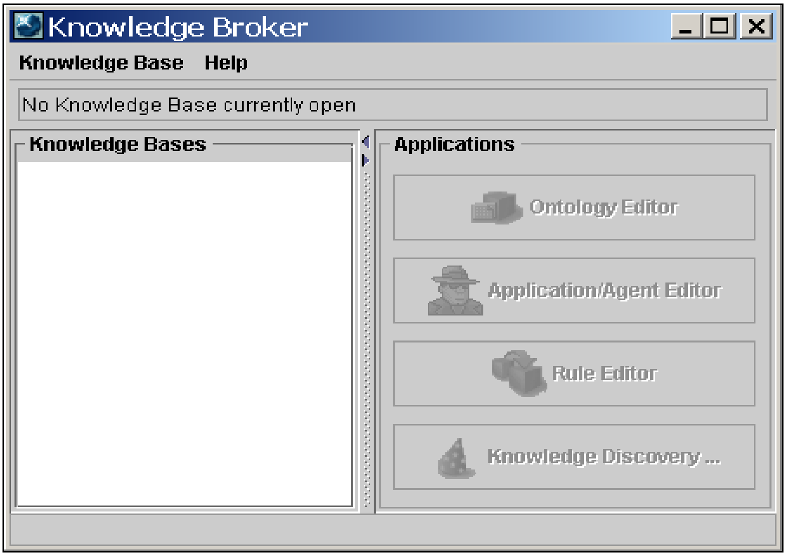


2. Type a suitable Username and Password.
3. Click Ok to open the session.

Using the Launcher

Knowledge Broker's initial display window, the Launcher, enables you to access all application functionality.

Figure 4-3. Knowledge Broker Launcher



The Launcher remains active throughout a *session*, which begins when you start Knowledge Broker and ends when you exit the Launcher and the associated command-line status windows. During a session you may add, modify, delete, or save ontology items, agents and applications, rules and rulesets, or datasets and models.

Opening a Task Editor or Center

The Launcher includes on-screen button options enabling you to access various Task selections, including the Agent/Application Editor and the Ontology Editor. To access a Task, you may:

- Click the appropriate button option. For example, to access the Ontology Editor, click the Ontology Editor button.

Re-sizing a Task Editor or Center

You can control the size of any Task Editor by using your windowing system's standard Size Control buttons.

Closing a Task Editor or Center

Only a single instance of a specific Task Editor or Center may be active at any one time, although multiple Tasks may have one or more windows open at the same time.

To close Task Editor or Center, you may either:

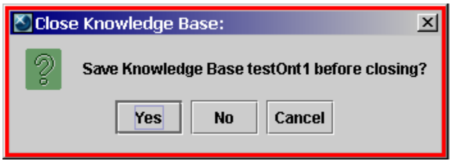
- Select **File > Close Editor** in the open Task Editor or Center window.
- Click the Close button in the upper right-hand corner of the open Task Editor or Center.

Ending a Session

A *session* starts when you launch the Knowledge Broker Client and log in and ends when you exit the application.

1 Select **Knowledge Base > Exit**. Knowledge Broker's Close dialog displays, asking if you want to save any changes that you made during the session. (If you saved your changes prior to selecting **Knowledge Base > Exit**, the dialog does not display.)

Figure 4-4. Close Knowledge Base - Save Knowledge Base?



Save Knowledge Base testOnt1 before closing?

2 Complete one of the following options:

a Click Yes to save the changes to the knowledge base.

b Click No to end the session without saving the changes.

c Click Cancel to continue the current session.

Part

III

Procedures

Using Knowledge Broker requires that you define an ontology and rules. Optimal use of the application includes defining a predictive model.

Part III: Procedures provides step-by-step instructions for using Knowledge Broker to complete these tasks.

Chapter

5

Defining a Knowledge Base

This chapter provides step-by-step instructions for defining a knowledge base that contains all your ontologies, business rules, agents, and applications relevant to your business requirements.

Managing Knowledge Bases

To begin working with Knowledge Broker you must first create a *knowledge base*. A knowledge base is the collection of business concepts, rules, agents, applications, and predictive models related to a particular industry or business.

Knowledge Base Concepts

Knowledge Broker enables you to create an explicit knowledge base that includes information about ontologies, business rules, agents, and applications relevant to your business requirements. It then uses that knowledge base to discern patterns in your raw data, understand the implications of those patterns, make real-time recommendations, and dynamically and iteratively refine and deepen the knowledge base.

Editing Knowledge Bases

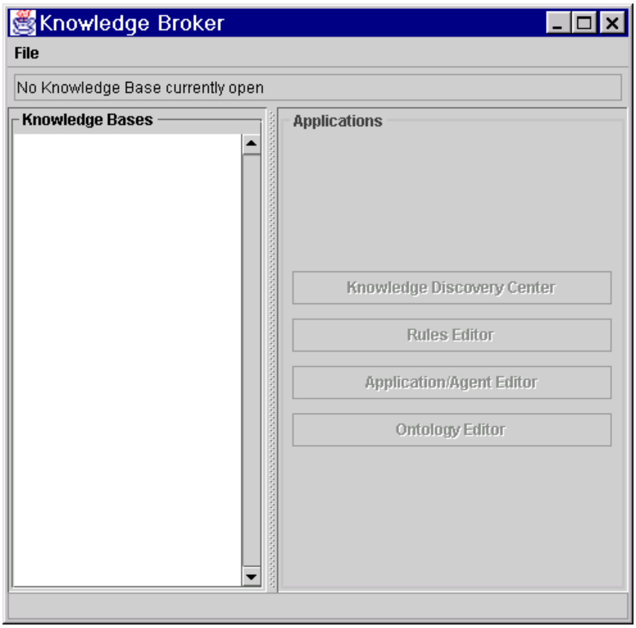
Knowledge bases can be created, edited, modified and deleted just like any standard object. Deleting a knowledge base deletes all ontology, agent, applications and related data. You should exercise care when deleting knowledge bases because such deletions are final.

Editing a Knowledge Base

Knowledge Broker's Launcher enables you to access all application functionality. The Launcher is a graphical display you see when you execute either the Standalone or the Client installation of Knowledge Broker. If you edit Knowledge Bases on a machine where Knowledge Broker Server is installed, you use a local Client.

Knowledge Broker's Launcher displays two panels. The right-hand-panel displays the application functionality. The left-panel is the Knowledge Base Editor. The Knowledge Base Editor enables you to create, open, close, and delete a knowledge base.

Figure 5-1. Knowledge Broker Launcher

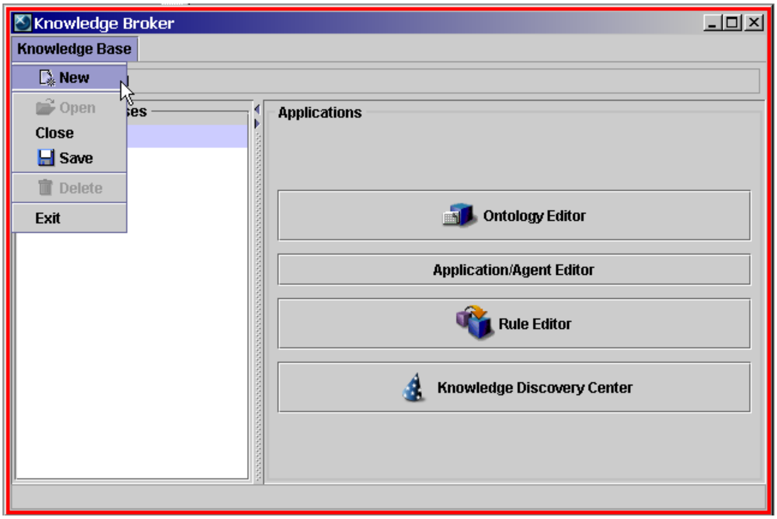


Creating a New Knowledge Base

You can create a new knowledge base using the Knowledge Base Editor.

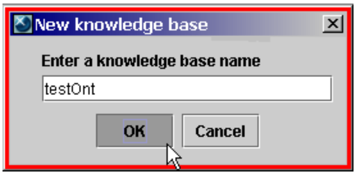
1 Go to **Knowledge Base > New.**

Figure 5-2. Creating a New Knowledge Base



The New Knowledge Base window displays.

Figure 5-3. New Knowledge Base Name Window



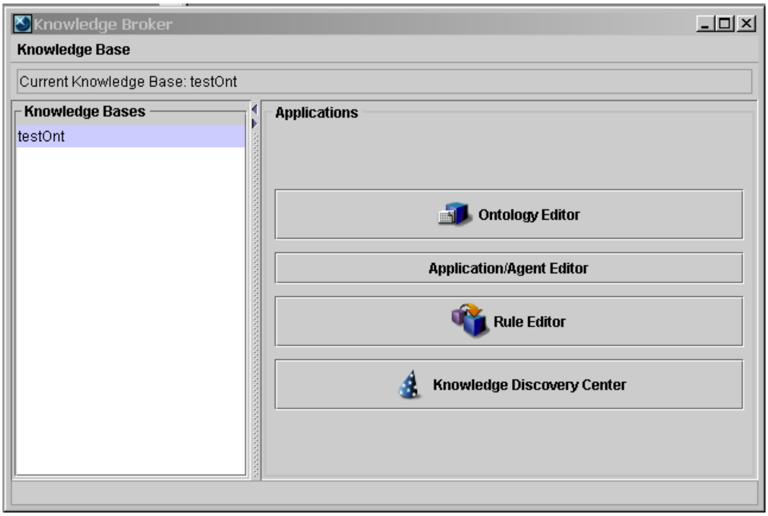
Enter a knowledge base name

testOnt

2 Enter a new knowledge base name and click OK.

You have now created a new Knowledge Base. The Knowledge Base Editor displays the newly created knowledge base.

Figure 5-4. New Knowledge Base Displays



Current Knowledge Base: testOnt

testOnt

Before you close the Knowledge Base Editor, you must save the new knowledge base.

1 Select the knowledge base.

2 Go to **Knowledge Base > Save**.

3 Close the Knowledge Base Editor.

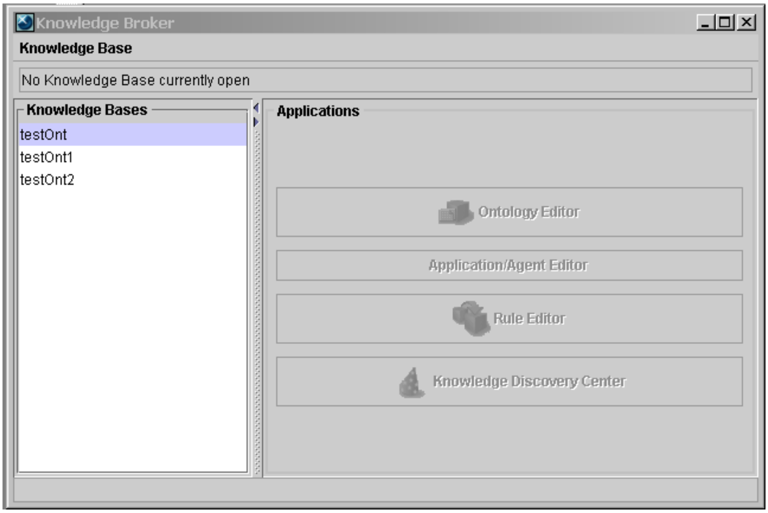
You have saved the knowledge base. The next time you open the Knowledge Base Editor the saved knowledge base appears.

To create or modify a different new Knowledge Base, you must close the currently open Knowledge Base.

Opening an Existing Knowledge Base

1 Existing Knowledge Bases display within the Knowledge Base Editor.

Figure 5-5. Existing Knowledge Bases



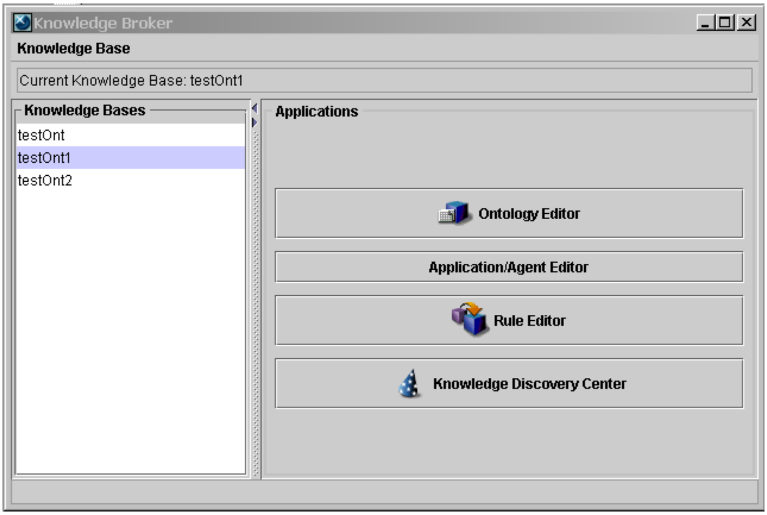
No Knowledge Base currently open

testOnt
testOnt1
testOnt2

2 Select the desired Knowledge Base.

3 Go to **Knowledge Base > Open** (or double-click). The Knowledge Base Editor display updates to reflect the opened Knowledge Base.

Figure 5-6. Existing Knowledge Base Opened



Current Knowledge Base: testOnt1

- testOnt
- testOnt1
- testOnt2

Only one knowledge base can be open at any time.

During loading, Knowledge Broker performs validation using a SAX parser and notifies user concerning discovered errors.

Saving a Knowledge Base

1 In the Knowledge Base Editor, select the desired knowledge base.

2 Go to **Knowledge Base > Save**. The knowledge base is saved.

Closing a Knowledge Base

1 In the Knowledge Base Editor, select the desired knowledge base.

2 Go to **Knowledge Base > Close**. The knowledge base is closed.

Deleting a Knowledge Base

1. First, close all Knowledge Bases.
2. Select which Knowledge Base to delete.
3. Go to **Knowledge Base > Delete**. The Knowledge Base is deleted.

Sometimes because of system locking, the delete may not work. Ask an administrator to delete the unwanted knowledgebase.

Closing the Knowledge Base Editor

To close the Knowledge Base Editor go to **Knowledge Base > Exit**.

Editing the Knowledge Base (Multi-User)

When multiple Clients access a Server installation of Knowledge Broker, only one user can have full read-write client access to the knowledge base at any one time If multiple users invoke Knowledge Broker client sessions and attempt to edit the same knowledge base (that is, a specific ontology and associated set of rules), the first user to initiate a session “wins” complete read-write access while subsequent users gain access in read-only mode.

Knowledge Broker will alert you during launch if read-only mode is being enforced.

While the initial user’s session remains active, the knowledge base remains locked for read-only access. When the initial user quits their session, the knowledge base becomes available once again for read-write access. The next user to launch a Knowledge Broker session gains full read-write access.

This behavior continues for as long as the server runs. When the read-write user quits their session, the next user to launch gains read-write access while all other users gain access in read-only mode.

Currently, there is no notification of the “freeing” of the knowledge base for read-write access, and the read-only status of current user sessions will not change mode for the duration of an individual session. Therefore, periodically quitting and restarting Knowledge Broker is the only way for a read-only mode user to obtain read-write access.

While operating in read-only mode, users can still type in the GUI. They can edit ontology items and agents/applications. However, there is no save functionality and all related menu options are disabled; these display ghosted or greyed out.

There is no timeout implemented for user sessions. By neglecting to release their lock, it is possible for the read-write access user to lock the knowledge base into read-only mode for an abnormal amount of time. This could occur if the read-write access user’s Knowledge Broker client session or system crashes unexpectedly, or if they go on vacation and leave their client session active.

In this case, the System Administrator must manually quit and restart Knowledge Broker Server. This removes the lock and permits read-write access to resume.

Chapter

6

Defining an Ontology

This chapter provides step-by-step instructions for defining an ontology that links to your datasources and is relevant to your business requirements. Knowledge Broker's ontology consists of two main concepts:

- Event Business Concepts
- Persistent Business Concepts

Business Users deal only with these Business Concepts. Deployment Engineers address the complex task of also establishing Mappings and Connections. Advanced Knowledge Broker Users can create and edit Interactions.

Introducing The Ontology

An *Ontology* is a collection of concepts and contexts used to provide a common vocabulary for defining rules, querying disparate datasources, analyzing information patters, recognizing implications, and making actionable recommendations. The *Ontology Editor* consists of three tabs: Business Concepts, Mappings, and Connections. The Mappings and Connections tabs are used by the Deployment Engineer. The Business User uses the Business Concept tab.

The Business User can do the following tasks using the Ontology Editor:

- Create a New Root Business Concept (RBC)
- Create a Derived Root Business Concept (DRBC)
- Create a Derived Business Concept (DBC)
- Add a Calculated Property

Knowledge Broker represents Business Concepts as a hierarchy. At the "top" are Root Business Concepts (RBCs). These have a direct connection to external datasources. You can structure your data by selecting a pre-existing schema or discovering a schema using schema auto-generation. Once the data is accessed by the Root Business Concept it's available to Derived Root Business Concepts and Derived Business Concepts.

Business Users can create secondary Business Concepts using two methods.

1. Create Derived Root Business Concepts (DRBCs) using a subset of the Agent Control Language to extract and process Element data from Root Business Concepts, other Derived Root Business Concepts, or Derived Business Concepts. An advantage of using Derived Root Business Concepts is that they have access to the data that exists in the Root Business Concept.
2. Create Derived Business Concepts (DBCs) using filters to transform Element data from Root Business Concepts, Derived Root Business Concepts, or other Derived Business Concepts. The advantage of DBCs over DRBCs is that their creation process is quicker and more straightforward. A Filter differentiates a Business Concept from its parent Business Concept by specifying selection parameters using a logical expression in this format:

   ```
   <property_value> <operator> <comparison_value>
   ```

   Knowledge Broker derives the subset of DBCs from the larger set of RBCs by selecting only those instances of RBCs that satisfy the logical expression. For example, you could begin with a RBC named `Equity` and derive a DBC named `Volatile Stocks` by selecting only those stocks with a parameter such as `Beta > 5`.

Calculated Properties are the remaining ontology objects. Using these, Business Users can transform Element data using complex mathematical formulas or statistical or case-based reasoning models. The transformed data can be incorporated within DRBCs or DBCs for further processing or output.

Understanding Business Concepts

A *Business Concept* is a vocabulary term that is relevant to your industry or company. You define Business Concepts in the Ontology Editor. A concept is an abstract image about the who, what, when, and where of data and data relationships. Within Knowledge Broker, concepts are the vocabulary terms used for defining rules, querying disparate datasources, analyzing information patters, recognizing implications, and making actionable recommendations. A Connection maps directly to one or more datasource objects.

The ability to define Business Concepts within Knowledge Broker allows you to impose a familiar logical structure on disparate and sometimes haphazardly organized data. Through using a familiar, hierarchical, English-like structure, you enhance your ability to think about your business processes and domain knowledge. This is because in addition to using your traditional analytical abilities, you are also leveraging your brain's powerful language processing centers. Using language to reason about concepts introduces a degree of flexibility, extensibility, and maintainability that is simply impossible using mere analytical processing. You have made the leap from processing data to thinking about knowledge. Combining Knowledge Broker's flexible, English-like Rules, Agents, and Applications with Business Concepts produces actionable facts and recommendations.

Event and Persistent Business Concepts

The Business Concepts arranged in a hierarchy within the ontology are also differentiated according to their type of datasource.

Event Business Concepts (EBCs) attach to message streams, wires, or news dispatches. Their contents change rapidly and continually; think of a faucet continually "on" with water or data streaming out. Using EBCs, you can listen for specific messages or types of messages, or push Knowledge Broker recommendations back "out" to wires.

Persistent Business Concepts (PBCs) attach to databases, files, web pages, or other static data repositories. You expect their contents to change infrequently; think of a silo or warehouse where you must request access to put something in or take something out. Using PBCs, you can query datasources and retrieve specific information, or publish Knowledge Broker recommendations "out" to structured storage media.

Flat and Structured Business Concepts

Knowledge Broker uses a technology known as XML Schema (XSD) to model business concepts. Advanced XSD supports complex type elements. Whereas XSD simple type elements can contain only text, XSD complex type elements can contain other elements and attributes. Earlier versions of Knowledge Broker (supporting only XSD simple type elements) provided a hierarchy of “flat” business concepts. Although their Element slots could contain any of the text types included in the XML Schema definition (boolean, string, date, user-defined, and so on), this tabular schema was suited more for normalized data and lacked flexibility when representing heterogeneous, often hierarchical data sources.

During business concept creation, you can select whether or not to create simple (flat) or complex (nested) business concepts.

For more details about XSD and related technologies, see *Writing XQueries on page 499*.

Using Root Business Concepts

Knowledge Broker supports XML Schema (XSD) complex type elements. Whereas XSD simple type elements can contain only text, XSD complex type elements can contain other elements and attributes. Earlier versions of Knowledge Broker (supporting only XSD simple type elements) provided a hierarchy of “flat” business concepts. Although their Element slots could contain any of the text types included in the XML Schema definition (boolean, string, date, user-defined, and so on), this tabular schema was suited more for normalized data and lacked flexibility when representing heterogeneous, often hierarchical data sources.

You can create three types of Root Business Concepts, distinguished by their support for XSD:

a Simple Elements only

b Complex Elements

c External Schema

An RBC with Simple Elements only is similar to a database or table. It does not support complex hierarchical data.

An RBC with Complex Elements enables you to nest Business Concepts within each other. It can model complex hierarchical data, but can be complex.

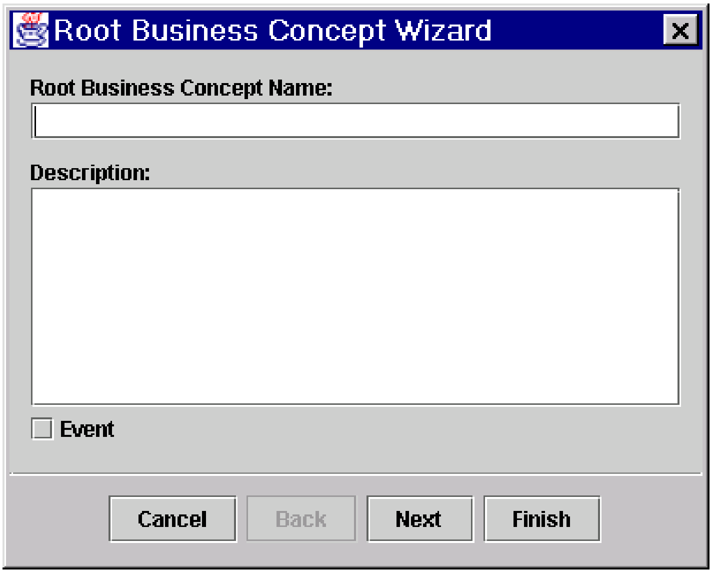
An RBC with External Schema support enables you to use a third party XML schema editor to create or modify business-specific schema and incorporate it into Knowledge Broker.

Creating a New RBC with Simple Elements Only

1 Start the Ontology Editor, if it is not currently open.

2 Go to **Concept > New > New Root Business Concept**. The Root Business Concept Wizard displays.

Figure 6-1. New Root Business Concept Wizard- With Simple Elements Only



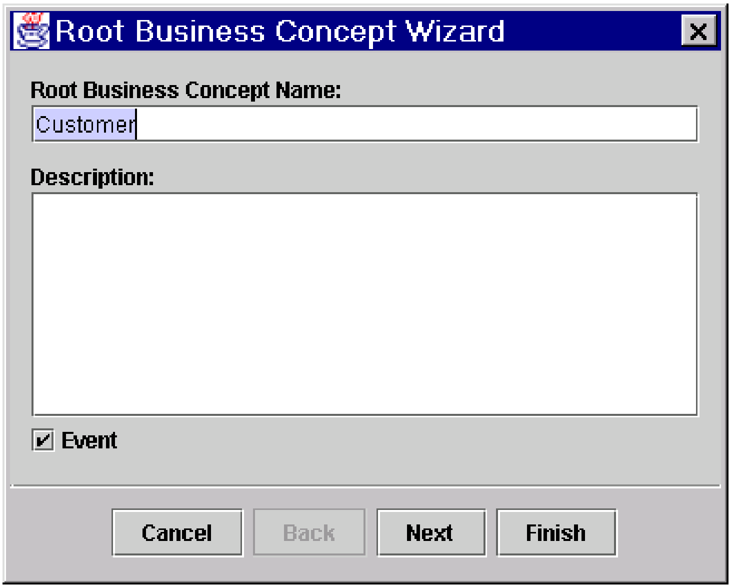
3 Type a name for the Root Business Concept in the Root Business Concept Name field.

Knowledge Broker's uses an underlying data query mechanism known as XQuery to traverse the Ontology. Because of this, there are certain XML Schema-reserved words that should not be used for Ontology object names. Using these words can produce errors and abnormal program termination. You can find a list of these reserved words in *Appendix C, XQuery Reserved Keywords on page 498*.

4 Type a description for the Root Business Concept in the Description field.

5 Select Event if the Business Concept is an Event Business Concept.

Figure 6-2. New Root Business Concept Wizard- With Simple Elements Only - Naming



Root Business Concept Wizard

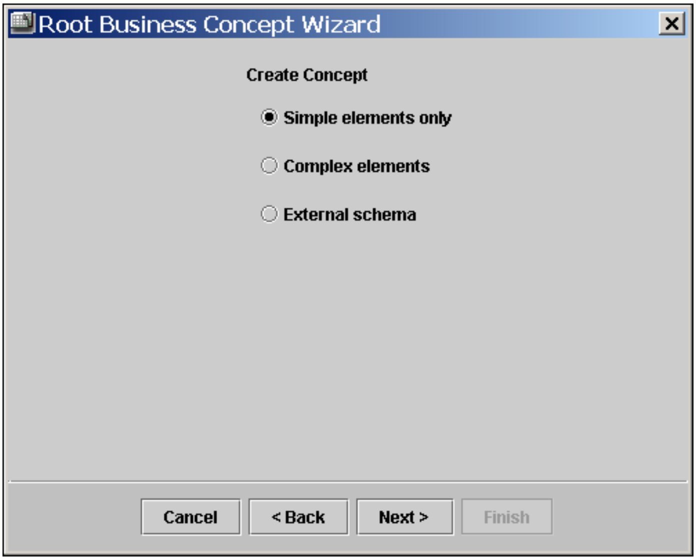
Root Business Concept Name: Customer

Description:

☑ Event

6 Click Next. The Root Business Concept Wizard displays.

Figure 6-3. New Root Business Concept Wizard- With Simple Elements Only - Type Selector



Create Concept

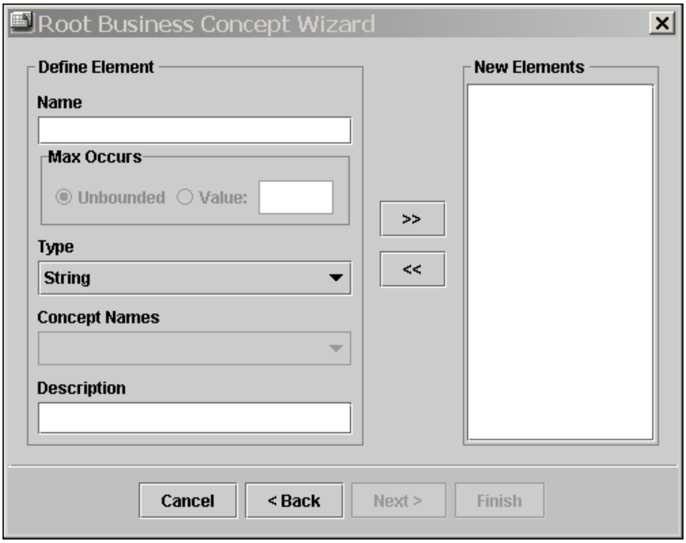
● Simple elements only

○ Complex elements

○ External schema

7 Select Simple Elements Only and select the Next button to continue. The Root Business Concept Wizard displays.

Figure 6-4. New Root Business Concept Wizard- With Simple Elements Only - Define Elements



8 Type a name for the Element in the Add Element Name field.

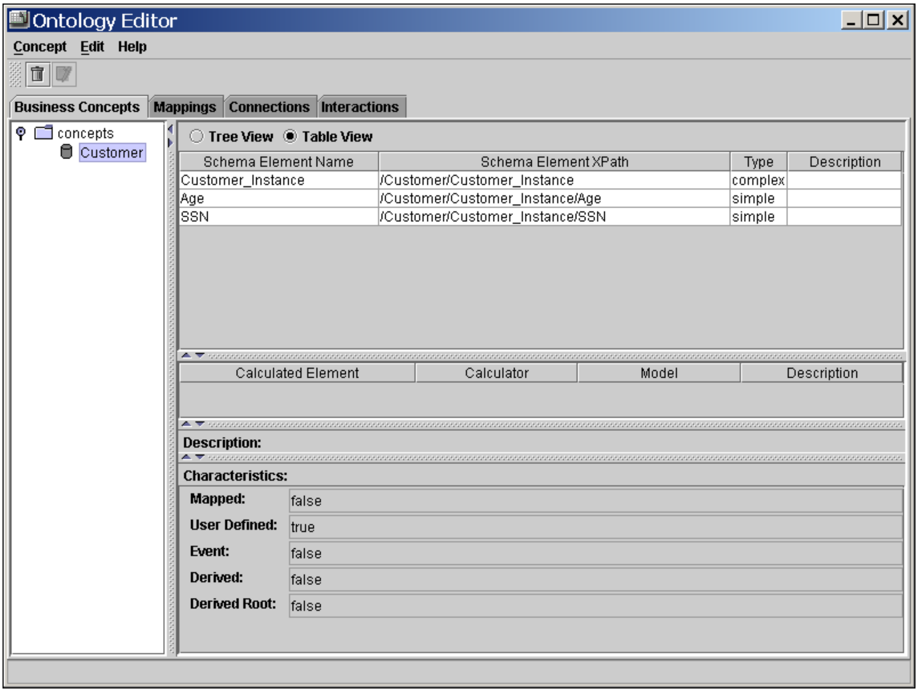
9 Type a description of the Element in the Element Description field.

10 Select Element Type using the pull-down menu String as element type tells the Root Business Concept Wizard the element is text. Decimal as an element type tells the Root Business Concept Wizard the element is numbers. (For example, if you enter an address as an element name you select String as type. If you enter age as an element type you select Decimal as type.) To add the element to the Root Business Concept click >> . The element displays in the right panel. (To remove the element from the Root Business Concept select the element and click << .) You can add additional elements to the RBC by using a similar procedure.

11 Click Finish. The newly created Root Business Concept with Simple Elements only displays in Table View in the Ontology Editor. The second type of view, Tree View, is

used to view Structured Business Concepts. See *Creating a New RBC with Complex Elements on page 104*.

Figure 6-5. Ontology Editor - Table View



○ Tree View ● Table View

Schema Element Name	Schema Element XPath	Type	Description
Customer_Instance	/Customer/Customer_Instance	complex	
Age	/Customer/Customer_Instance/Age	simple	
SSN	/Customer/Customer_Instance/SSN	simple	

Calculated Element	Calculator	Model	Description

Description:

Characteristics:

Mapped: false

User Defined: true

Event: false

Derived: false

Derived Root: false

You have just created a new Root Business Concept with Simple Elements only.

Creating a New RBC with Complex Elements

Knowledge Broker now features preliminary support for Structured Business Concepts (SBCs). SBCs enable business concepts to be “nested” within other business concepts, creating complex hierarchies. Because this support is preliminary, SBCs exhibit certain XSD-related restrictions:

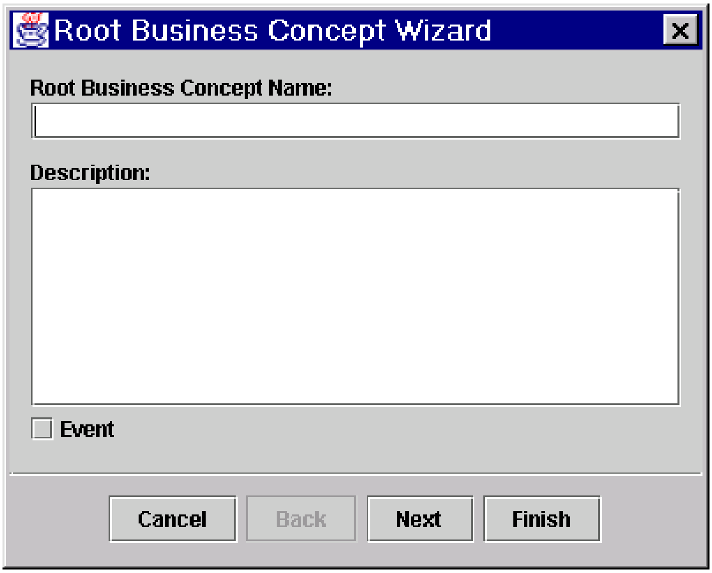
a When embedding a concept within a complex concept, the defintion of the embedded concept must be based on a named complexType and not an anonymous complexType.

To create an RBC with Complex Elements (an SBC):

1 Start the Ontology Editor, if it is not currently open.

2 Go to **Concept > New > New Root Business Concept**. The Root Business Concept Wizard displays.

Figure 6-6. New Root Business Concept Wizard- With Complex Elements



Root Business Concept Wizard

Root Business Concept Name:

Description:

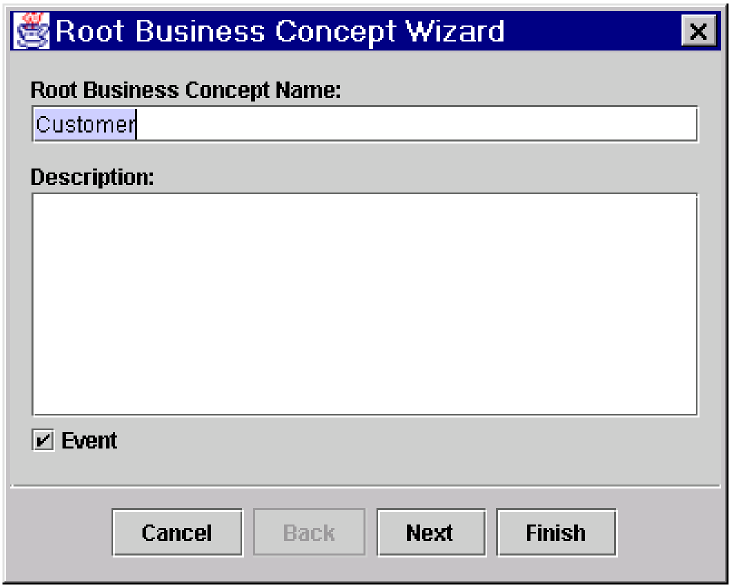
□ Event

3 Type a name for the Root Business Concept in the Root Business Concept Name field.

Knowledge Broker's uses an underlying data query mechanism known as XQuery to traverse the Ontology. Because of this, there are certain XML Schema-reserved words that should not be used for Ontology object names. Using these words can produce errors and abnormal program termination. You can find a list of these reserved words in *Appendix C, XQuery Reserved Keywords on page 498*.

4 Type a description for the Root Business Concept in the Description field. Event if the Business Concept is an Event Business Concept.

Figure 6-7. New Root Business Concept Wizard- With Complex Elements - Naming



Root Business Concept Wizard

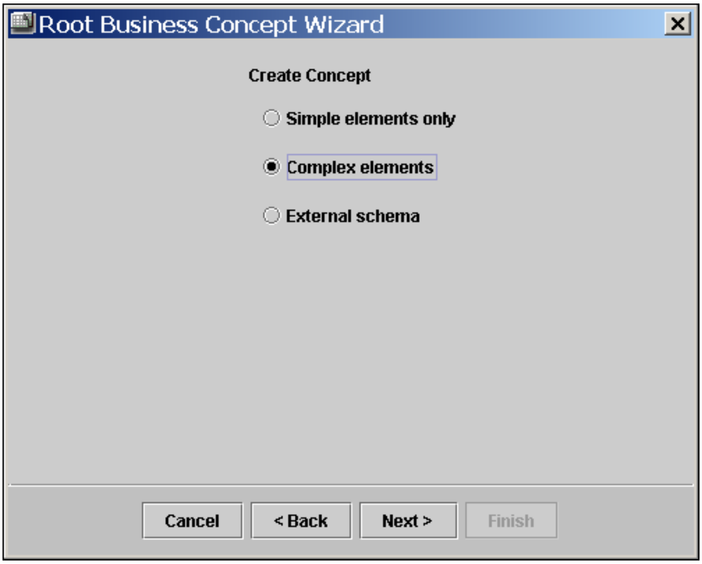
Root Business Concept Name: Customer

Description:

☑ Event

5 Click Next. The Root Business Concept Wizard displays.

Figure 6-8. New Root Business Concept Wizard- With Complex Elements - Type Selector



Root Business Concept Wizard

Create Concept

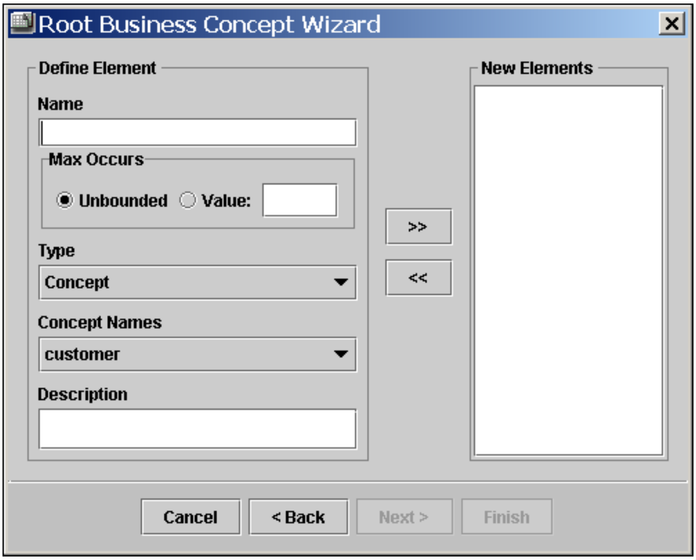
○ Simple elements only

● Complex elements

○ External schema

6 For SBCs, you must select Complex Elements. Click the Next button to continue. The Root Business Concept Wizard displays.

Figure 6-9. New Root Business Concept Wizard- With Complex Elements - Define Elements



Root Business Concept Wizard

Define Element

Name

Max Occurs

● Unbounded ○ Value:

Type

Concept

Concept Names

customer

Description

New Elements

7 Type a name for the Element in the Name field.

8 You must select a Max Occurs (Maximum Occurrences). Complete one of the following:

a Select Unbounded if you want all occurrences of the element to show.

b Select Value and enter a number in the Value field. This shows only the specified number of occurrences for this element.

9 You must select Type using the pull-down menu. Complete one of the following:

a Select String as type if the element is alphanumeric. For example, if you define “Address” as an element name you select String because address is specified by a combination of letters and numbers.

b Select Decimal as type if the element is composed only of numbers. This tells the RBC Wizard the element is numeric. For example, If you define “Age” as an element name you select Decimal as type because age is specified by numbers only.

c Concept as an element type tells the Root Business Concept Wizard the element is a previously defined Flat Business Concept.

10 Type a description for the element in the Description field.

11. To add an element to the Root Business Concept click >> . The element displays in the right panel. (To remove the element from the Root Business Concept select the element and click << .) You can add additional elements to the RBC by using a similar procedure.
12. If you selected Concept from the Type pull-down menu the Concept Names pull-down menu is active. Select a previously defined Business Concept.
13. Type a description of the Element in the Element Description field.
14. Click Finish.

The Ontology Editor detects SBCs, iterates through their contents, and displays them using XPath notation:

Figure 6-10. Structured Business Concept

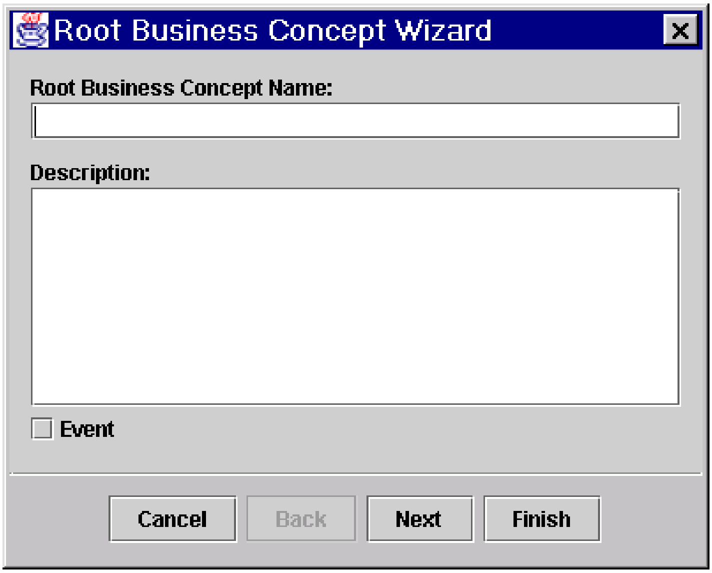
You have just created a New Root Business Complex with Complex Elements (or a Structured Business Concept).

Creating a New RBC with External Schema

1 Start the Ontology Editor, if it is not currently open.

2 Go to **Concept > New > New Root Business Concept**. The Root Business Concept Wizard displays.

Figure 6-11. New Root Business Concept Wizard- With External Schema



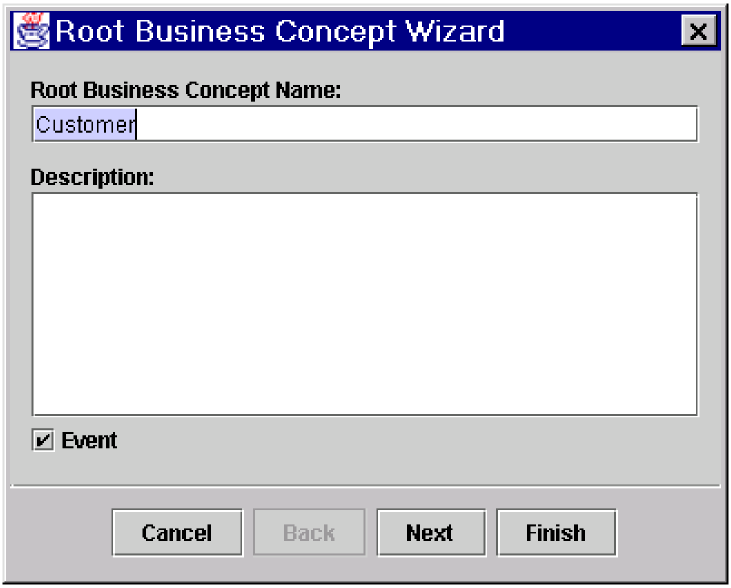
3 Type a name for the Root Business Concept in the Root Business Concept Name field.

Knowledge Broker's uses an underlying data query mechanism known as XQuery to traverse the Ontology. Because of this, there are certain XML Schema-reserved words that should not be used for Ontology object names. Using these words can produce errors and abnormal program termination. You can find a list of these reserved words in *Appendix C, XQuery Reserved Keywords on page 498*.

4 Type a description for the Root Business Concept in the Description field.

5 Select Event if the Business Concept is an Event Business Concept.

Figure 6-12. New Root Business Concept Wizard- With External Schema - Naming



Root Business Concept Wizard

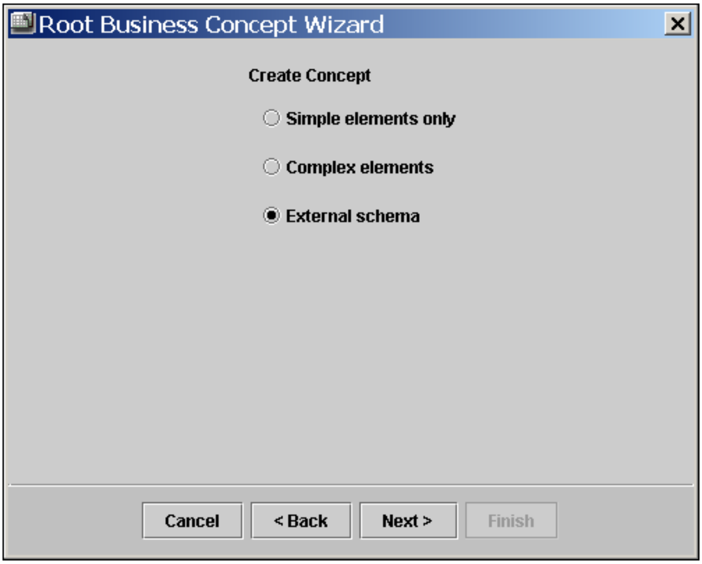
Root Business Concept Name: Customer

Description:

☑ Event

6 Click Next. The Root Business Concept Wizard displays.

Figure 6-13. New Root Business Concept Wizard- With External Schema - Type Selector



Create Concept

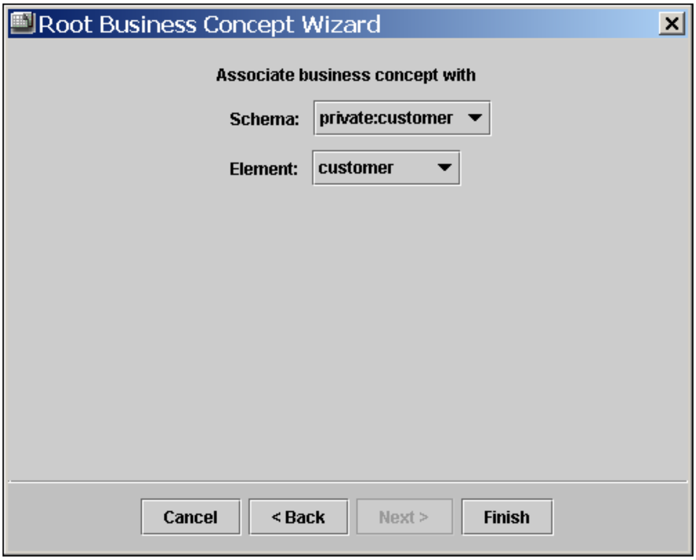
○ Simple elements only

○ Complex elements

● External schema

7 Select External Schema. Use this option if you have pre-existing schemas. Click the Next button to continue. The Root Business Concept Wizard displays.

Figure 6-14. New Root Business Concept Wizard- With External Schema - Type Selector



Associate business concept with

Schema: private:customer

Element: customer

8 Select a pre-existing Schema from the Schema pull-down menu.

9 Click Finish. The newly created Root Business Concept displays in the Ontology Editor.

Figure 6-15. Ontology Editor - Table View

You have just created a new Root Business Concept with External Schema.

Using Derived Root Business Concepts

RBCs are the mechanism for transferring data into Knowledge Broker. Derived Root Business Concepts (DRBCs) provide a mechanism for manipulating and transforming this data, as well as applying probabilistic models and complex analytics. DRBCs support a wide variety of data manipulation commands that can be sequenced together in a "flow" to model business domains and hunches.

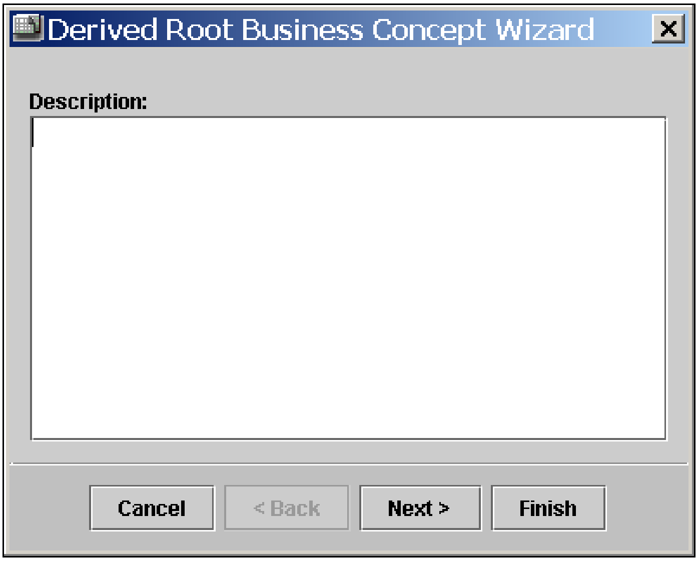
During DRBC creation and modification, you create subsets of RBC data. You can create DRBCs within other DRBCs and these inherit the "parent" specifications. You can "get" RBCs using language descriptors and so enlarge or shrink the subset of data contained within the DRBC.

Creating a New Derived Root Business Concept

1 Open the Ontology Editor, if it is not currently open.

2 Go to **Concept > New > New Derived Root Business Concept**. The Derived Root Business Concept Wizard displays.

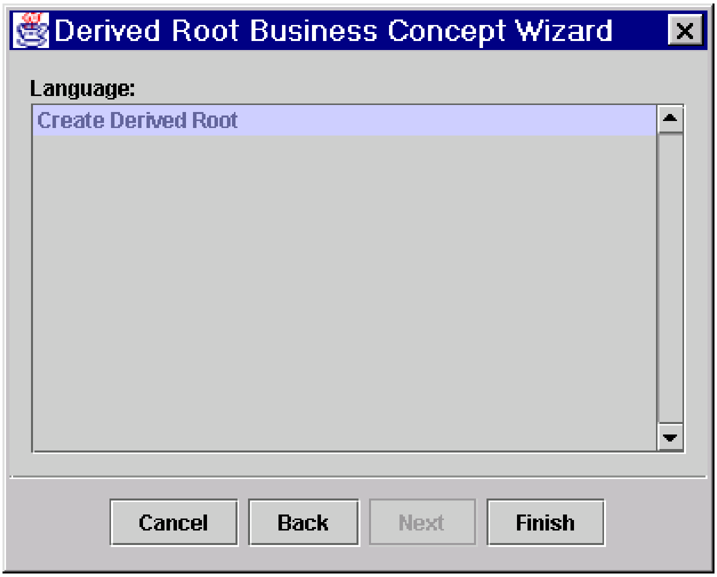
Figure 6-16. Derived Root Business Concept Wizard



3 Type a description for the new Derived Root Business Concept in the Description field.

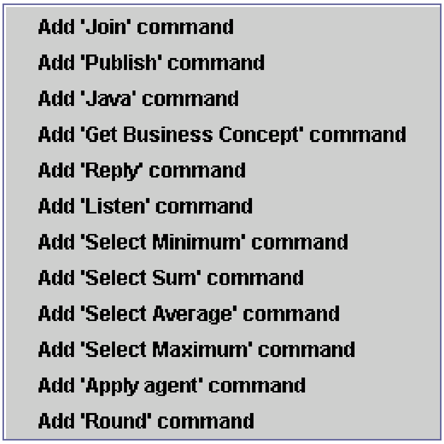
4 Click Next. The Derived Root Business Concept Wizard displays.

Figure 6-17. Derived Root Business Concept Wizard



5 Option-click on Create Derived Root. The Command Menu displays.

Figure 6-18. Command Menu



Add 'Join' command
Add 'Publish' command
Add 'Java' command
Add 'Get Business Concept' command
Add 'Reply' command
Add 'Listen' command
Add 'Select Minimum' command
Add 'Select Sum' command
Add 'Select Average' command
Add 'Select Maximum' command
Add 'Apply agent' command
Add 'Round' command

6 Select Commands to create flow for the new Derived Root Business Concept.

You must select commands in a logical command structure. if you do not, an error message with display asking you to check the language.

7 Select Create Business Concept Command to create new Derived Root Business Concept. Click Finish. The newly created Derived Root Business Concept displays in the Ontology Editor.

You have just created a new Derived Root Business Concept.

Using Derived Root Manipulation Language

Knowledge Broker selects how to populate a DRBC with data by reading sequentially a list of instructions encoded with each DRBC. These instructions are written in Derived Root Manipulation Language (DRML) and progressively narrow or widen the set of data suitable for inclusion, until there are no more instructions remaining.

There are twelve DRML commands:

1 Add Join

2 Add Publish

3 Add Java

4 Add Get Business Concept

5 Add Reply

6 Add Listen

7 Add Select Minimum

8 Add Select Sum

9 Add Select Average

10 Add Select Maximum

11 Add Apply Agent

12 Add Round

Add Join

A = JOIN <Rowset1>,<Rowset2> ON Rowset1,<Attribute>,Rowset2.<Attribute>

Add Publish

PUBLISH <Rowset list> TO <Media> USING <Template>

Add Java

This enables advanced users to insert Java programming language directions within a DRBC. For more details, see *Java Exits on page 491*.

Add Get Business Concept

GET <PBC> (FIRST <n rows>) (WITH <attribute> <relation> {text, number})

Add Reply

REPLY WITH <Rowset list> (SUING <XSLT>)

Add Listen

LISTEN FOR <EBC> DURING <range>

Add Select Minimum

X = MINIMUM OF <Rowset>.<Attribute> (GROUP BY <Attribute>) (WHERE <Attribute> <relation> {text or number})

Add Select Sum

X = SUM OF <Rowset>.<Attribute> (GROUP BY <Attribute>) (WHERE <Attribute> <relation> {text or number})

Add Select Average

X = AVERAGE OF <Rowset>.<Attribute> (GROUP BY <Attribute>) (WHERE <Attribute> <relation> {text or number})

Add Select Maximum

X = MAXIMUM OF <Rowset>.<Attribute> (GROUP BY <Attribute>) (WHERE <Attribute> <relation> {text or number})

Add Apply Agent

APPLY <Agent>

Add Round

Y = ROUND <variable> TO {n} PLACES

Using Derived Business Concepts

RBCs are the mechanism for transferring data into Knowledge Broker. Like DRBCs, Derived Business Concepts (DBCs) provide a mechanism for manipulating and transforming this data. The key difference is that DBCs use filters to create subsets of RBC or DRBC data. You can create DBCs within other DRBCs or DBCs and these inherit the "parent" filters. Thus, DBCs always lead to a "narrowing" of the data set as instance items that fail to satisfy the filter conditions are eliminated.

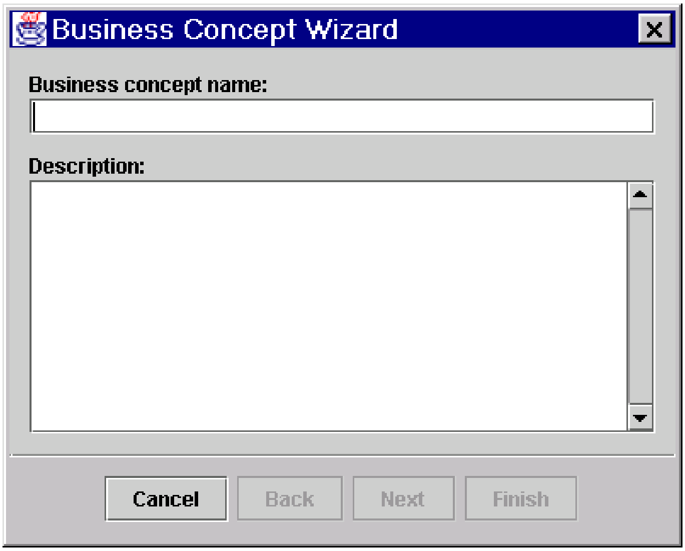
Creating a New Derived Business Concept

A *filter* differentiates a Business Concept from its Parent Concept by specifying the parameters that Knowledge Broker will use to evaluate, process, and sort data.

1 Open the Ontology Editor, if it is not currently open.

2 Select a Business Concept and go to **Concept > New > New Derived Concept** The Business Concept Wizard displays.

Figure 6-19. Business Concept Wizard



Business Concept Wizard

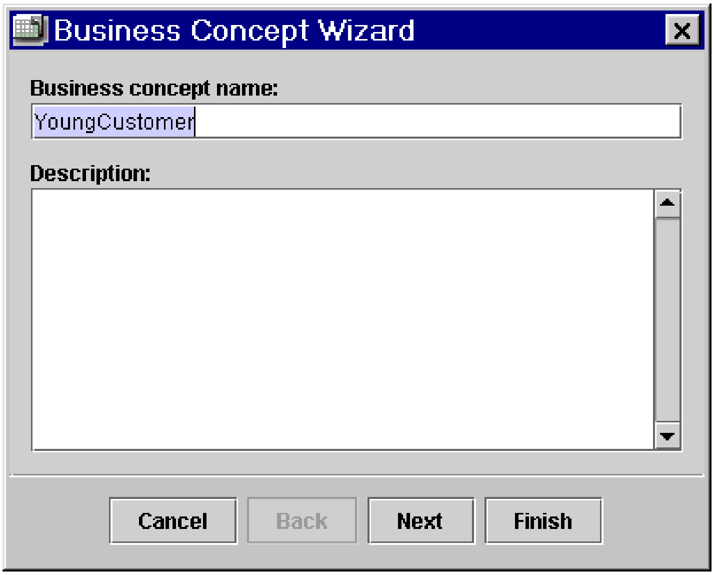
Business concept name:

Description:

3 Type a name for the Business Concept in the Business Concept Name field.

4 Type a description for the Business Concept in the Description field.

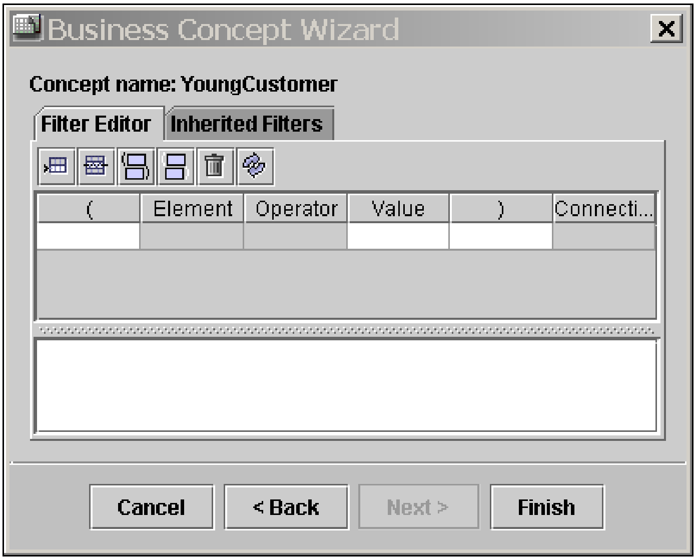
Figure 6-20. Business Concept Wizard



Business concept name: YoungCustomer

5 Click Next. The Business Concept Wizard displays.

Figure 6-21. Business Concept Wizard



Concept name: YoungCustomer

(	Element	Operator	Value	)	Connecti...

6 Select the Filter Editor tab.

For information about using Filters, see *Using Filters on page 123*.

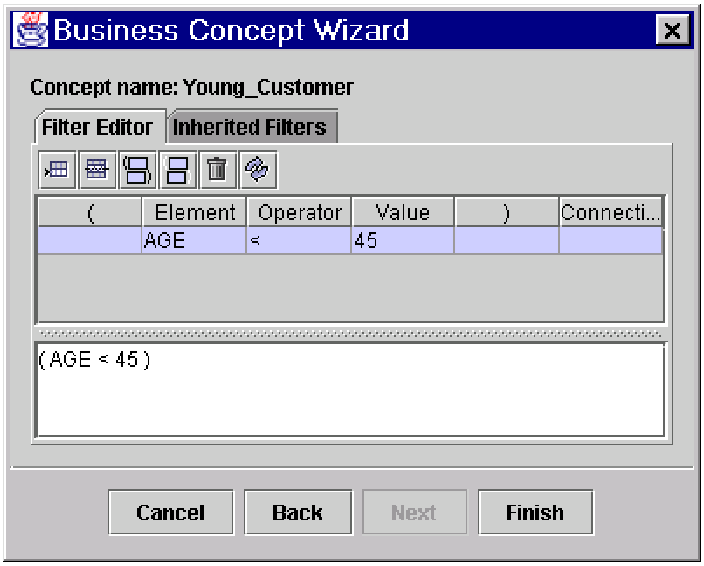
7 Click the Element field, and select the appropriate Element from the drop-down list.

8 Click the Operator field, and select the appropriate Operator from the drop-down list.

9 Click Value, and type the appropriate Value.

10 (Optional) If a more complex filter is required, click the Connective field and select the appropriate Connective from the drop-down list. Repeat steps 6, 7, and 8 for each filter that you want to associate with the Business Concept.

Figure 6-22. Filter Added



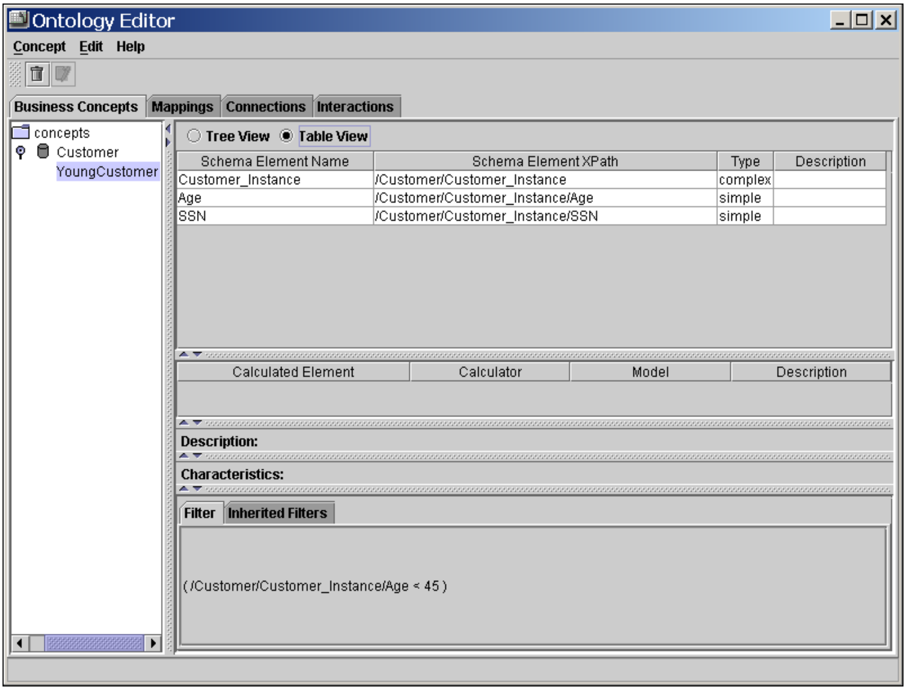
Concept name: Young_Customer

(	Element	Operator	Value	)	Connecti...
	AGE	<	45		

(AGE < 45)

11 Click Finish. The newly created filter displays in the Ontology Editor under the Parent Concept .

Figure 6-23. Ontology Editor



Schema Element Name	Schema Element XPath	Type	Description
Customer_Instance	/Customer/Customer_Instance	complex	
Age	/Customer/Customer_Instance/Age	simple	
SSN	/Customer/Customer_Instance/SSN	simple	

Calculated Element	Calculator	Model	Description

(/Customer/Customer_Instance/Age < 45)

Using Filters

Filters are used in DBCs (and also Rules) to quickly structure raw data into business objects and domain-specific categories. All filters use the principle of logical selection and combination to eliminate non-relevant data. The remaining data satisfies the defined constraints and is the instance data.

Because of XSD restrictions, comparing a literal value against a string element requires you to quote the literal value using the "`value`" convention.

Adding a Filter Row

Each filter is defined on a single filter row. If you want to add a filter, you must click on the Add Filter Row icon in the Business Concept Wizard when editing a Business Concept.

1 Start the Ontology Editor, if it is not currently open.

2 Select a Business Concept from the Concept list.

3 Option-click and select **Edit...** The Business Concept Wizard displays.

4 Select the Filter Editor tab.

5 Click the Add Filter Row icon. A new filter row displays. Complete steps 7-9 in Creating a New Derived Concept.

Clearing a Filter Row

If you want to modify the parameters for a filter, you must use the Clear Filter icon to remove the filter but retain the filter row.

1 Start the Ontology Editor, if it is not currently open.

2 Select a Business Concept from the Concept list.

3 Option-click and select **Edit...** The Business Concept Wizard displays.

4 Select the Filter Editor tab.

5 Select the filter row you wish to modify, and click the Clear Filter icon. Complete steps 7-9 in Creating a New Derived Concept.

Deleting a Filter

If you want to remove both the filter and filter row, you must use the Delete Filter icon.

1 Start the Ontology Editor, if it is not currently open.

2 Select a Business Concept from the Concept list.

3 Option-click and select **Edit...** The Business Concept Wizard displays.

4 Select the Filter Editor tab.

5 Click the Delete Filter icon. The Filter is removed.

Grouping Filter Rows

The Ontology Editor supports left-to-right precedence and creates a default grouping sequence using parentheses. You can alter calculation sequence by creating your own grouping order. The procedure is similar to the grouping operations you can perform on Rules.

1 Start the Ontology Editor, if it is not currently open.

2 Select a Business Concept from the Concept list.

3 Option-click and select **Edit...** The Business Concept Wizard displays.

4 Select the Filter Editor tab.

5 Select the first row in the group. Shift-click the next row in the group.

6 Click the Group Filters Icon. The filters are grouped.

Ungrouping Filter Rows

1 Start the Ontology Editor, if it is not currently open.

2 Select a Business Concept from the Concept list.

3 Option-click and select **Edit...** The Business Concept Wizard displays.

4 Select the Filter Editor tab.

5 Select the first row in the group to be ungrouped. Shift-click the next row in the group.

6 Click the Ungroup Filters Icon. The filters are ungrouped.

Using Filter ANDs and Filter ORs

The Filter AND and OR connectives exhibit very different characteristics that you must consider when creating DBCs.

The AND operator is true (returns instances) if all the logical expressions joined by AND are themselves true. Therefore, creating complex DBCs by chaining ANDs introduces new expressions, **all of which must be true** for the DBC to be true, or non-zero. Every time you connect with an AND, you decrease the probability that instances will exist for your DBC. You are shrinking the scope of your DBC by making it more specific.

The OR operator is true (returns instances) if at least one of the logical expressions joined by OR are themselves true. Therefore, creating complex DBCs by chaining ORs introduces new expressions, **only one of which needs to be true** for the DBC to be true, or non-zero. Every time you connect with an OR, you increase the probability that instances will exist for your DBC. You are enlarging the scope of your DBC by making it less specific and more generalized.

Understanding Calculated Elements

Calculated Elements enable you to define a set of values that work as a single value when determining real-time recommendations and running predictive models. For example, you can use a Calculated Element to create a single debt-to-ratio to predict credit risk, rather than use debt and income as two independent variables.

A Derived Root Business Concept inherits calculated elements from the Root Business Concept.

Calculated Elements are only available to Business Concepts that are not associated with a pre-defined schema.

Knowledge Broker supports the following Calculated Elements:

MLP Calculator

The MLP Calculator embeds a pre-computed neural network. These can be generated within the Knowledge Discovery Center.

CBR Calculator

The CBR Calculator embeds a pre-computed CBR model. These can be generated within the Knowledge Discovery Center.

Function

Weighted Sum Calculator

The Weighted Sum Calculator performs a weighted sum of property values. If you use only a single property as input, you can use this calculator to multiply that property's values by an arbitrary constant.

Table 0-1. Weighted Sum Calculator

Parameter	Description	Example Value
First Argument Weight	The weight of the first value.	5
First Argument Value	The first value.	10
Second Argument Weight	The weight of the second value.	2
Second Argument Value	The second Value.	12

Data Calculator

The Date Calculator calculates the difference between two dates in years/months/days.

Table 0-2. Data Calculator

Parameter	Description	Example Value
Result Format	The date format for the result	year, month, day
Date Format	The date format for input.	mm/dd/yy, yyyy/mm/dd
First Date	The date from which the second date is subtracted	
Second Date	The date to subtract from the first date.	

Age Calculator

The Age Calculator converts a date into a floating point number representing the number of years elapsed from the input date to the present.

Table 0-3. Age Calculator

Parameter	Description	Example Value
Date Format	The format the date will be input and returned.	yyyy/mm/dd, mm/dd/yy
Date	Input date.	Date of Birth

Weighted Min Calculator

The Weighted Min Calculator performs a weighted minimum of property values.

Table 0-4. Weighted Min Calculator

Parameter	Description	Example Value
First Argument Weight	The weight of the first value.	
First Argument Value	The first value.	
Second Argument Weight	The weight of the second value.	
Second Argument Value	The second value.	

Add String Calculator

The Add String Calculator appends a string to another string.

Table 0-5. Add String Calculator

Parameter	Description	Example Value
Append Position	The position of the append.	Start, End
Input String Value	The append to string.	
Append String Value	The string to be appended.	

Weighted Average Calculator

The Weighted Average Calculator calculates the weighted average of property values.s

Table 0-6. Weighted Average Calculator

Parameter	Description	Example Value
First Argument Weight	The weight of the first value.	
First Argument Value	The first value.	
Second Argument Weight	The weight of the second value.	
Second Argument Value	The second Value.	

Conditional Calculator

The Conditional Calculator evaluates a conditional expression.

Table 0-7. Conditional Calculator

Parameter	Description	Example Value
First Argument Value	The value to be compared against.	
Conditional Operator	Operators to be used in calculation.	=, >, <, .=>, =<

Table 0-7. Conditional Calculator

Parameter	Description	Example Value
Second Argument Value	The value to be compared.	

Weighted Max Calculator

The Weighted Max Calculator performs a weighted maximum of property values.

Table 0-8. Weighted Max Calculator

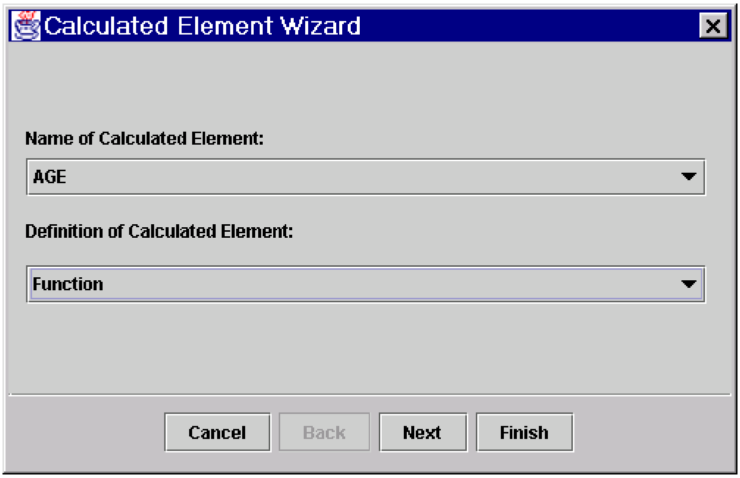
Parameter	Description	Example Value
First Argument Weight	The weight of the first value.	
First Argument Value	The first value.	
Second Argument Weight	The weight of the second value.	
Second Argument Value	The second value.	

Adding a Calculated Element

1 Open the Ontology Editor, if it is not currently open.

2 Select a Business Concept and option-click. The Calculated Element Wizard displays.

Figure 6-24. Calculated Element Wizard



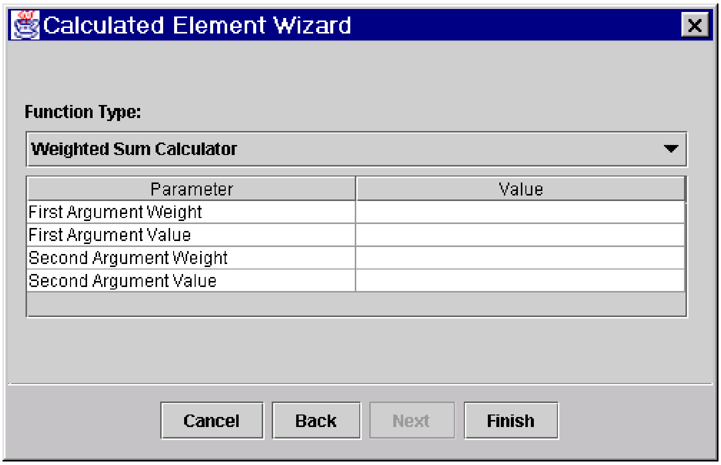
Calculated Element Wizard

Name of Calculated Element: AGE

Definition of Calculated Element: Function

3 Select the name of the Calculated Element from the drop-down list. Select a calculator from the drop-down list. The Calculated Element Wizard displays.

Figure 6-25. Calculated Element Wizard



Function Type: Weighted Sum Calculator

Parameter	Value
First Argument Weight	
First Argument Value	
Second Argument Weight	
Second Argument Value	

4 Type the required Parameters.

5 Click Finish. You have now created a new Calculated Element.

Modifying and Deleting Business Concepts

The Ontology Editor operations for modifying and deleting RBCs, DRBCs, and DBCs are similar.

Editing a Business Concept

1 Open the Ontology Editor, if it is not currently open.

2 Select the desired Business Concept.

3 Go to **Concept > Edit**. The Business Concept Wizard displays. You can move through the wizard by clicking Next and make changes, additions, or deletions to the Business Concept. When you are finished editing the Business Concept, click Finish. The changes you made are applied to the Business Concept.

Deleting a Business Concept

1 Open the Ontology Editor, if it is not currently open.

2 Select the desired Business Concept.

3 Go to **Concept > Delete**. The Confirm Deletion Dialog box displays.

4 Click Yes. You have now deleted a Business Concept.

Using Instances

Understanding Instances

Business Concepts define a class of objects and Knowledge Broker Agents, Applications, and Rules reason logically using these objects. However, when Knowledge Broker is actually processing a search for a recommendation, it creates and reasons with instances of Business Concepts.

An *Instance* is a specific record associated with a Business Concept where the instance data (that is, the data enumerated for every Element of a Business Concept) matches the Business Concept specifications in either the filter conditions (for DBCs) or the ACL (for DRBCs). For example, every record identified as a volatile stock is an Instance of the Volatile Business Concept.

Business Concepts can be thought of as abstract patterned descriptions or templates for creating concrete actualizations of the objects using real-world data. The Business Concepts contain Element "boxes" (or values). Knowledge Broker obtains data using Connections and Mappings, processes it according to Calculated Properties or other data transformations, then slots the results into the Element values to create separate Business Concept Instances.

Understanding Multiple Instances

Knowledge Broker submits multiple instances of Business Concepts as evidence.

For example, suppose you defined a Customer DBC called *Parent* with a filter condition such that `Children >= 1`. Knowledge Broker will create several instances of the *Parent* Business Concept for all Customers with a Children value greater than or equal to 1, each corresponding to a Customer with one or more children. Rules and ACL that analyze the tax brackets and income Properties of *Parent* Business Concepts may activate differently for separate *Parent* instances. This is because their tax bracket and income Property Values may vary; one *Parent* may earn $50,000 annually while another may earn $80,000 annually.

The Business Concept defines *a* Parent, but the instances describe each of *the* Parents.

Knowledge Broker can take in multiple instances of Parent (*Parent1* and *Parent2*, for example) and use both (or more) instances of the Parent DBC to produce recommendations.

This has an important impact on the flexibility of the reasoning process to deal with complex events. For example, suppose you had an RBC called *Market Conditions* with an Element called News that receives news feeds from a source such as TIBCO/rv.

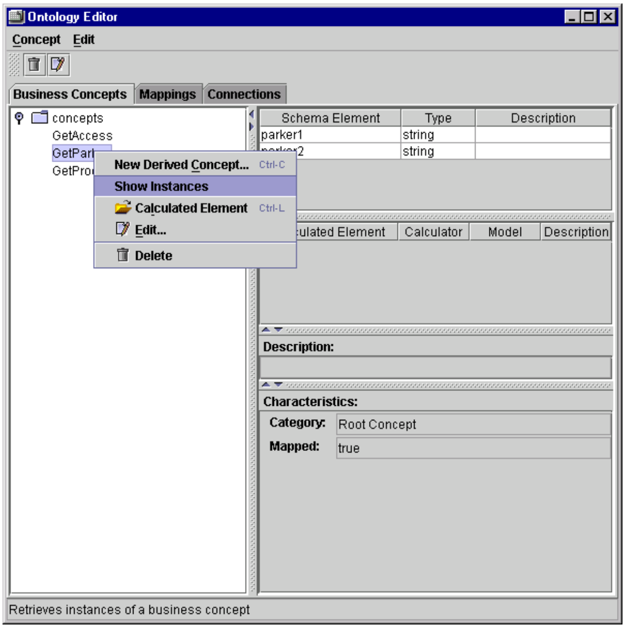
You could create a DBC called *NASDAQ Drops* using the filter condition `News = Tech Stocks Fall`. You could derive another Business Concept called *Dow Rises* using the filter condition `News = DJIA Rises`.

It is possible that the News Property Value could simultaneously input "Tech Stocks Fall" and "DJIA Rises" during the same reasoning round of Knowledge Broker. Without multiple instantiation, the later News value would obliterate all earlier values to create a single instance. However, Knowledge Broker's multiple instance functionality takes in separate instances using the different News Property Values as evidence. This parallel tasking instantiation improves the reasoning abilities of Knowledge Broker and the accuracy of its recommendations.

Showing Instances

1. Open the Ontology Editor, if it is not currently open.
2. Select the Business Concepts tab.
3. Select a Business Concept and option-click.

Figure 6-26. Show Instances



Schema Element	Type	Description
parker1	string	
parker2	string	

Description:

Characteristics:

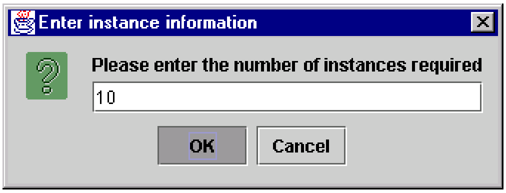
Category: Root Concept

Mapped: true

Retrieves instances of a business concept

4 Select Show Instances from the options window. The Enter Instance Dialog displays.

Figure 6-27. Enter Instance Dialog



Enter instance information

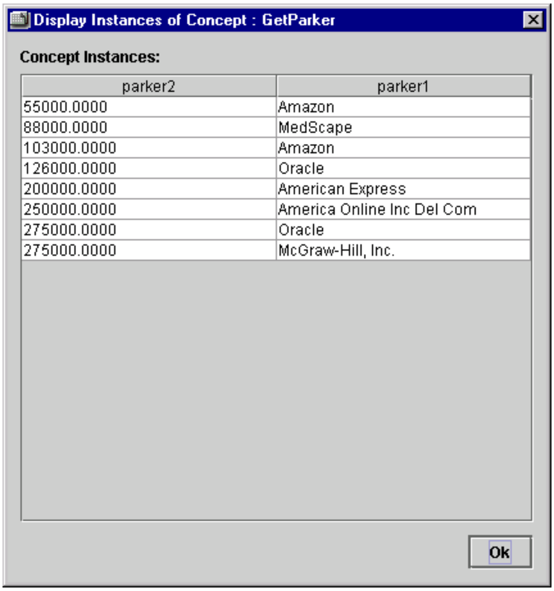
Please enter the number of instances required

10

5 Enter the number of instances required.

6 Click the **OK** button. The instances display.

Figure 6-28. Show Instances Display



Display Instances of Concept : GetParker

Concept Instances:

parker2	parker1
55000.0000	Amazon
88000.0000	MedScape
103000.0000	Amazon
126000.0000	Oracle
200000.0000	American Express
250000.0000	America Online Inc Del Com
275000.0000	Oracle
275000.0000	McGraw-Hill, Inc.

To exit the show instances window, click the **OK** button.

Chapter

7

Making Connections and Mappings

This chapter provides step-by-step instructions for defining a connection that link to external datasources and mappings that link these connections to business concepts.

Deployment Engineers alone address the complex task of establishing Mappings and Connections. Business Users do not use the Connection or Mapping functionality and should proceed to the following Ontology chapter.

Introducing Connections

Connections are the basic method Knowledge Broker uses to take in data from the outside world. Connections channel information between external datasources and Knowledge Broker. Every connection is named, and each named connection creates a unique conduit between Knowledge Broker and the datasource. Using the name as a reference, Knowledge Broker requests a set of some or all of the data from a datasource using supplied credentials and connection parameters specific to that datasource. Using the named connection, Knowledge Broker can publish or put data back to external datasources.

Read and write access availability to external datasources depends on the role-based permissions of individual users and groups, as configured by system administrators and Deployment Engineers, and the supply of appropriate names, passwords, and identification tokens.

The Ontology Editor consists of three tabs: Business Concepts, Mappings, and Connections. The Mappings and Connections tabs are used by the Deployment Engineer (the Business User uses only the Business Concepts tab). The Deployment Engineer can accomplish the following tasks using the Ontology Editor:

- Create a Connection between an external datasource and Knowledge Broker using the Connection Wizard.
- Create a Mapping between a Connection and a Business Concept using the Mapping Wizard.

Types of Datasources

There are two main categories of datasource: persistent and event.

Persistent datasources store their data in databases, files, web pages, warehouses, or other generally static "silos." The silos' contents may change between accesses on a random or periodic basis, but during a particular connection access window the instance data is typically stable.

Event datasources store their data in message-based queues or streams, such as MQ, TIBCO/rv, or JMS. The stream's contents change dynamically, with old messages being continually retired and new messages replacing them.

Types of Connections

Knowledge Broker ships with ten types of Connection defined:

1. RDBMS
2. Sybase
3. File
4. MS SQL
5. TIBCO
6. ODBC
7. DB2
8. Oracle
9. JMS
10. WWW

RDBMS

RDBMS or Relational Database Connections point to a variety of structured datasources. RDBMS Connections are persistent.

Sybase

Sybase Connections feature a custom template that reduces the number and complexity of required user-supplied connection parameters when connecting to Sybase RDBMS persistent datasources.

File

FILE Connections Point to XML files stored locally on a user's hard disk or within designated directories on the local network. FILE Connections are persistent.

MS SQL

Microsoft SQL Server Connections feature a custom template that reduces the number and complexity of required user-supplied connection parameters when connecting to RDBMS persistent datasources.

TIB Rendezvous

TIBCO Connections point to a Tibco Rendezvous message bus. These Connections are event.

ODBC

Open Database Connectivity (ODBC) is an open standard API for accessing a database.

DB2

DB2 Connections feature a custom template that reduces the number and complexity of required user-supplied connection parameters when connecting to IBM DB2 RDBMS persistent datasources.

Oracle

Oracle Connections feature a custom template that reduces the number and complexity of required user-supplied connection parameters when connecting to Oracle RDBMS persistent datasources.

JMS

The Java Message System (JMS) is a Javasoft-specified common interface for communication between J2EE platforms. JMS Connections are event-based.

WWW

WWW or Web Connections point to XML or HTML 4.x documents that can be retrieved using the HTTP protocol from online web servers. WWW Connections are persistent-based.

Beginning Connections

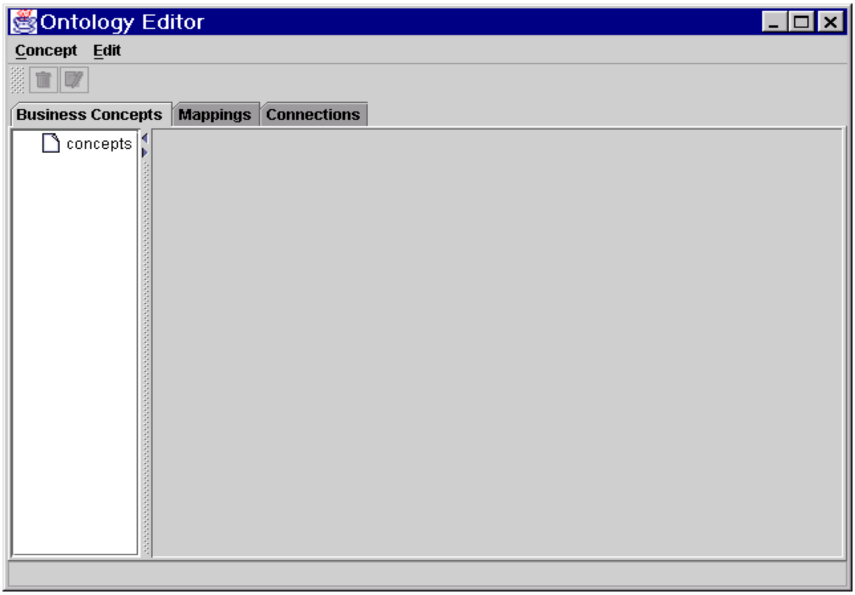
To begin making Connections, Deployment Engineers (DEs) must create a new knowledge base or open and edit an existing knowledge base. DEs use the Ontology Editor to create Connections.

Opening the Ontology Editor

Select the Knowledge Broker Launcher.

Click on the Ontology Editor button. The Ontology Editor displays

Figure 7-1. The Ontology Editor



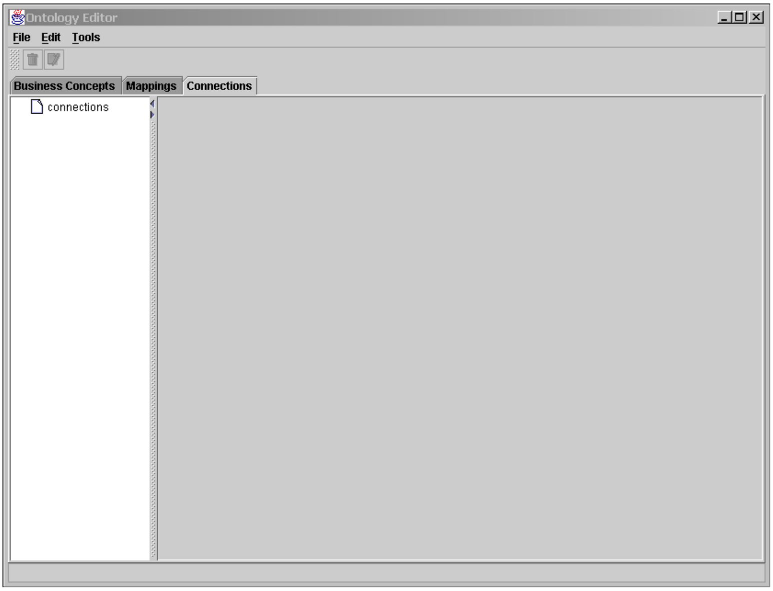
Accessing the Connections Tab

Initially, the Ontology Editor displays with the Business Concepts tab highlighted. This is the basic Business User functionality. To select the DEs Connections functionality, click the Connections tab.

Business Users will be unable to select either the Connections or Mappings tabs or functionality.

The Ontology Editor Connections tab displays. For empty knowledge bases, this appears very similar to the Ontology Editor Business Concepts tab.

Figure 7-2. Ontology Editor Connections tab

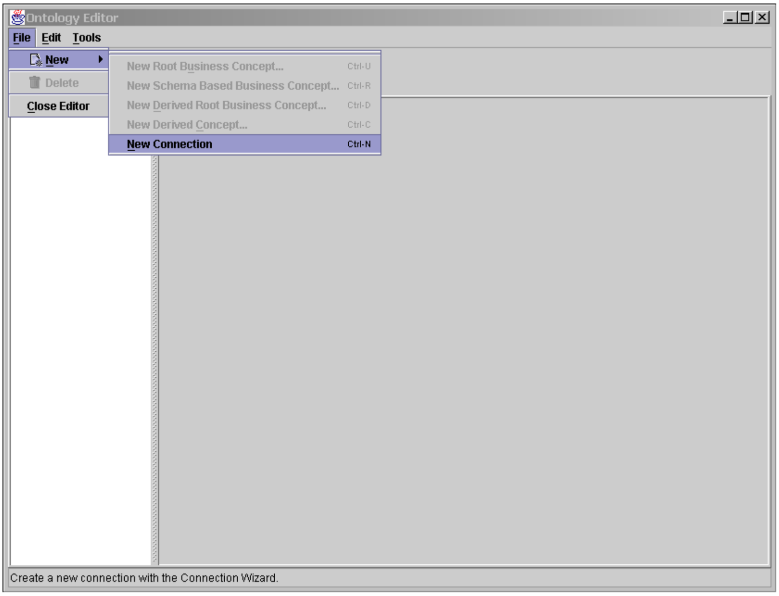


Activating the Connection Wizard

1 Highlight the connections root property in the left-hand panel.

2 Select **Concept > New > New Connection.**

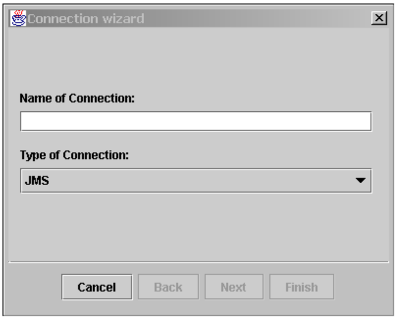
Figure 7-3. File > New > New Connection


Ontology Editor
File
Edit
Tools
New
Delete
Close Editor
New Root Business Concept... Ctrl-U
New Schema Based Business Concept... Ctrl-R
New Derived Root Business Concept... Ctrl-D
New Derived Concept... Ctrl-C
New Connection Ctrl-N

Create a new connection with the Connection Wizard.

3 The Connection Wizard displays.

Figure 7-4. Connection Wizard



Name of Connection:

Type of Connection: JMS

You can also access the Connection Wizard by highlighting the connections root element and option-clicking.

Creating Connections

All Connections share a similar creation procedure. You access the Connection Wizard, type a name for your connection, and select the Type of Connection.

The Connection Wizard then displays a list of properties and parameters you must complete to specify the connection. Some of these will be defaulted and supplied by Knowledge Broker Server. Others will be specific, such as access credentials, and you must enter these yourself.

If a property is displayed as “required”, then it must be present for Knowledge Broker to establish a Connection. Non-required properties are Knowledge Broker optional, but often required by particular Connections.

Using Auto-Generated or User-Created Schemas

Schemas are required to enable Knowledge Broker to correctly “channel” the external data to and from the ontology. You will create Mappings between Business Concept Elements and Source Schema Elements.

The Connection Wizard prompts you to select a suitable Schema to describe your data connection. A Schema contains meta-information about the types and relations of the information in the selected datasource. Users with XML Schema coding ability can hand-craft or fine-tune the Schemas. These are stored in the `[KBInstallDir]\config\Data\SourceSchemas` directory.

Knowledge Broker Schemas are saved as XML Schema (`.XSD`) files.

The Connection Wizard can automatically generate Schemas using data discovery on target datasources. Currently, this functionality is available for these connectors:

- MS Access (ODBC)
- MS Excel (ODBC)
- DB2
- Oracle
- Sybase
- MS MS SQL (RDBMS)
- WebLogic JMS (requires a sample message file)
- Tibco (requires a sample message file)

The auto-generate feature does not presently function for these connectors:

- File (XML format) and WWW (html format).

Creating an RDBMS Connection

The RDBMS Connection functions as a channel for many structured datasources. The RDBMS Connection currently supports the following Connections: Oracle, Microsoft Access, and Microsoft Excel.

RDBMS Connection Wizard Access Properties

The RDBMS Connection Wizard Access Properties are

Table 7-1. RDBMS Connection Wizard Access Properties

Name	Description	Required
connectionURL	Specifies the RDBMS records using a Connection String defined for a particular RDBMS.	Yes
driver	Identifies the Java class that contains a driver suitable for communication with the RDBMS. This class file must be accessible through the Java `CLASSPATH` environment setting.	Yes

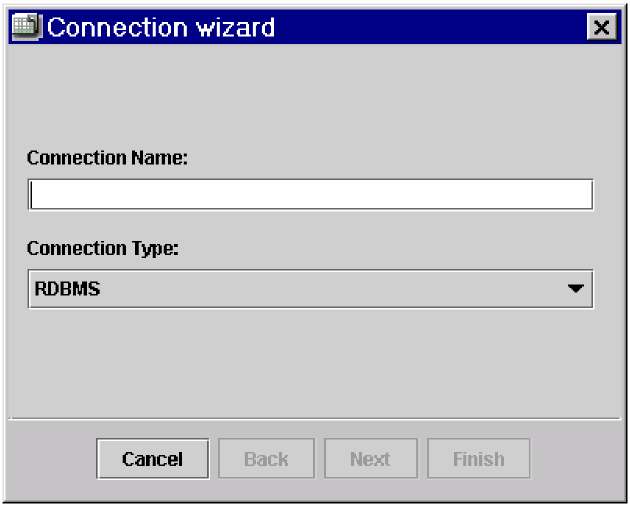
RDBMS Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections Tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select RDBMS from the Connection Type drop-down menu.

Figure 7-5. Connection Wizard - RDMBS



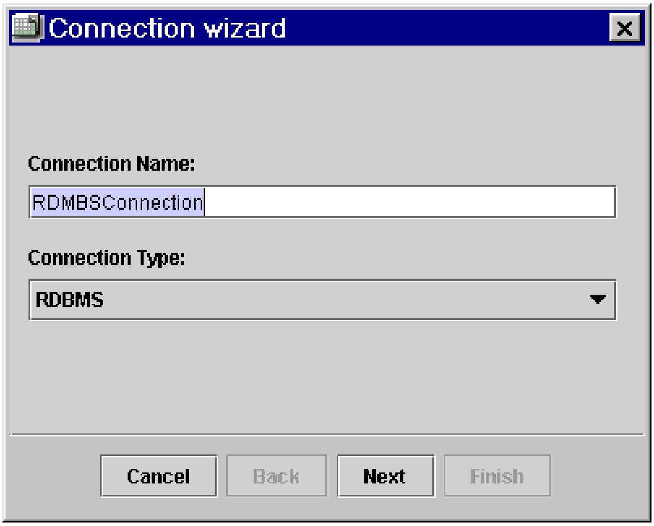
Connection wizard

Connection Name:

Connection Type: RDBMS

4 Type a name for the RDBMS Connection.

Figure 7-6. Connection Wizard - RDBMS - Connection Name

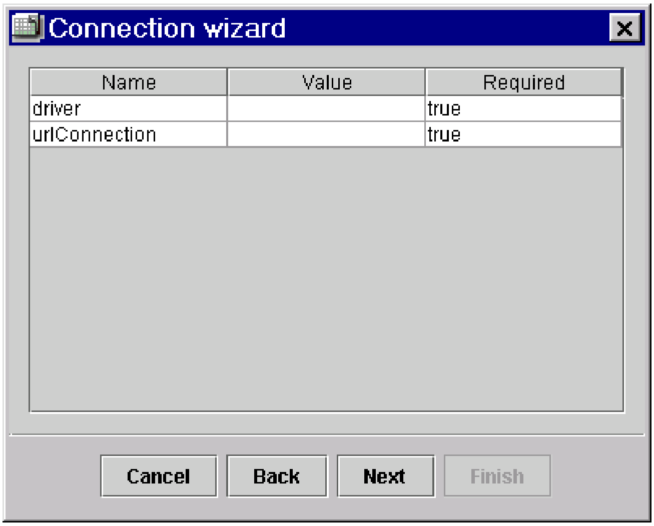


Connection Name: RDMBSConnection

Connection Type: RDBMS

5 Click Next. The Connection Wizard now displays the RDBMS connection properties.

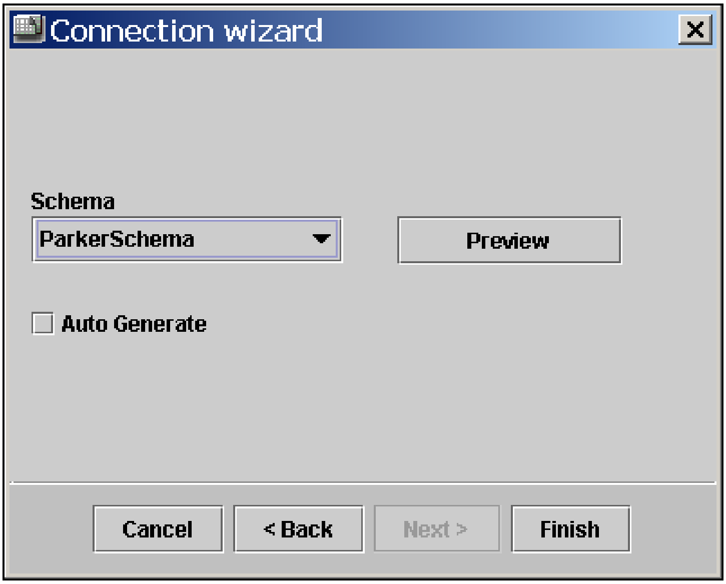
Figure 7-7. Connection Wizard - RDBMS - Connection Properties



Name	Value	Required
driver		true
urlConnection		true

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-8. Connection Wizard - RDBMS - Schema Selector



Schema: ParkerSchema

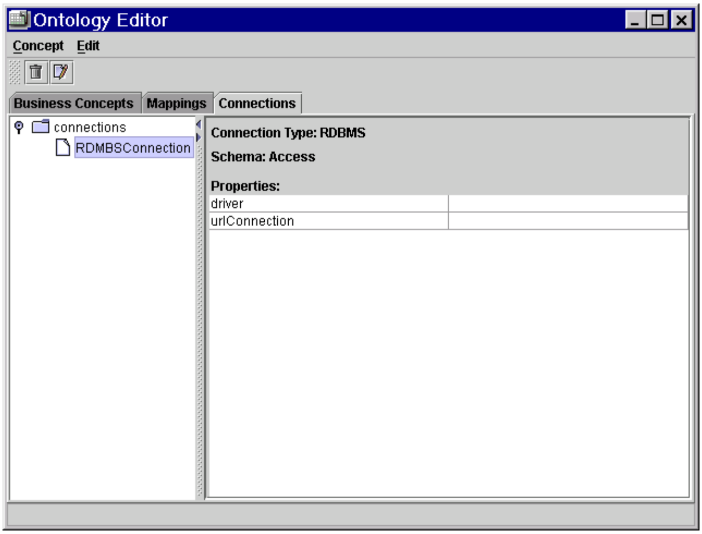
□ Auto Generate

7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connections Wizard. You return to the Connections tab of the Ontology Editor.

Figure 7-9. RDBMS - Connection



Connection Type: RDBMS

Schema: Access

Properties:

driver	
urlConnection	

Creating a Sybase Connection

Sybase Connection Wizard Properties

The Sybase Connection Properties are

Table 7-2. Sybase Connection Wizard Properties

Name	Description	Required
serverName	Identifies the Sybase server host using standard IP addressing.	Yes
password	Supplies the password credentials of a principal authorized to access the Sybase Connection.	Yes
portNumber	Identifies the port used for communication with the Sybase server.	Yes
protocol	Identifies the protocol used for communication with the Sybase Connection.	Yes
driver	Identifies the Java class that contains a driver suitable for communication with the Sybase Connection. This class file must be accessible through the Java `CLASSPATH` environment setting.	Yes
UserName	Specifies connection name.	Yes
URLPrefix	odbc:sybase	Yes
timeout	Specifies how long in milliseconds to wait for a response from the Sybase Connection.	No

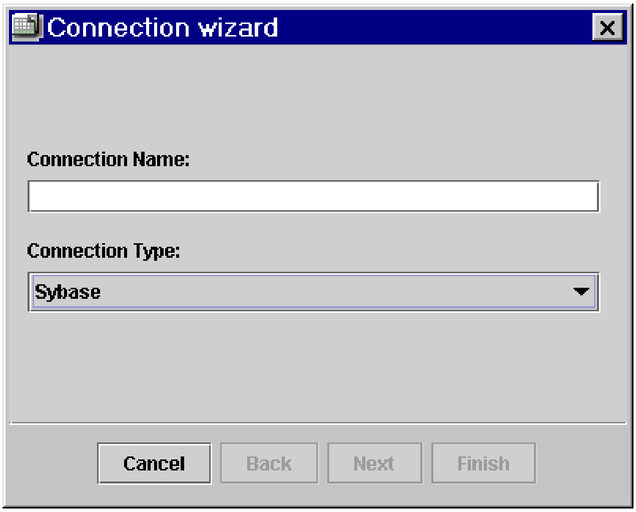
Sybase Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select Sybase from the Connection Type drop-down menu.

Figure 7-10. Connection Wizard - Sybase



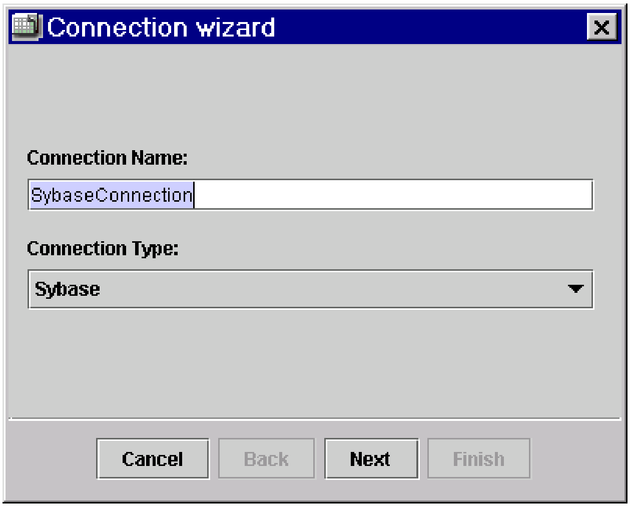
Connection wizard

Connection Name:

Connection Type: Sybase

4 Type a name for the Sybase Connection.

Figure 7-11. Connection Wizard - Sybase - Connection Name

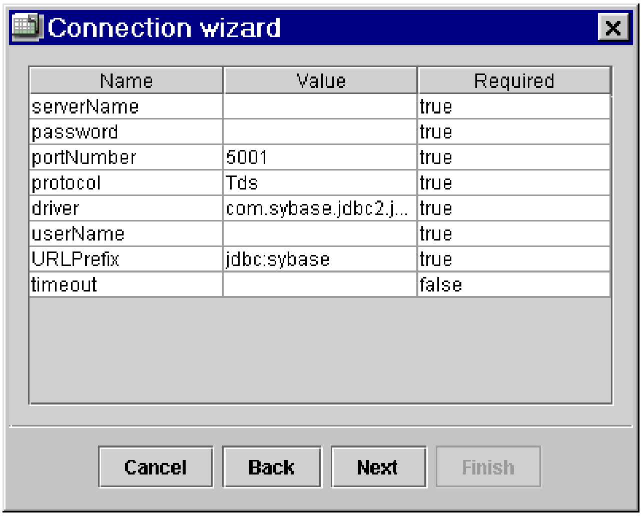


Connection Name: SybaseConnection

Connection Type: Sybase

5 Click Next. The Connection Wizard now displays the Sybase connection properties.

Figure 7-12. Connection Wizard - Sybase - Connection Properties

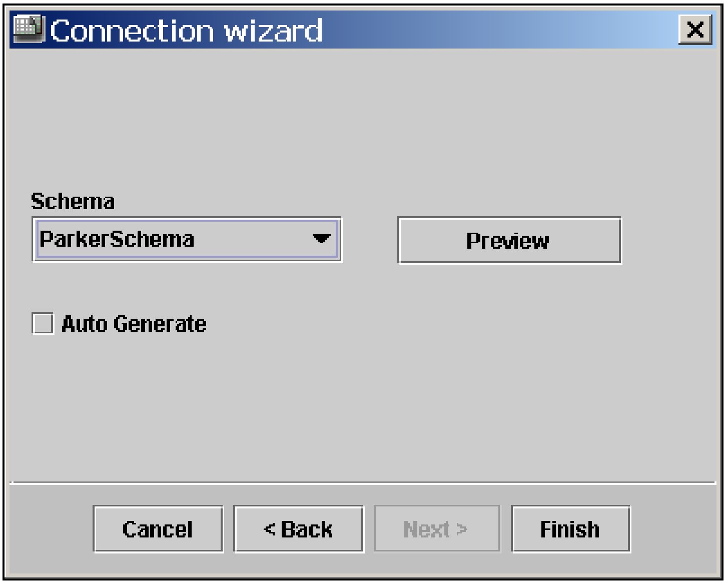


Connection wizard

Name	Value	Required
serverName		true
password		true
portNumber	5001	true
protocol	Tds	true
driver	com.sybase.jdbc2.j...	true
userName		true
URLPrefix	jdbc:sybase	true
timeout		false

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-13. Connection Wizard - Sybase - Schema Selector

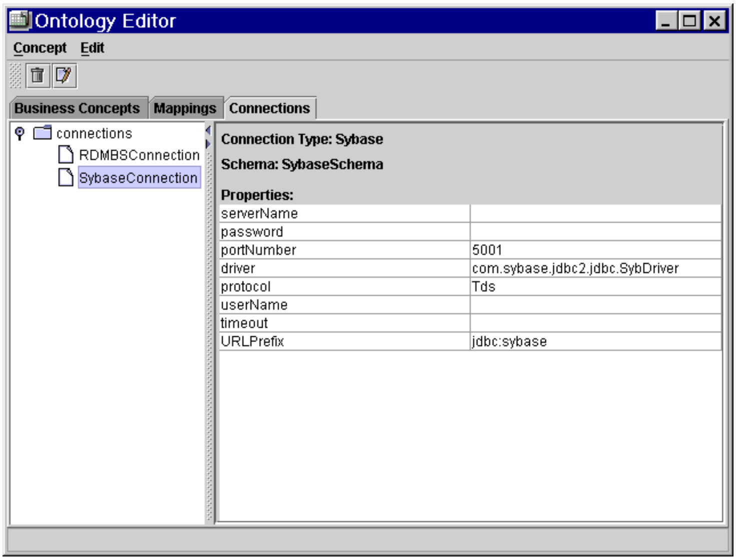


7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connections Wizard. You return to the Connections tab of the Ontology Editor.

Figure 7-14. Sybase - Connection



Connection Type: Sybase

Schema: SybaseSchema

Properties:

serverName	
password	
portNumber	5001
driver	com.sybase.jdbc2.jdbc.SybDriver
protocol	Tds
userName	
timeout	
URLPrefix	jdbc:sybase

Creating a File Connection

FILE Connections Point to XML files stored locally on a user's hard disk or within designated directories on the local network. FILE Connections are persistent-based.

File Connection Wizard Properties

The File Connection Wizard Properties are

Table 7-3. File Connection Wizard Properties

Name	Description	Required
url	Path name that points to a file.	Yes
Schema	Specifies the Schema name.	Yes
name	Specifies connection name.	Yes
connectorName	Contains a short string that prefixes the connector URL and helps to identify the type of resource and protocol required.	Yes

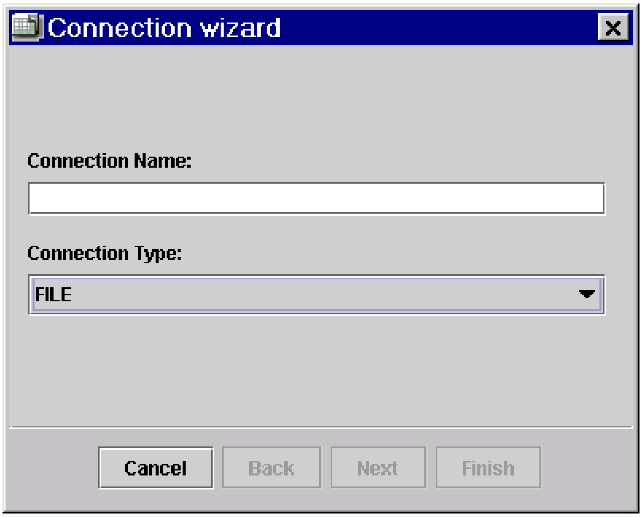
File Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select File from the Connection Type pull-down menu.

Figure 7-15. Connection Wizard - File

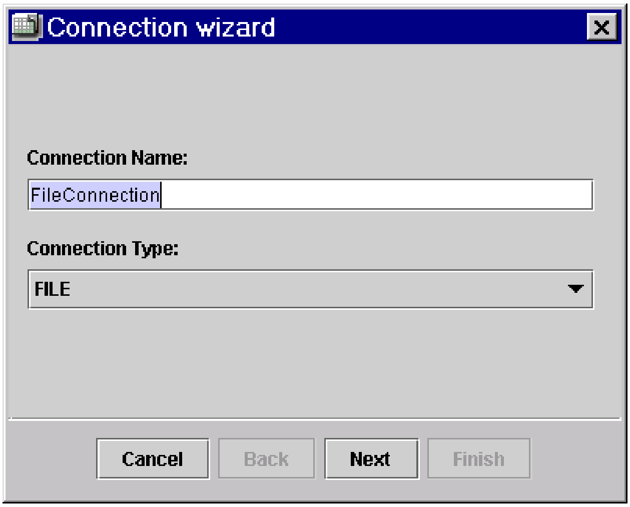


Connection Name:

Connection Type: FILE

4 Type a name for the File Connection.

Figure 7-16. Connection Wizard - File - Connection Name.

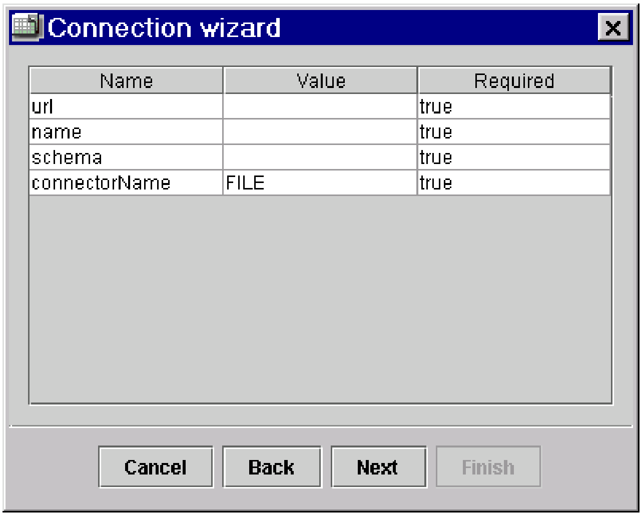


Connection Name: FileConnection

Connection Type: FILE

5 Click Next. The Connection Wizard now displays the File connection properties..

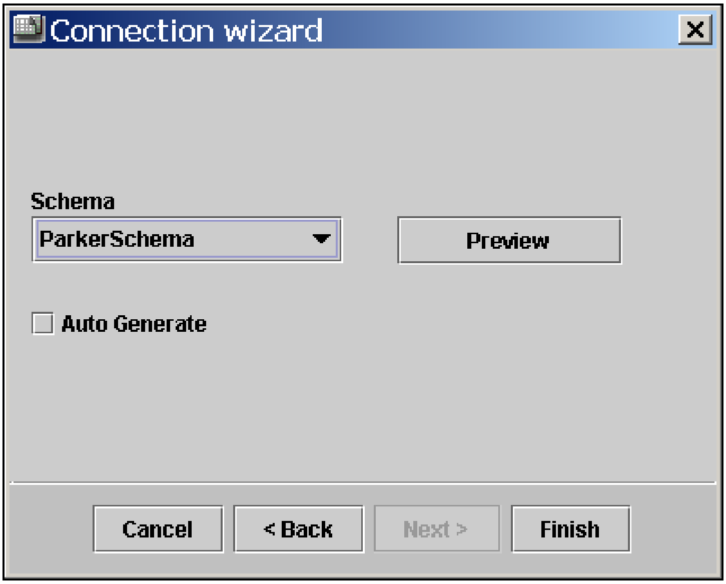
Figure 7-17. Connection Wizard - File - Connection Properties



Name	Value	Required
url		true
name		true
schema		true
connectorName	FILE	true

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-18. Connection Wizard - File - Schema Selector



Schema: ParkerSchema

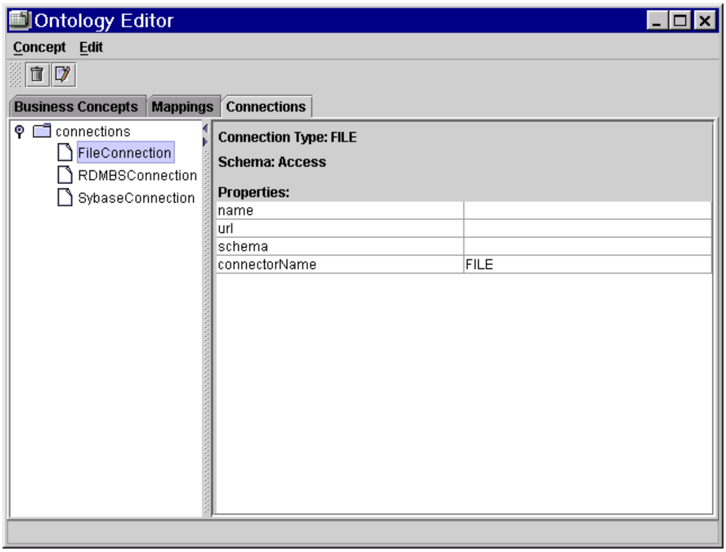
☐ Auto Generate

7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connections Wizard. You return to the Connections tab of the Ontology Editor.

Figure 7-19. File - Connection



Connection Type: FILE

Schema: Access

Properties:

name	
url	
schema	
connectorName	FILE

Creating a MS SQL Connection

Microsoft SQL Server Connections feature a custom template that reduces the number and complexity of required user-supplied connection parameters when connecting to persistent datasources.

MS SQL Connection Wizard Properties

The MS SQL Connection Wizard Properties are

Table 7-4. Tibco Connection Wizard Properties

Name	Description	Required
serverName	Identifies the MS SQL server host using standard IP addressing.	Yes
portNumber	Identifies the port used for communication with the MS SQL server.	Yes
UserName	Supplies the username credentials of a principal authorized to access the MS SQL.	Yes
password	Supplies the password credentials of a principal authorized to access the MS SQL.	Yes
driver	Identifies the Java class that contains a driver suitable for communication with the MS SQL.	Yes
timeout	Specifies how long in milliseconds to wait for a response from the MS SQL Server.	No
databaseSchema	Specifies to which Schema the connection should attach.	NO
URLPrefix	Specifies the MS SQL records using a Connection String defined for a particular MS SQL.	Yes

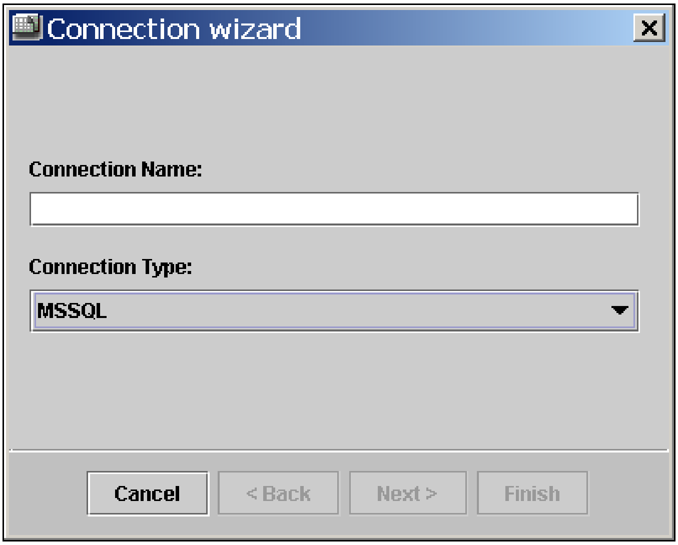
MS SQL Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select MS SQL from the Connection Type pull-down menu.

Figure 7-20. Connection Wizard - MS SQL

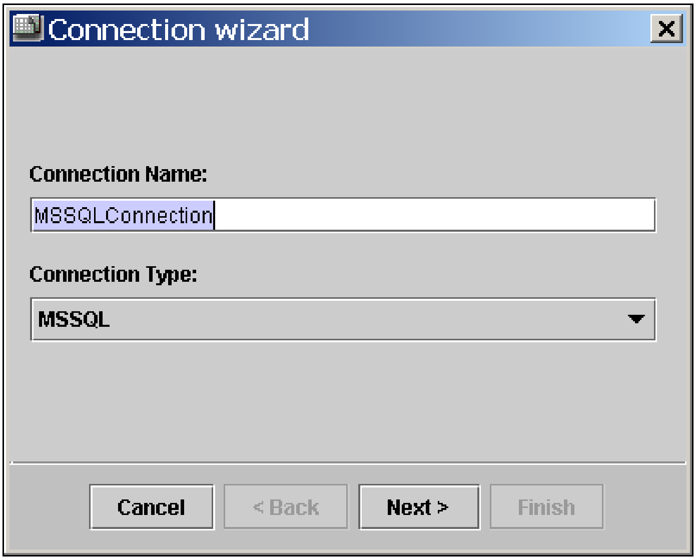


Connection Name:

Connection Type: MSSQL

4 Type a name for the MS SQL Connection.

Figure 7-21. Connection Wizard - MS SQL - Connection Name.

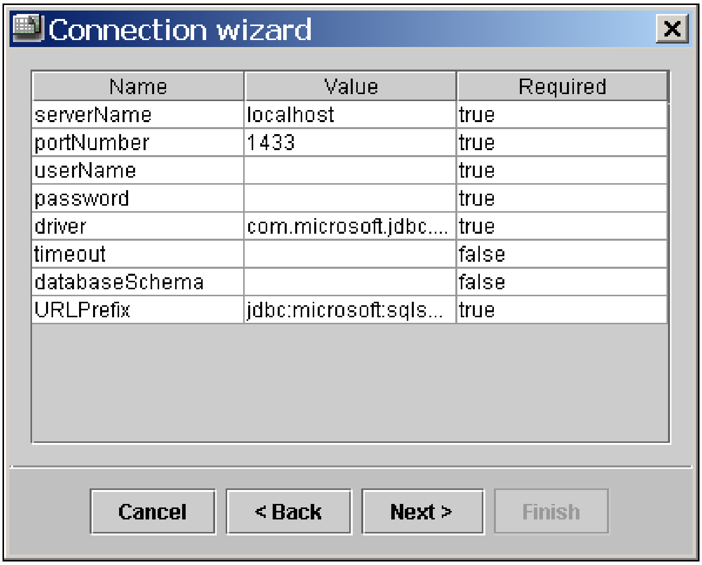


Connection Name: MSSQLConnection

Connection Type: MSSQL

5 Click Next. The Connection Wizard now displays the MS SQL connection properties..

Figure 7-22. Connection Wizard - MS SQL - Connection Properties

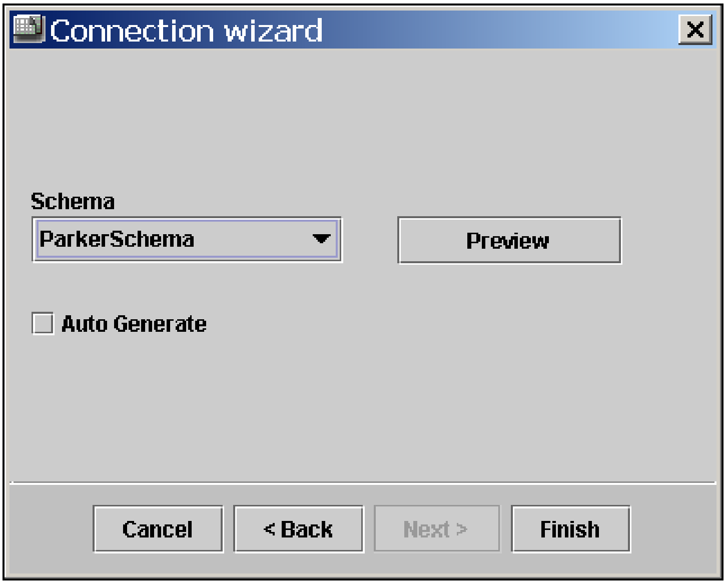


Connection wizard

Name	Value	Required
serverName	localhost	true
portNumber	1433	true
userName		true
password		true
driver	com.microsoft.jdbc....	true
timeout		false
databaseSchema		false
URLPrefix	jdbc:microsoft:sqls...	true

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-23. Connection Wizard - MS SQL - Schema Selector



Connection wizard

Schema

ParkerSchema

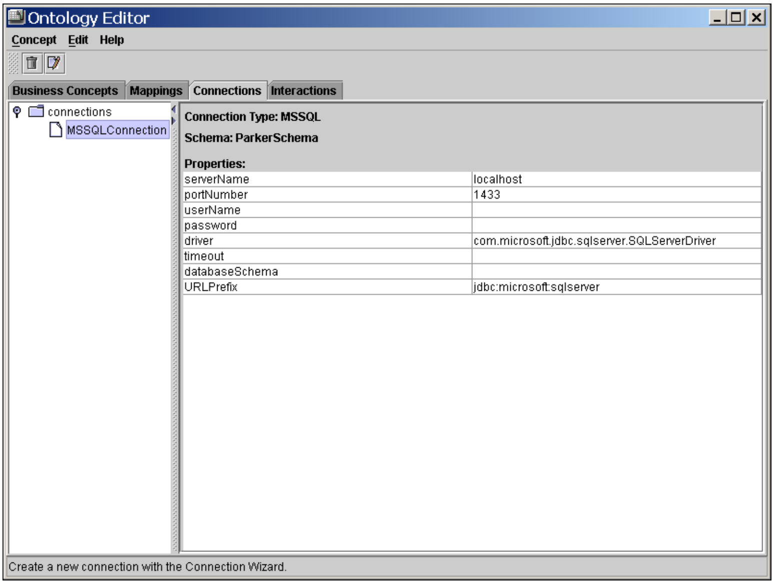
Auto Generate: □

7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connections Wizard. You return to the Connections tab of the Ontology Editor.

Figure 7-24. File - Connection



Connection Type: MSSQL

Schema: ParkerSchema

Properties:

serverName	localhost
portNumber	1433
userName	
password	
driver	com.microsoft.jdbc.sqlserver.SQLServerDriver
timeout	
databaseSchema	
URLPrefix	jdbc:microsoft:sqlserver

Create a new connection with the Connection Wizard.

Creating a Tibco Connection

The TIBCO Connection enables Knowledge Broker to communicate with TIB/Rendezvous-enabled applications that exchange data across a TIB network.

Tibco Connection Wizard Properties

The Tibco Connection Wizard Properties are

Table 7-5. Tibco Connection Wizard Properties

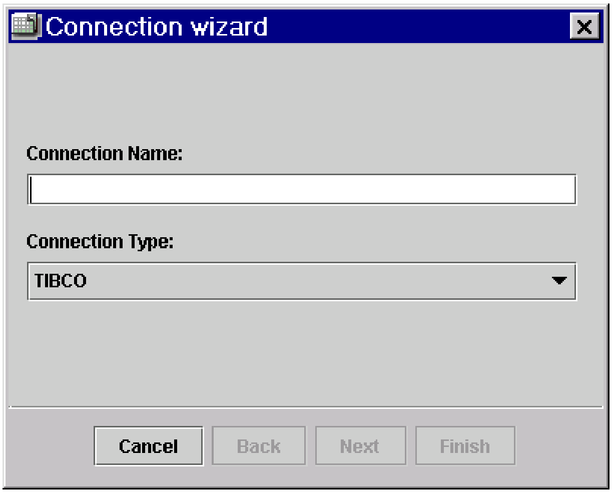
Name	Description	Required
connectorName	Contains a short string that prefixes the connector and helps to identify the type of resource and protocol required. The default here is `TIBCO`.	Yes
defaultDestination	This has the same functionality as "subject" in the Rendezvous API (Version 6). It is a unique string that identifies a message, and several messages can have same subject.	No
distributed Destination	This is a TIB/Rendezvous administrative setting that enables you to forward messages between different message segments connected by TIB/Rendezvous routing daemons.	No
cmName	This is a Rendezvous API (Version 6) Network Transport Parameter. `cmName` is a reusable name variable that enables sending programs to anticipate listeners (and hence requests for certified delivery before listening programs actually register). `cmName` also enables sending programs to accept requests for certified delivery from listeners with corresponding names.	No
Schema	Identifies the named XML Schema file that describes how to convert between the TIBCO/Rv message format and Knowledge Broker's internal in-memory representations.	Yes
Service	This is a Rendezvous API (Version 6) Network Transport Parameter. Different values of `service` can isolate independent distributed applications running on the same network from each other.	No
name	Uniquely identifies the connection.	Yes
network	This is a Rendezvous API Network Transport Parameter. It controls multicast addressing or addressing a specific outbound network interface on systems with multiple network interfaces. Selecting the null value specifies the default NIC. The format is a standard IP address in dotted decimal: `XXX.XXX.XXX.XXX`.	Yes
Daemon	This is a Rendezvous API Network Transport Parameter. You can specify a particular remote TIB/Rendezvous daemon using the `daemon` parameter.	No

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays Select TIBCO from the Connection Type pull-down menu..

Figure 7-25. Connection Wizard - TIBCO



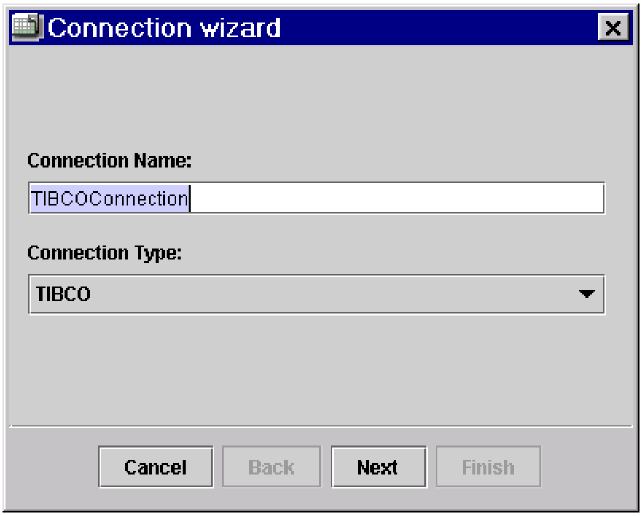
Connection wizard

Connection Name:

Connection Type: TIBCO

4 Type a name for the Tibco Connection.

Figure 7-26. Connection Wizard - TIBCO - Name of Connection



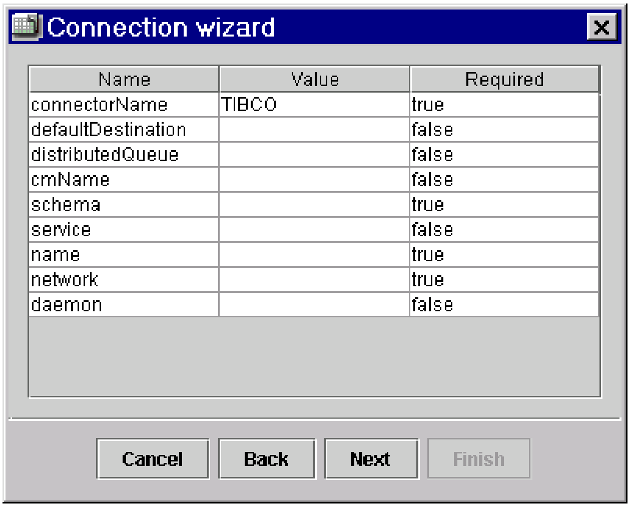
Connection wizard

Connection Name: TIBCOConnection

Connection Type: TIBCO

5 Click Next. The Connection Wizard displays the TIBCO connection properties .

Figure 7-27. Connection Wizard - TIBCO - Connection Properties

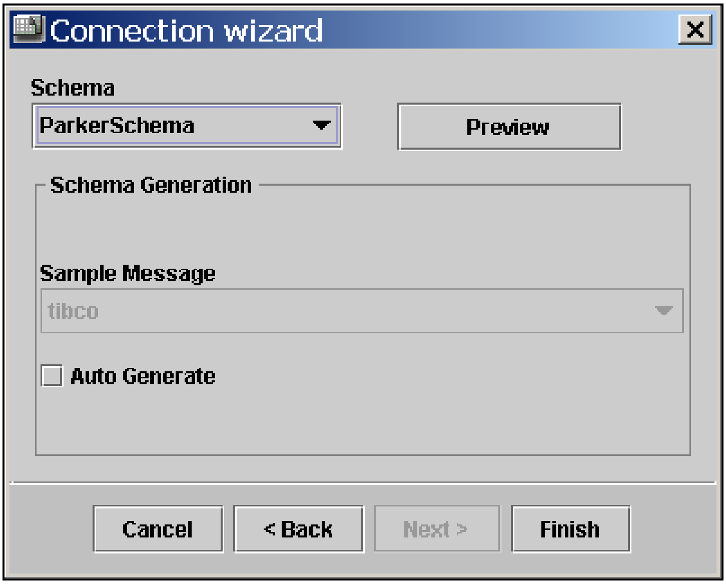


Name	Value	Required
connectorName	TIBCO	true
defaultDestination		false
distributedQueue		false
cmName		false
schema		true
service		false
name		true
network		true
daemon		false

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays. This dialog enables you to select an

instance of a TIBCO message as a sample input for schema generation. You can select Preview to preview your selection in the Schema Preview Dialog.

Figure 7-28. Connection Wizard - TIBCO - Schema Selector



Schema: ParkerSchema

Schema Generation

Sample Message: tibco

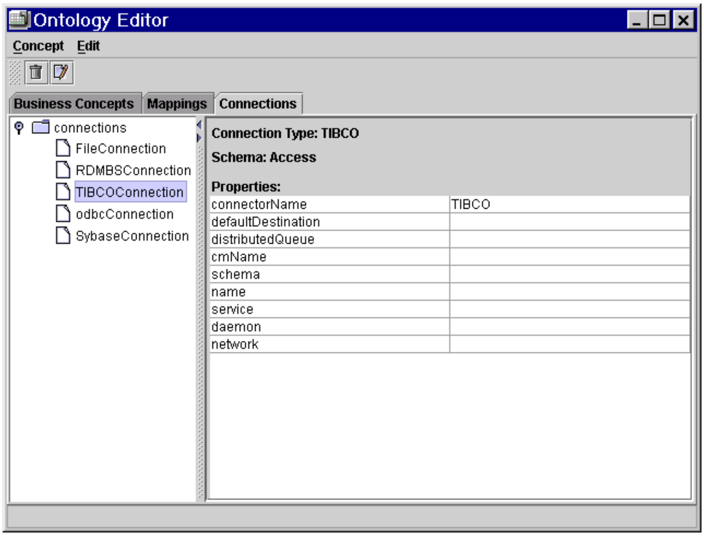
□ Auto Generate

To auto-generate a Schema, select the Auto Generate checkbox.

Because of implementation restrictions, the presence of a colon character (**:**) in a sample TIBCO message will halt the auto-generation of Tibco Schema.
This restriction will be removed with a later release of Knowledge Broker.

7 Click Finish to exit the Connections Wizard. You return to the Connections tab of the Ontology Editor.

Figure 7-29. TIBCO - Connection



Connection Type: TIBCO

Schema: Access

Properties:

connectorName	TIBCO
defaultDestination	
distributedQueue	
cmName	
schema	
name	
service	
daemon	
network	

Creating an ODBC Connection

The Microsoft ODBC datasource Administrator provides a convenient single location that aggregates many datasource drivers for easy access.

ODBC Connection Wizard Properties

The ODBC Connection Wizard Properties are

Table 7-6. ODBC Connection Wizard Properties

Name	Description	Required
driver	Identifies the Java class that contains a driver suitable for communication with the ODBC. This class file must be accessible through the Java `CLASSPATH` environment setting. For example, the default driver for the ODBC connection is `sun.jdbc.Odbc.JdbcOdbc.Driver`.	Yes
password	Supplies the password credentials of a principal authorized to access the ODBC.	No
userName	Supplies the username credentials of a principal authorized to access the ODBC	No
timeout	Specifies how long in milliseconds to wait for a response from the ODBC.	No
ODBCname	Specifies connection name.	Yes
protocol	Identifies the protocol used for communication with the RDBMS. For example, the default protocol for the ODBC connector is `jdbc:odbc`.	Yes

Creating an ODBC Driver

To create and specify system-wide ODBC drivers, follow these steps:

1. Select **Start > Settings > Control Panel > ODBC datasource** (Windows NT4) or **Start > Settings > Control Panel > Administrative Tools > datasources (ODBC)** (Windows 2000). The ODBC datasource Administrator displays, listing all currently configured datasource drivers.
2. Click the System DSN tab. The ODBC datasource Administrator/System DSN dialog displays.
3. Click Add. The Create New datasource dialog displays.
4. Select the appropriate datasource driver.
5. Click Finish. The ODBC datasource Administrator dialog re-displays.
6. Click Ok. The ODBC Setup dialog displays.

7 Complete each of the Setup fields, including the path to the datasource.

Contact your Database Administrator or read the database documentation for the exact information to enter in the ODBC Setup dialog.

8 Click OK. The System DSN dialog re-displays.

9 Click Ok again. The Control Panel re-displays.

10 Exit the Control Panel.

If you change datasource types, you will need to modify the ODBC Setup dialog information. Select **Start > Settings > Control Panel > ODBC datasources**. Click System DSN. Select the driver to be modified. Click Modify. Make the appropriate changes..

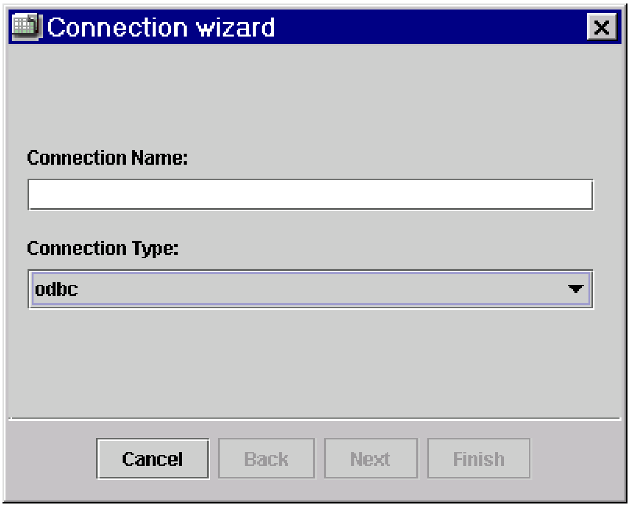
ODBC Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays Select ODBC from the Connection Type pull-down menu.

Figure 7-30. Connection Wizard - ODBC



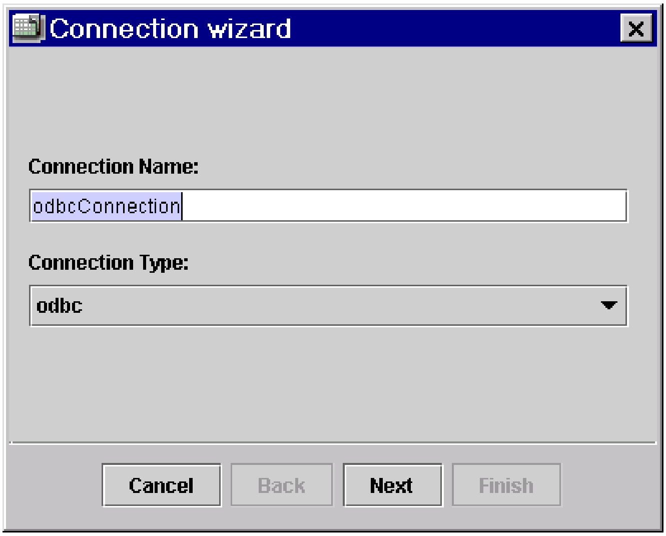
Connection wizard

Connection Name:

Connection Type: odbc

4 Type a name for the ODBC Connection.

Figure 7-31. Connection Wizard - ODBC - Name of Connection

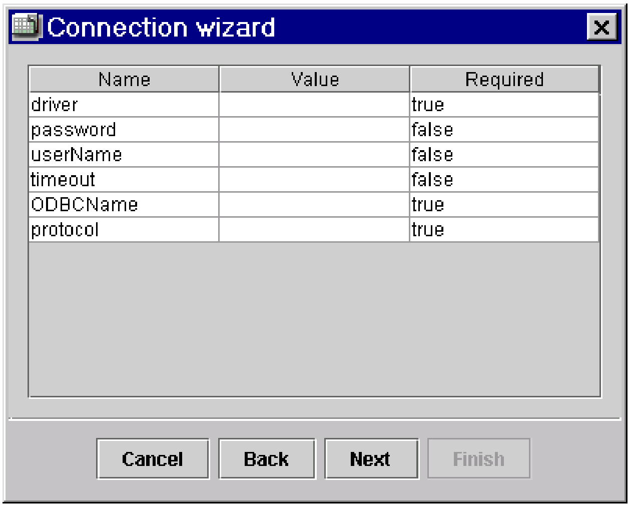


Connection Name: odbcConnection

Connection Type: odbc

5 Click Next. The Connection Wizard display the ODBC connection properties.

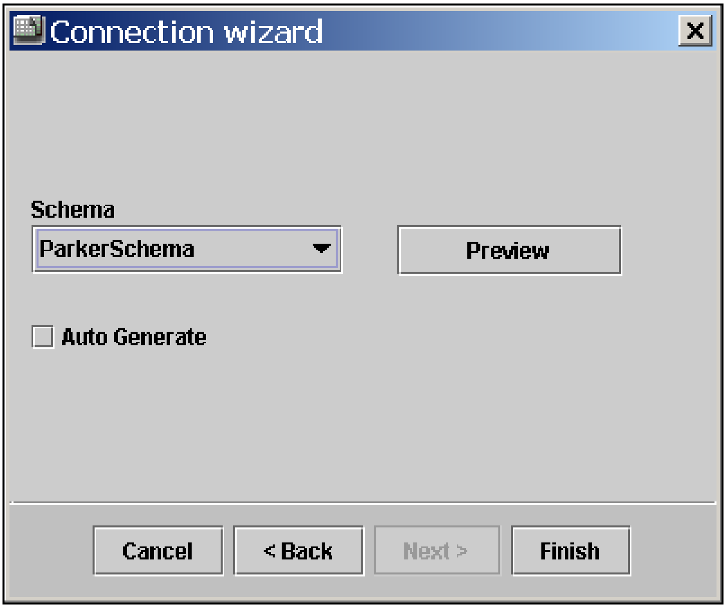
Figure 7-32. Connection Wizard - ODBC - Connection Properties



Name	Value	Required
driver		true
password		false
userName		false
timeout		false
ODBCName		true
protocol		true

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-33. Connection Wizard - ODBC - Schema Selector



Connection wizard

Schema: ParkerSchema

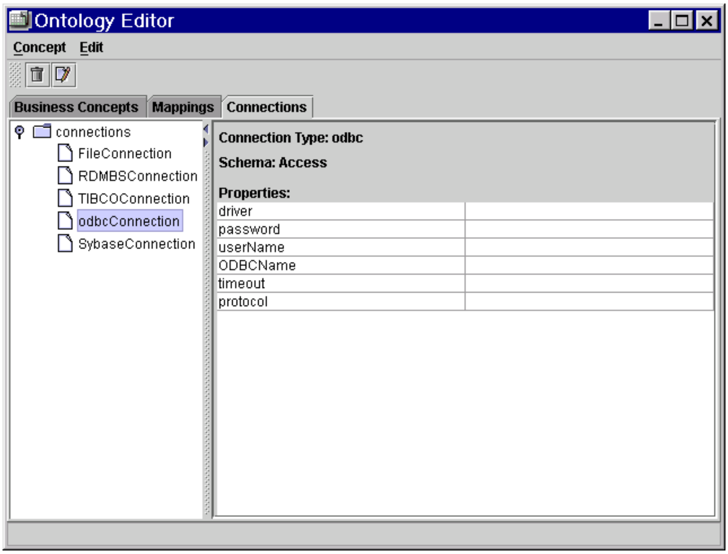
□ Auto Generate

7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connections Wizard. You return to the Connections tab of the Ontology Editor.

Figure 7-34. ODBC - Connection



Connection Type: odbc

Schema: Access

Properties:

driver	
password	
userName	
ODBCName	
timeout	
protocol	

Microsoft Access Connection (Windows-only)

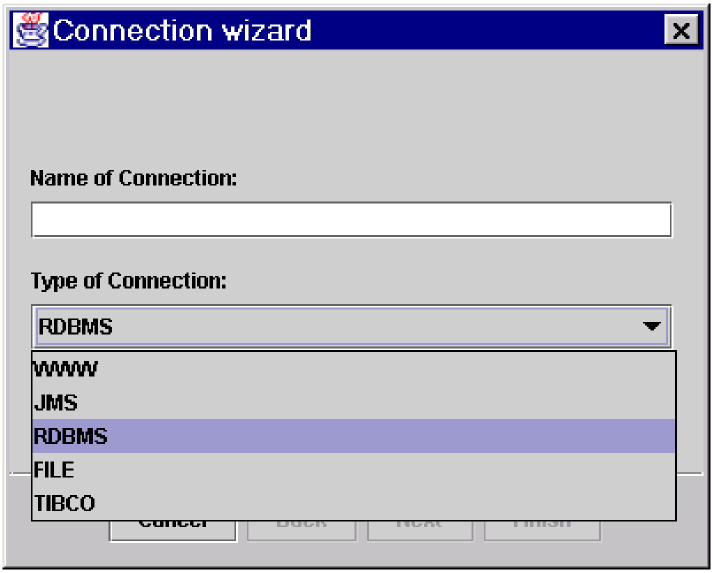
The Microsoft Access Connection is for Windows machines only.

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays.

Figure 7-35. Microsoft Access Connection Wizard - Type of Connection



Connection wizard

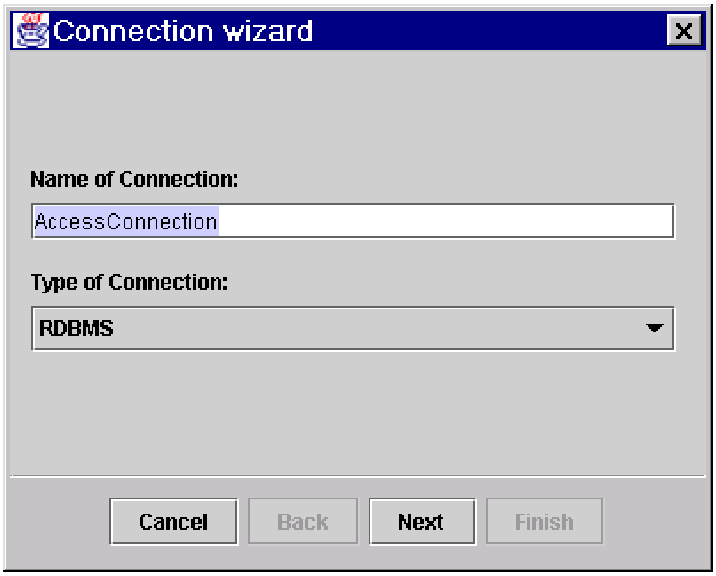
Name of Connection:

Type of Connection: RDBMS

4 Type AccessConnection for Name of Connection.

5 Select RDBMS as Type of Connection.

Figure 7-36. Microsoft Access Connection Wizard - Name of Connection



Name of Connection: AccessConnection

Type of Connection: RDBMS

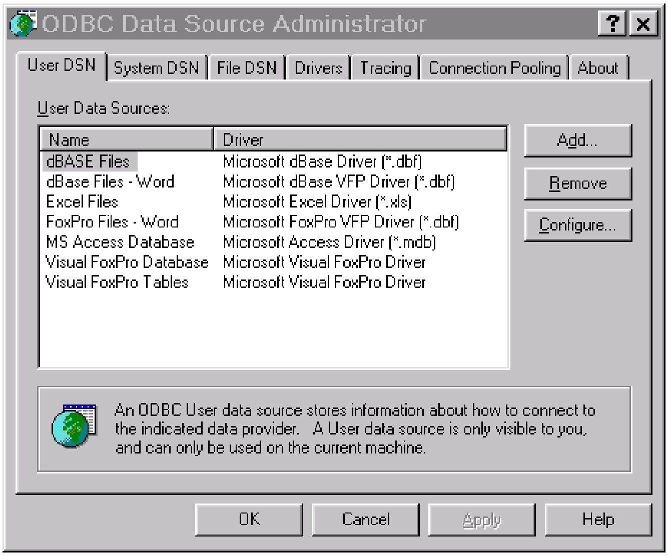
6 Click Next.

You must now create an ODBC datasource.

7 Select **Programs > Control Panel**. The Control Panel displays.

8 Select datasource (ODBC). The ODBC datasource Administrator window displays.

Figure 7-37. ODBC datasource Administrator



User Data Sources:

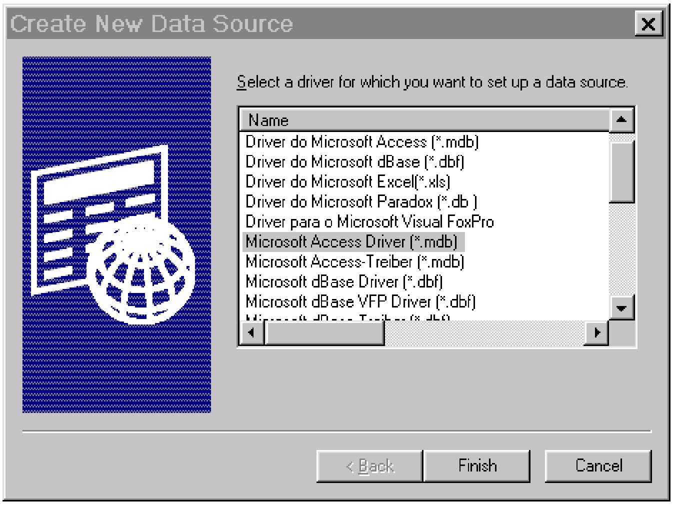
Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files - Word	Microsoft FoxPro VFP Driver (*.dbf)
MS Access Database	Microsoft Access Driver (*.mdb)
Visual FoxPro Database	Microsoft Visual FoxPro Driver
Visual FoxPro Tables	Microsoft Visual FoxPro Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

9 Click Add to add a new datasource. Make sure that the User DSN tab is selected. The Create New datasource window displays.

10 Select Microsoft Access Driver (*.mdb).

Figure 7-38. Create New datasource

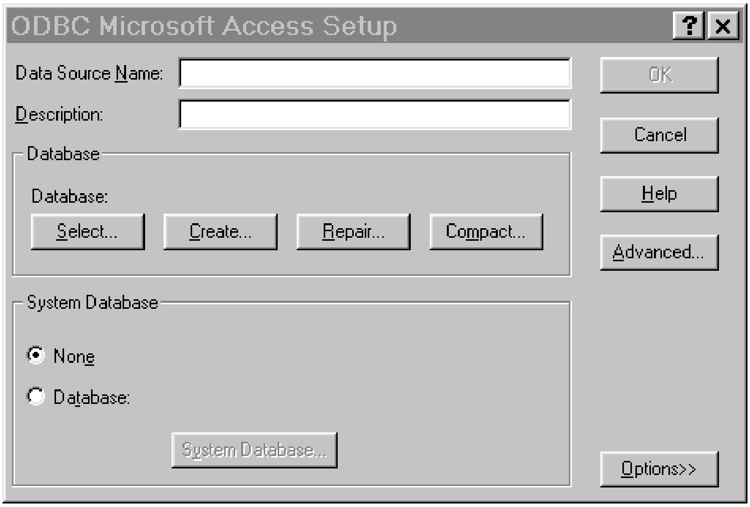


Select a driver for which you want to set up a data source.

Driver do Microsoft Access (*.mdb)
Driver do Microsoft dBase (*.dbf)
Driver do Microsoft Excel(*.xls)
Driver do Microsoft Paradox (*.db)
Driver para o Microsoft Visual FoxPro
Microsoft Access Driver (*.mdb)
Microsoft Access-Treiber (*.mdb)
Microsoft dBase Driver (*.dbf)
Microsoft dBase VFP Driver (*.dbf)

11 Click Finish. The ODBC Microsoft Access Setup dialog box displays.

Figure 7-39. ODBC Microsoft Access Setup



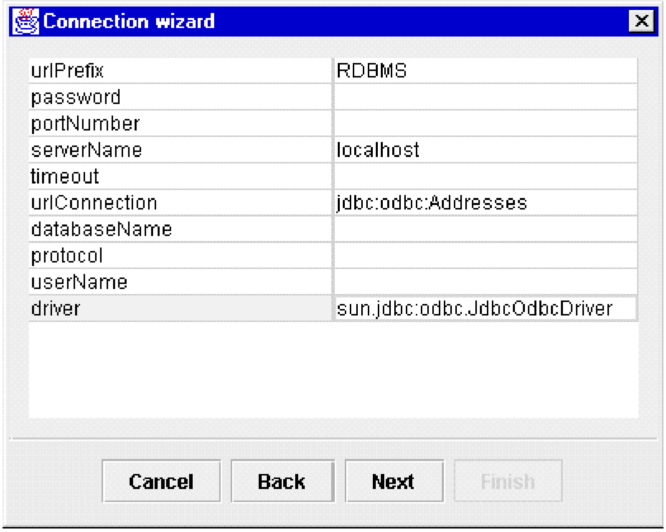
12 Type Addresses in the datasource Name field.

13 Click Select. The Select Database file requester displays.

14 Navigate to the location of the desired datasource using the Select Database file requester.

15 Click Ok. You have now created an ODBC Connection called Addresses and connected it an ODBC datasource.

Figure 7-40. Microsoft Access Connection Wizard 2

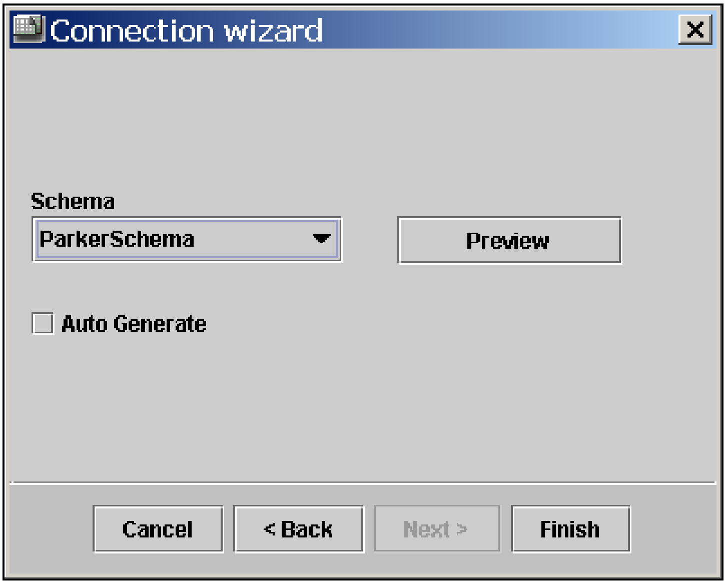


urlPrefix	RDBMS
password	
portNumber	
serverName	localhost
timeout	
urlConnection	jdbc:odbc:Addresses
databaseName	
protocol	
userName	
driver	sun.jdbc:odbc.JdbcOdbcDriver

16 Click Next.

17 You must now select a Source Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

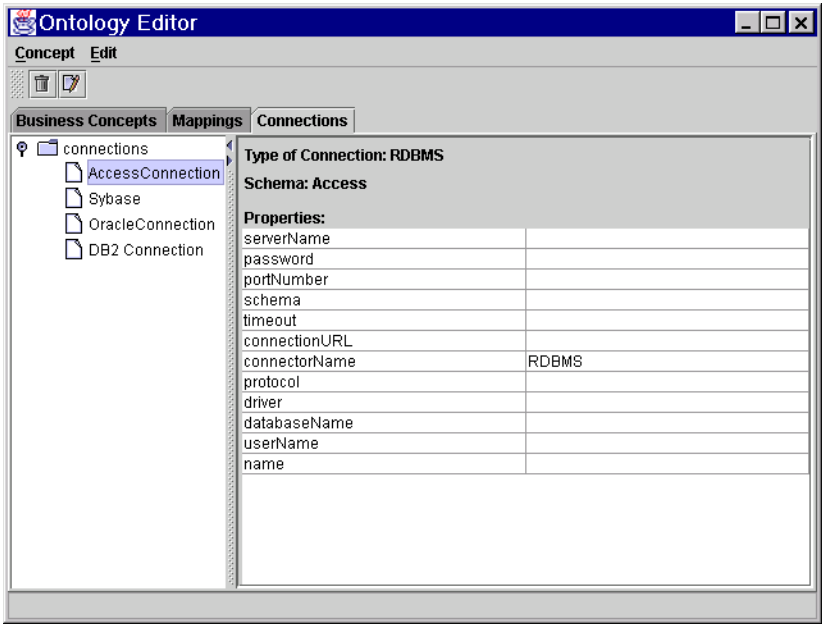
Figure 7-41. Microsoft Access Connection Wizard 3



To auto-generate a Schema, select the Auto Generate checkbox.

18 Click Finish. The Access Connection displays in the Ontology Editor.

Figure 7-42. Access Connection



Type of Connection: RDBMS

Schema: Access

Properties:

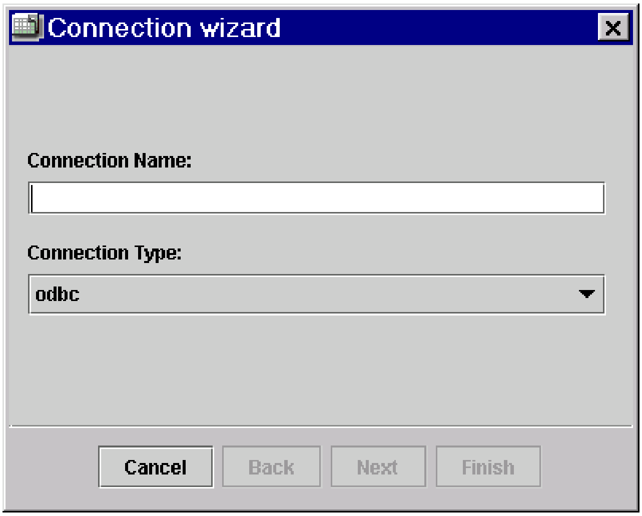
serverName	
password	
portNumber	
schema	
timeout	
connectionURL	
connectorName	RDBMS
protocol	
driver	
databaseName	
userName	
name	

Excel Connection (Windows-only)

1. Open the Ontology Editor, if it is not currently open.
2. Select the Connections tab.
3. Select **Concept > New > New Connection**. The Connection wizard displays.

4 Select ODBC as Connection Type.

Figure 7-43. Connection Wizard - ODBC - Excel



Connection wizard

Connection Name:

Connection Type: odbc

5 Type a name for the Connection.

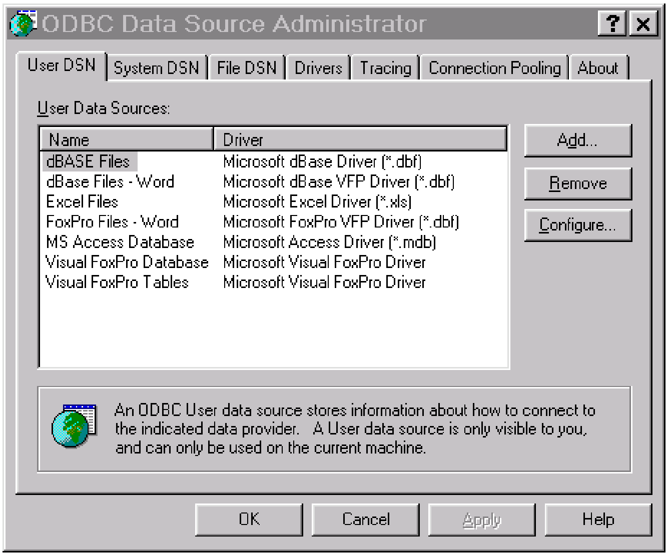
6 Click Next. The ODBC Connection Properties display. Type in the appropriate properties.

You must now create an ODBC datasource.

7 Select **Start > Settings > Control Panel**. The Control Panel displays.

8 Select datasource (ODBC). The ODBC datasource Administrator window displays.

Figure 7-44. ODBC datasource Administrator



User Data Sources:

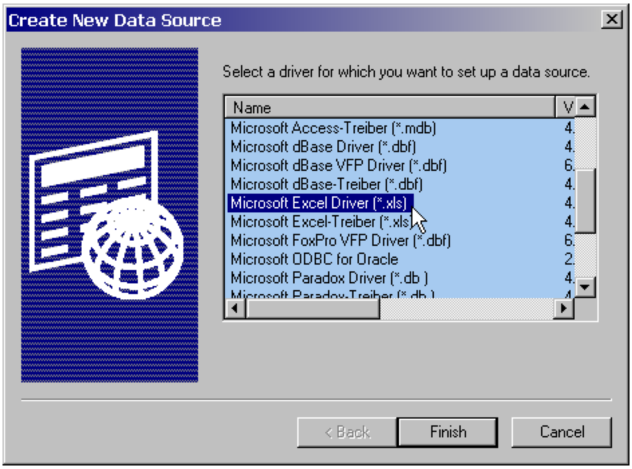
Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files - Word	Microsoft FoxPro VFP Driver (*.dbf)
MS Access Database	Microsoft Access Driver (*.mdb)
Visual FoxPro Database	Microsoft Visual FoxPro Driver
Visual FoxPro Tables	Microsoft Visual FoxPro Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

9 Click Add to add a new datasource. Make sure that the User DSN tab is selected. The Create New datasource window displays.

10 Select Microsoft Excel Driver (*.xls).

Figure 7-45. Create New ODBC datasource - Excel

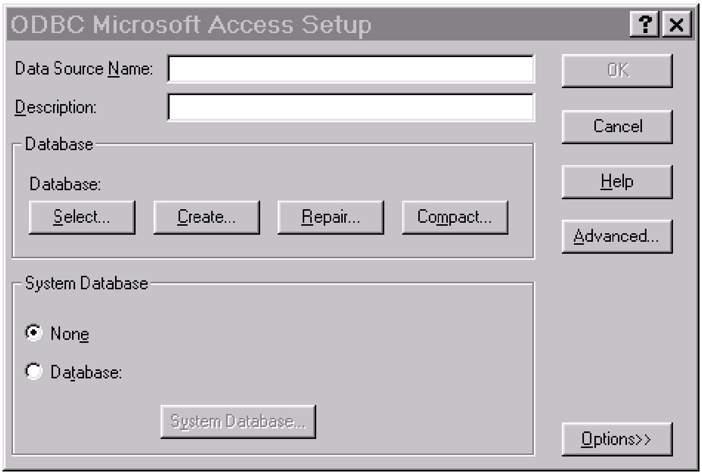


Select a driver for which you want to set up a data source.

Name	V
Microsoft Access-Treiber (*.mdb)	4.
Microsoft dBase Driver (*.dbf)	4.
Microsoft dBase VFP Driver (*.dbf)	6.
Microsoft dBase-Treiber (*.dbf)	4.
Microsoft Excel Driver (*.xls)	4.
Microsoft Excel-Treiber (*.xls)	4.
Microsoft FoxPro VFP Driver (*.dbf)	6.
Microsoft ODBC for Oracle	2.
Microsoft Paradox Driver (*.db)	4.
Microsoft Paradox-Treiber (*.db)	4.

11 Click Finish. The ODBC Microsoft Excel Setup dialog box displays.

Figure 7-46. ODBC Microsoft Excel Setup



Data Source Name:

Description:

Database:

● None

○ Database:

12 Type Accounts in the datasource Name field.

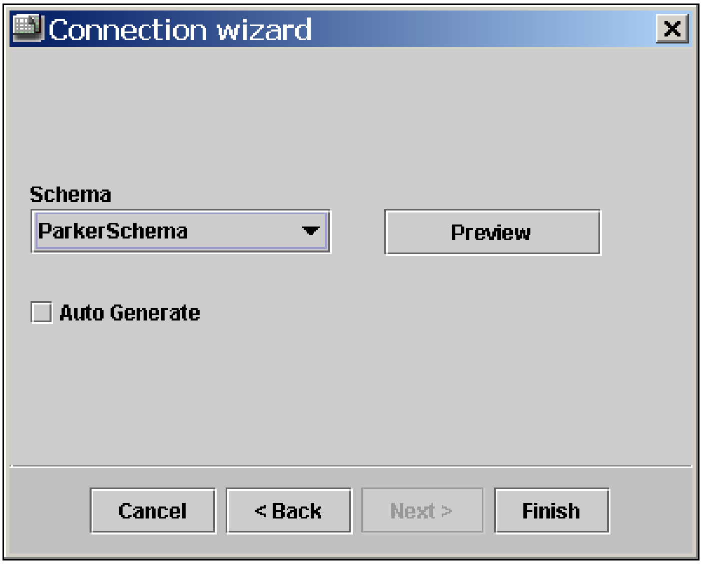
13 Select Select. The Select Database file requester displays.

14 Navigate to the location of the desired data base using the Select... file requester.

15 Click Ok. You have now created the an Accounts ODBC Connection and linked it with the Accounts Excel datasource.

16 You must now select a Source Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

Figure 7-47. Excel Connection Wizard 3



Connection wizard

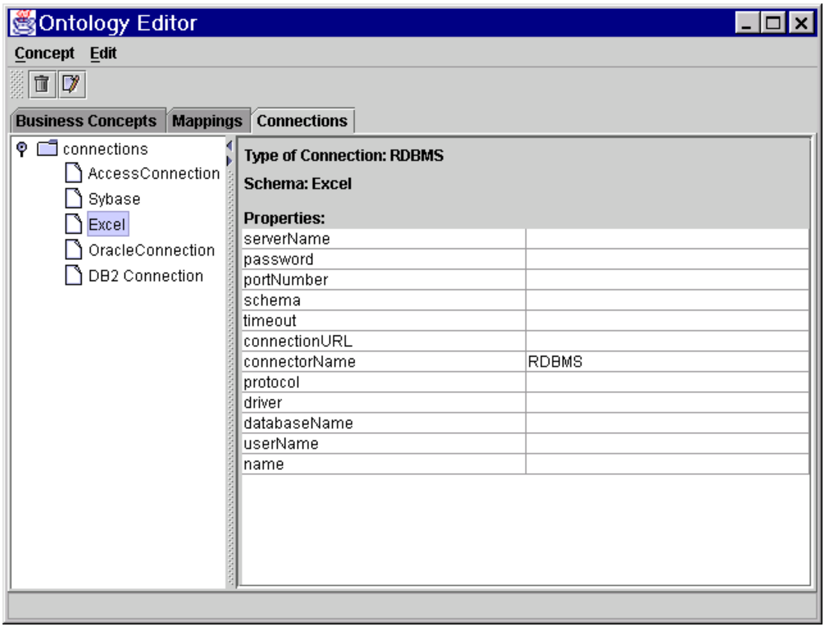
Schema: ParkerSchema

☐ Auto Generate

To auto-generate a Schema, select the Auto Generate checkbox.

Click Finish. The Excel Connection displays in the Ontology Editor.

Figure 7-48. Excel Connection



Type of Connection: RDBMS

Schema: Excel

Properties:

serverName	
password	
portNumber	
schema	
timeout	
connectionURL	
connectorName	RDBMS
protocol	
driver	
databaseName	
userName	
name	

Creating a DB2 Connection

DB2 Connection Wizard Properties

The DB2 Connection Wizard Properties are

Table 7-7. DB2 Connection Wizard Properties

Name	Description	Required
serverName	Identifies the DB2 server host using standard IP addressing.	Yes
password	Supplies the password credentials of a principal authorized to access the DB2	Yes
portNumber	Identifies the port used for communication with the DB2 server.	Yes
databaseName	Identifies the name of the DB2	Yes
URLPrefix	Specifies the DB2 records using a Connection String defined for a particular DB2, in this case, `jdbc:db2`.	Yes
timeout	Specifies how long in milliseconds to wait for a response from the DB2	No
driver	Identifies the Java class that contains a driver suitable for communication with the DB2. This class file must be accessible through the Java `CLASSPATH` environment setting.	Yes
userName	Supplies the username credentials of a principal authorized to access the DB2.	Yes
databaseSchema	Specifies to which Schema the connection should attach.	No

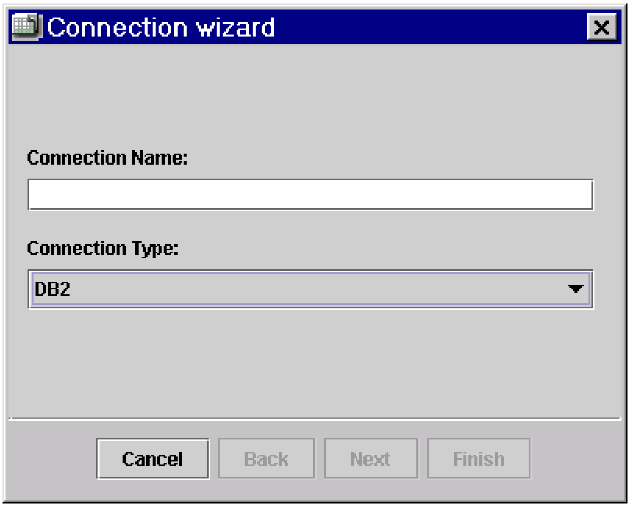
IBM DB2

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections Tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays.

Figure 7-49. IBM DB2 Connection Wizard 1



Connection wizard

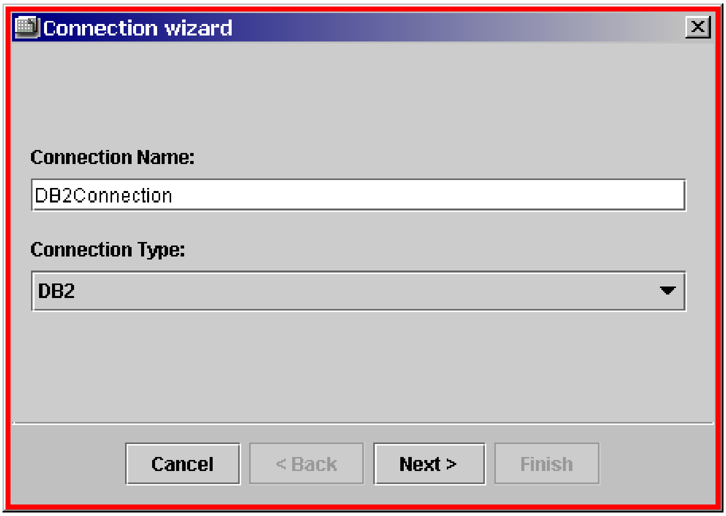
Connection Name:

Connection Type: DB2

4 Type the name Connection Name.

5 Select DB2 as type of connection.

Figure 7-50. IBM DB2 Connection Wizard 2

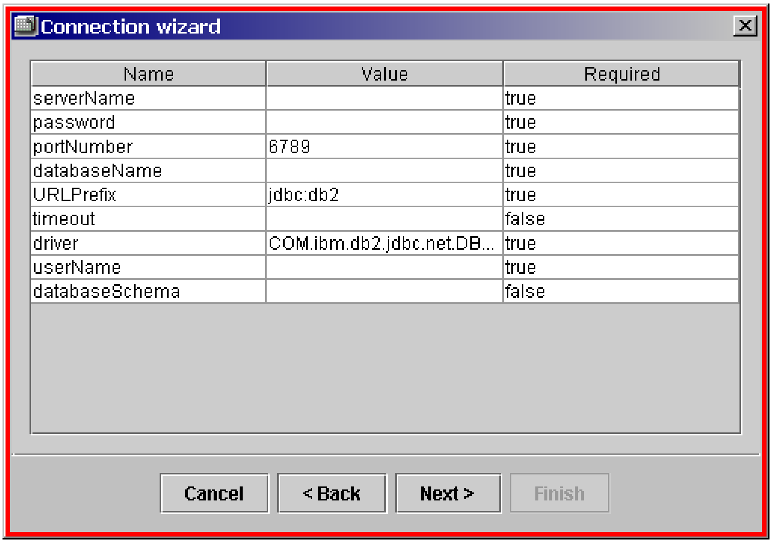


Connection Name: DB2Connection

Connection Type: DB2

6 Click Next. The DB2 default properties display in the Connection wizard.

Figure 7-51. IBM DB2 Connection Wizard - Connection Properties



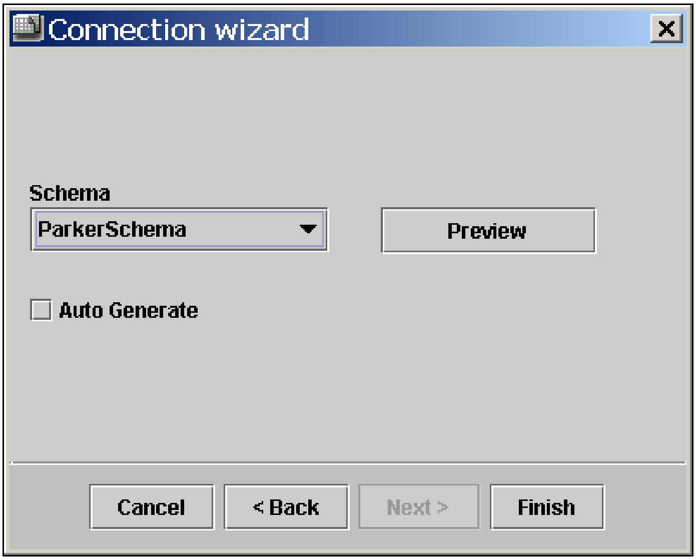
Name	Value	Required
serverName		true
password		true
portNumber	6789	true
databaseName		true
URLPrefix	jdbc:db2	true
timeout		false
driver	COM.ibm.db2.jdbc.net.DB...	true
userName		true
databaseSchema		false

7 Type the required properties

8 Click Next.

9 You must select a Source Schema. You can select Preview to preview your selection in the Schema Preview Dialog.

Figure 7-52. IBM DB2 Connection Wizard 4



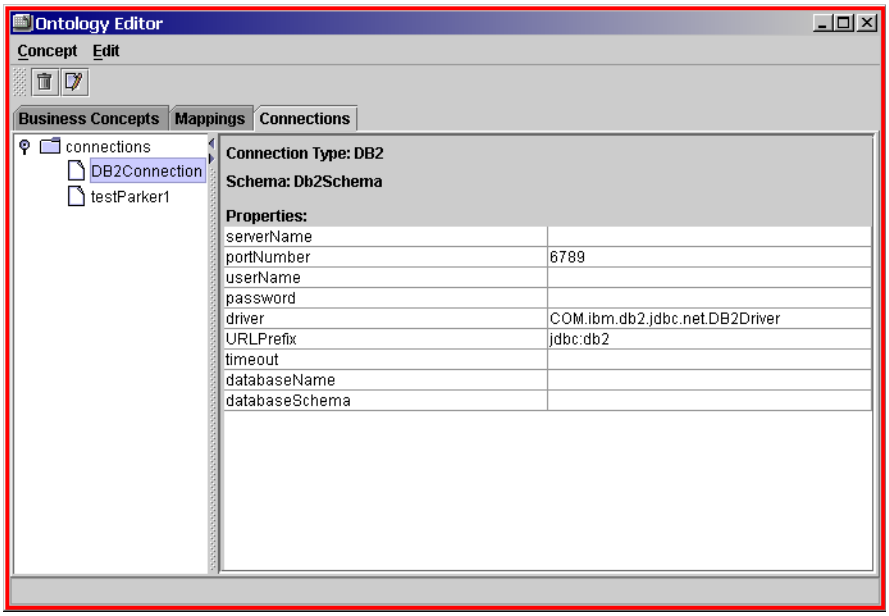
Schema: ParkerSchema

☐ Auto Generate

To auto-generate a Schema, select the Auto Generate checkbox.

Click Finish. The IBM DB2 Connection displays in the Ontology Editor.

Figure 7-53. IBM DB2 Connection Completed



Connection Type: DB2

Schema: Db2Schema

Properties:

serverName	
portNumber	6789
userName	
password	
driver	COM.ibm.db2.jdbc.net.DB2Driver
URLPrefix	jdbc:db2
timeout	
databaseName	
databaseSchema	

Creating an Oracle Connection

Oracle Connection Wizard Properties

The Oracle Connection Wizard Access Properties are

Table 7-8. Oracle Connection Wizard Properties

Name	Description	Required
URLPrefix	Specifies the Oracle records using a Connection String defined for a particular connection. The Connection String for Oracle is `jdbc:oracle`.	Yes
password	Supplies the password credentials of a principal authorized to access the Oracle Connection.	Yes
portNumber	Identifies the port used for communication with the Oracle server. For example, the default `portNumber` for an Oracle connector is 1521.	Yes
SID	serverID:oracle	Yes
serverName	Identifies the Oracle server host using standard IP addressing.	Yes
timeout	Specifies how long in milliseconds to wait for a response from the Oracle.	No
protocol	Identifies the protocol used for communication with the database. For example, the default protocol for the Oracle connector is `oracle:thin`.	Yes
databaseSchema	Specifies to which Schema the connection should attach.	No
userName	Supplies the username credentials of a principal authorized to access the database	Yes
driver	Identifies the Java class that contains a driver suitable for communication with the datbase. This class file must be accessible through the Java `CLASSPATH` environment setting. For example, the default driver for the Oracle connection is `oracle.jdbc.driver.OracleDriver`.	No

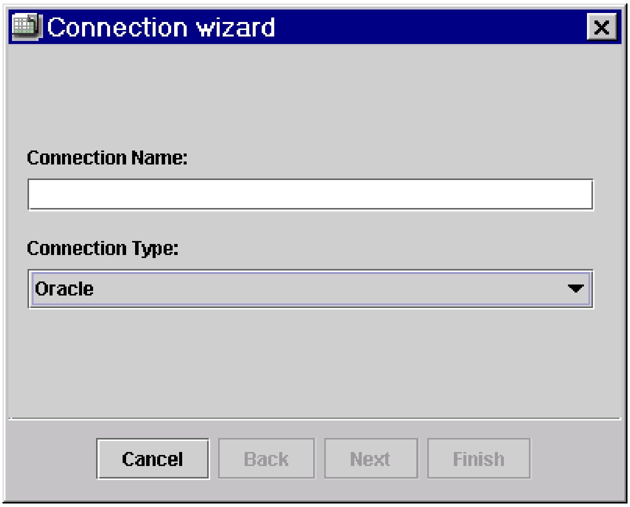
Oracle Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select Oracle from the Connection Type pull-down menu.

Figure 7-54. Connection Wizard - Oracle

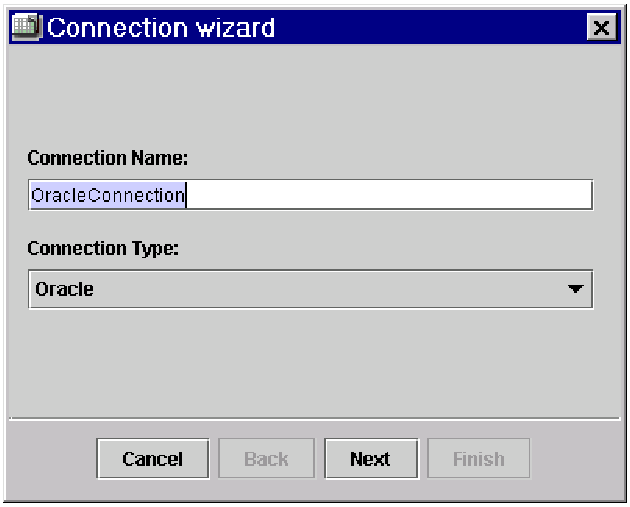


Connection Name:

Connection Type: Oracle

4 Type a name for the Oracle Connection.

Figure 7-55. Connection Wizard - Oracle - Connection Name

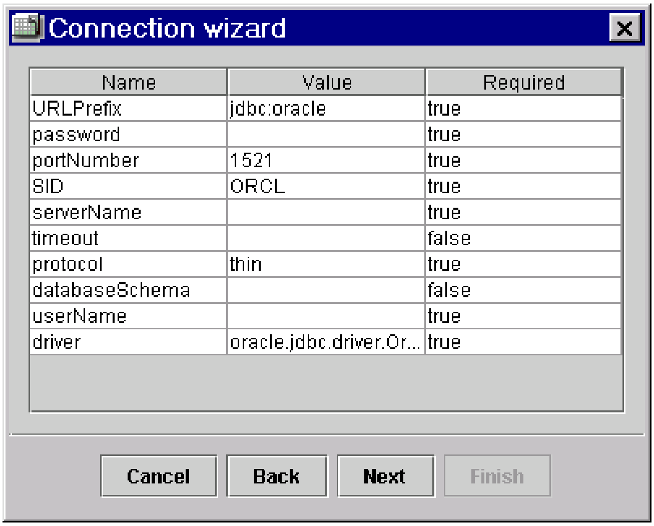


Connection Name: OracleConnection

Connection Type: Oracle

5 Click Next. The Connection Wizard now displays the Oracle connection properties.

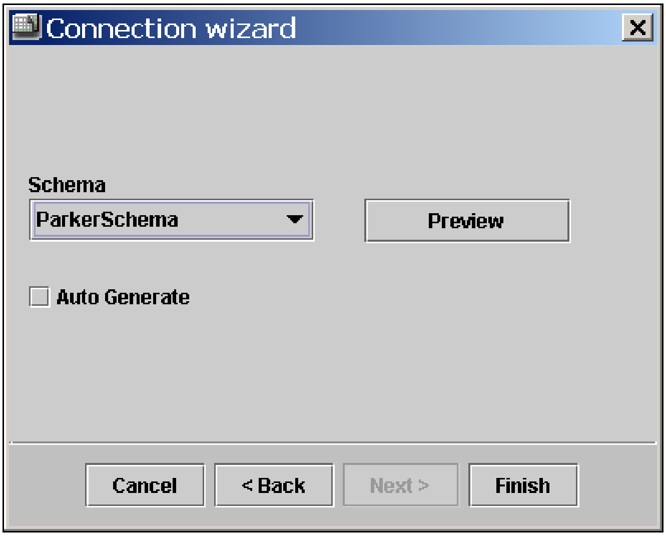
Figure 7-56. Connection Wizard - Oracle - Connection Properties



Name	Value	Required
URLPrefix	jdbc:oracle	true
password		true
portNumber	1521	true
SID	ORCL	true
serverName		true
timeout		false
protocol	thin	true
databaseSchema		false
userName		true
driver	oracle.jdbc.driver.Or...	true

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-57. Connection Wizard - Oracle - Schema Selector



Connection wizard

Schema: ParkerSchema

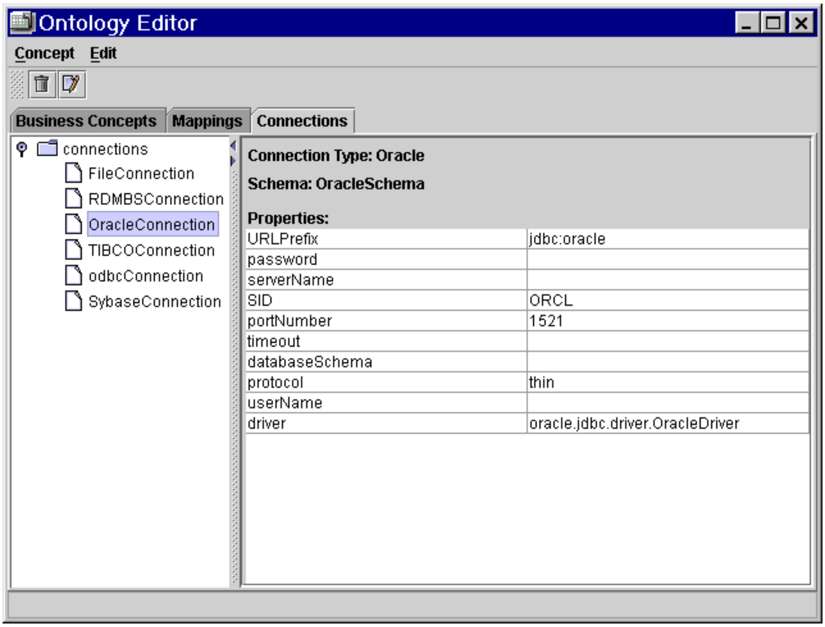
□ Auto Generate

7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connection Wizard. You return to the Connection tab of the Ontology Editor.

Figure 7-58. Oracle Connection



Connection Type: Oracle

Schema: OracleSchema

Properties:

URLPrefix	jdbc:oracle
password	
serverName	
SID	ORCL
portNumber	1521
timeout	
databaseSchema	
protocol	thin
userName	
driver	oracle.jdbc.driver.OracleDriver

Creating a JMS Connection

JMS Connection Wizard Properties

The JMS Connection Wizard Properties are

Table 7-9. JMS Connection Wizard Access Properties

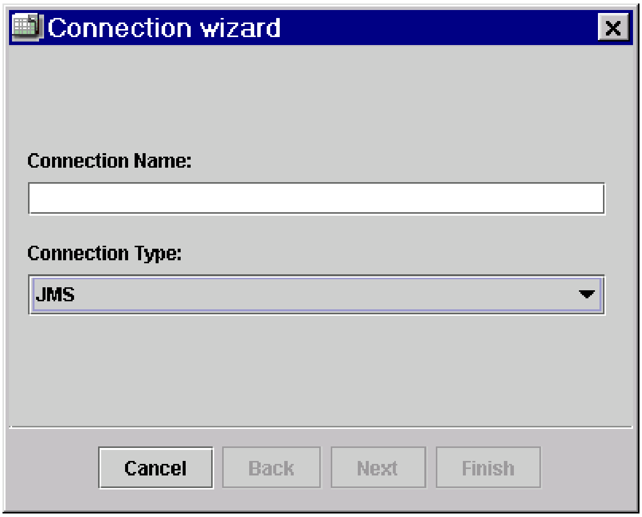
Name	Description	Required
security Credentials	NULL	No
schema	Specifies the Schema name.	Yes
connectorName	Contains a short string that prefixes the connector URL and helps to identify the type of resource and protocol required.	Yes
acknowledge Mode	Specifies message acknowledge method. The following methods are available auto, client, and duplicate. Auto - the system automatically send acknowledgement messages after each message is received. Client - the programmer must organize the acknowledgemessages. Duplicates - message delivery is lazily acknowledged.	No
providerURL	`TB://127.0.0.1:7001`	Yes
securityPrincipal		No
connectionType	JMS provides two communication paradigms. 1. Qqeue-based communication is point-to-point - a message is delivered to exactly one receiver. 2. Topic-based communication - a message is published and received by subscribers.	Yes
default Destination	`weblogic.examples.jms.examplequeue`	No
connection Factory	Specifies the name under which the factory for JMS connection is registered in JNDI. The name is chosen by the user, but the JMS server administrator uses the same name to create a new factory.	Yes
name	Specifies the connection name.	Yes
jndiFactory	Specifies the Java class name that is used to access JNDI. JMS requires that connection factories be obtained from a JNDI (Java Naming and Directory Service) service.	Yes
transacted	Specifies messages sent in transaction.	No

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select JMS from the Connection Type pull-down menu.

Figure 7-59. Connection Wizard - JMS



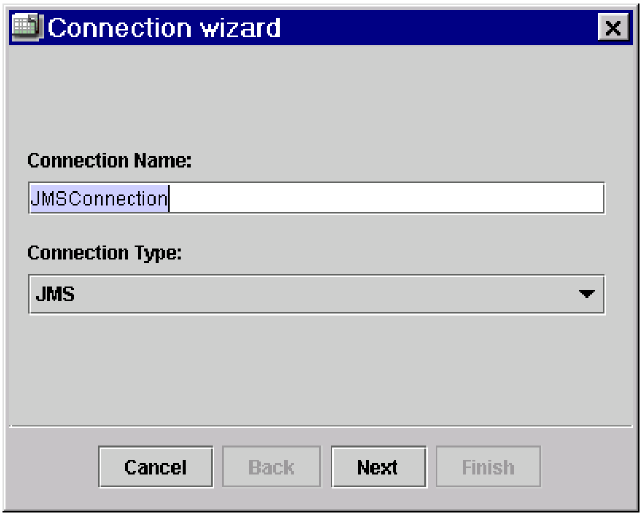
Connection wizard

Connection Name:

Connection Type: JMS

4 Type a name for the JMS Connection.

Figure 7-60. Connection Wizard - JMS - Connection Name

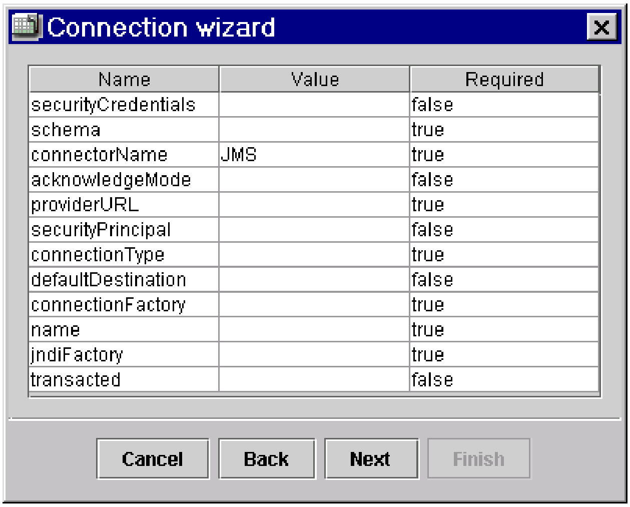


Connection Name: JMSConnection

Connection Type: JMS

5 Click Next. The Connection Wizard now displays the JMS connection properties.

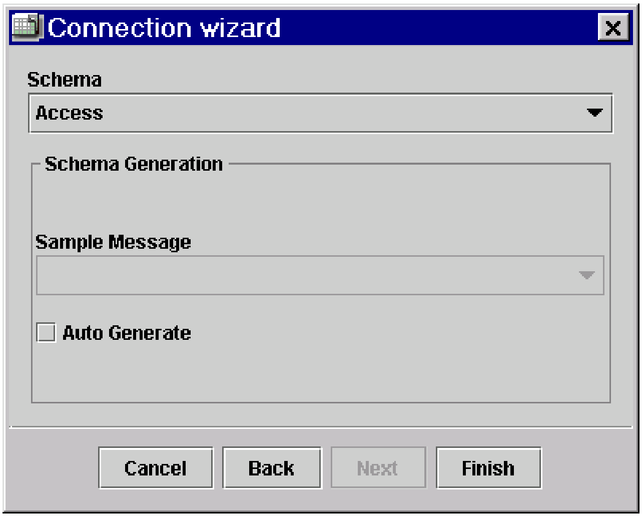
Figure 7-61. Connection Wizard - JMS - Connection Properties



Name	Value	Required
securityCredentials		false
schema		true
connectorName	JMS	true
acknowledgeMode		false
providerURL		true
securityPrincipal		false
connectionType		true
defaultDestination		false
connectionFactory		true
name		true
jndiFactory		true
transacted		false

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-62. Connection Wizard - JMS - Schema Selector


Connection wizard

Schema: Access

Schema Generation

Sample Message

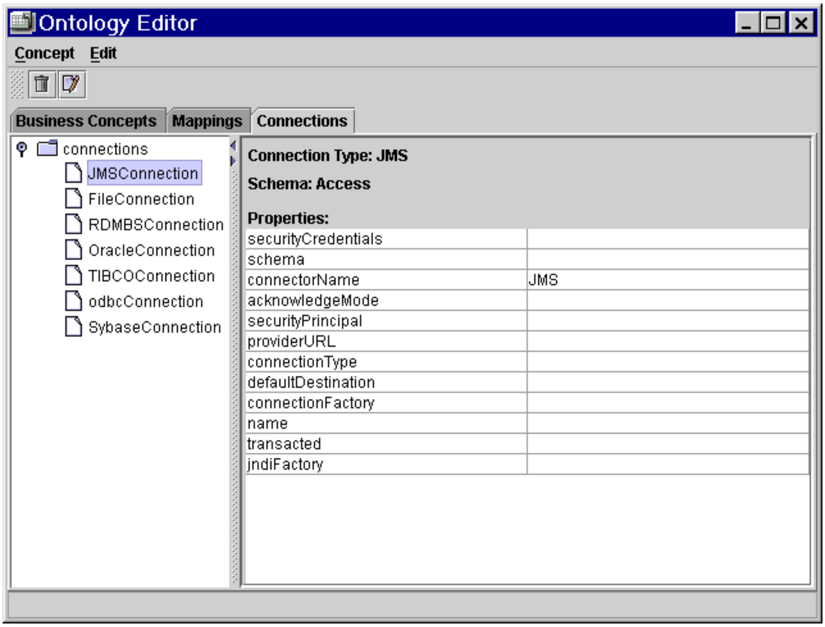
□ Auto Generate

7 Select the appropriate Schema.

To auto-generate a Schema, select the Auto Generate checkbox.

8 Click Finish to exit the Connection Wizard. You return to the Connection tab of the Ontology Editor.

Figure 7-63. JMS - Connection



Connection Type: JMS

Schema: Access

Properties:

securityCredentials	
schema	
connectorName	JMS
acknowledgeMode	
securityPrincipal	
providerURL	
connectionType	
defaultDestination	
connectionFactory	
name	
transacted	
jndiFactory	

Creating a WWW Connection

WWW Connection Wizard Properties

The WWW Connection Wizard Properties are

Table 7-10. WWW Connection Wizard Access Properties

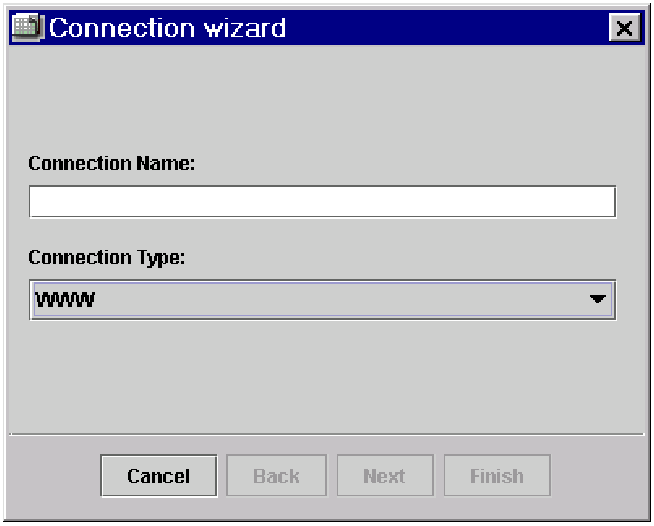
Name	Description	Required
dataFormat	Specifies the data format, either XML or HTML.	Yes
schema	Specifies the Schema name.	Yes
connectorName	Contains a short string that prefixes the connector URL and helps to identify the type of resource and protocol required.	Yes
url	Specifies the url.	Yes
name	Specifies connection name.	Yes
refreshTimeout	Specifies the time to refresh, in milliseconds.	Yes

1 Open the Ontology Editor, if it is not currently open.

2 Select the Connections tab.

3 Select **Concept > New > New Connection**. The Connection wizard displays. Select WWW from the Connection Type pull-down menu.

Figure 7-64. Connection Wizard - WWW



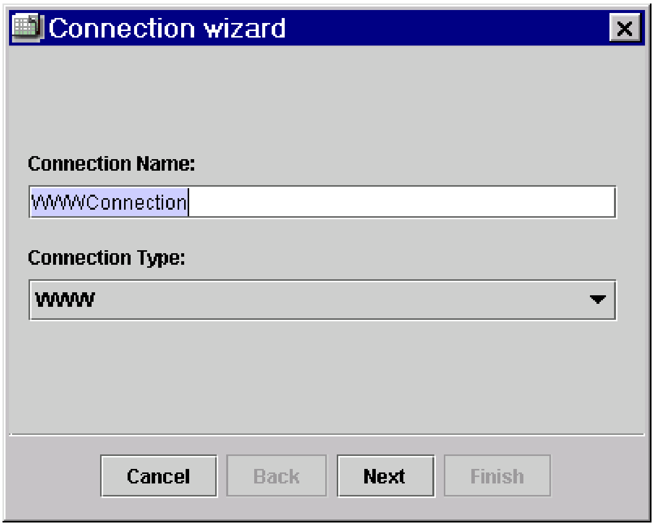
Connection wizard

Connection Name:

Connection Type: WWW

4 Type a name for the WWW Connection.

Figure 7-65. Connection Wizard - WWW - Connection Name

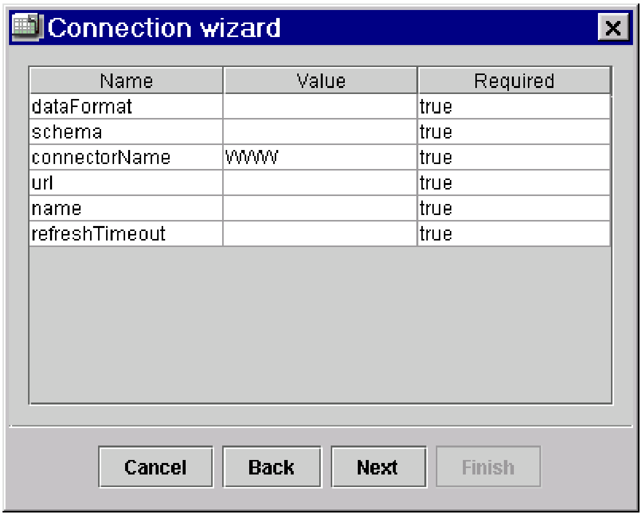


Connection Name: WWWConnection

Connection Type: WWW

5 Click Next. The Connection Wizard now displays the WWW connection properties.

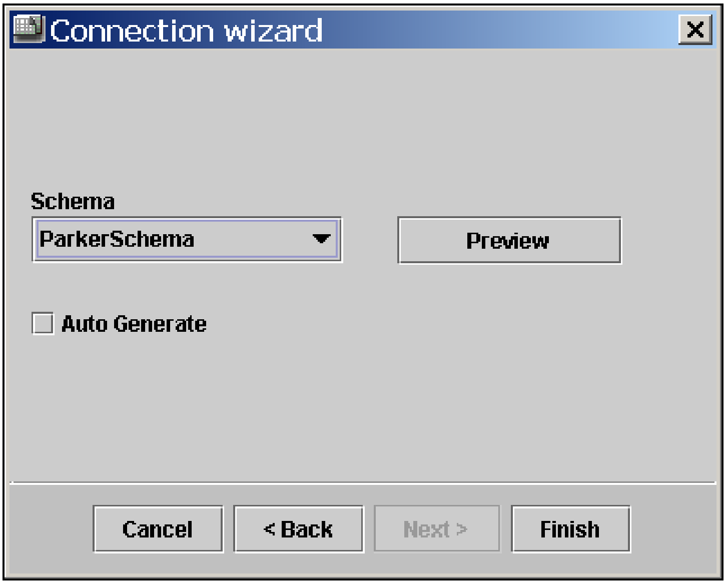
Figure 7-66. Connection Wizard - WWW - Connection Properties



Name	Value	Required
dataFormat		true
schema		true
connectorName	WWW	true
url		true
name		true
refreshTimeout		true

6 Fill in the Connection Properties as required. Click Next to continue. The Connection Wizard Schema Selector displays.

Figure 7-67. Connection Wizard - WWW - Schema Selector



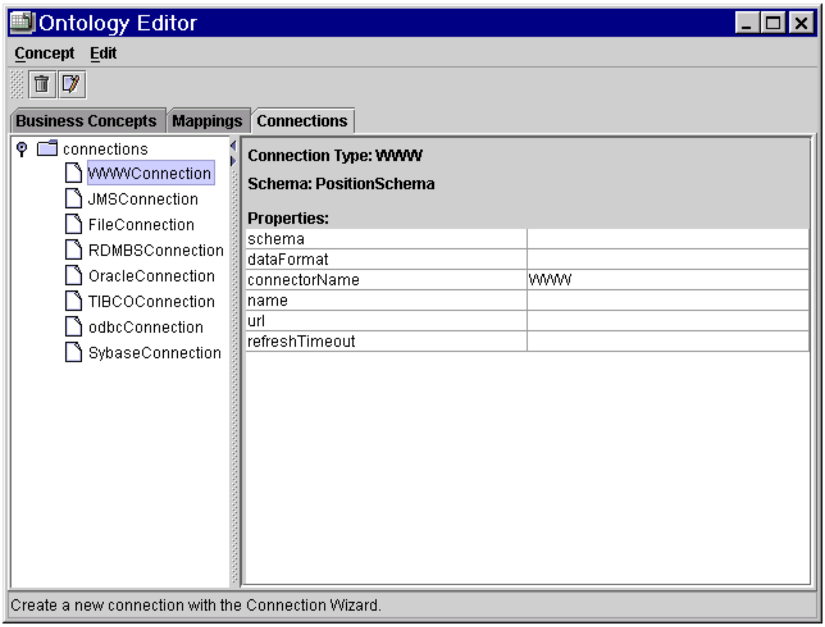
Schema: ParkerSchema

□ Auto Generate

7 Select the appropriate Schema. You can select Preview to preview your selection in the Schema Preview Dialog

Click Finish to exit the Connection Wizard. You return to the Connection tab of the Ontology Editor.

Figure 7-68. WWW - Connection



Connection Type: WWW

Schema: PositionSchema

Properties:

schema	
dataFormat	
connectorName	WWW
name	
url	
refreshTimeout	

Create a new connection with the Connection Wizard.

Editing a Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select Connections tab.

3 Select the desired connection.

4 Option-click and select **Edit...** The Connection Wizard displays. You can move through the wizard by clicking Next and make changes, additions, or deletions to the Connection. When you finish editing the Connection, click Finish. The changes you made are applied to the Connection.

Deleting a Connection

1 Open the Ontology Editor, if it is not currently open.

2 Select Connections tab.

3 Select the desired connection.

4 Option-click and select **Delete**. The Confirm Deletion dialog box displays.

5 Click Yes to delete Connection or click No to exit. If you clicked Yes, the Connection deletes and does not display in the Ontology Editor.

Understanding Mappings and Joins

Mappings and Joins are a key activity that links external connections with internal Knowledge Broker Business Concepts.

In the Connections Wizard you associated external data with Schemas that "channel" the data as meta-information. Schemas store their "child" data types as elements.

In the Business Concepts phase, you or Business Users (BUs) create a hierarchy of internal Business Concepts with child elements that model your business domain or problem.

During Mapping, you create links, or Mappings, between the Business Concept elements and the external Connection Schema elements. These Mappings are one-to-one. To create one-to-many or many-to-one relationships, you use a sub-feature of Mappings called Joins.

The Mapping Wizard enables you to create heterogeneous joins, that is, to federate across disparate and distributed datasources in the physical world, extract those data elements that are relevant to your business model, and tie them together as Business Concepts. Once aggregated like this, these Business Concepts can be manipulated by Business Users as single objects and they can remain unaware of their distributed, heterogeneous nature.

This is a powerful data abstraction and manipulation function of Knowledge Broker and allows Deployment Engineers to concentrate on data access and network architecture issues while freeing Business Users to concentrate on the business solution. Effectively decoupled, IT development and business development on a single project can proceed largely independent from each other.

Creating a Mapping

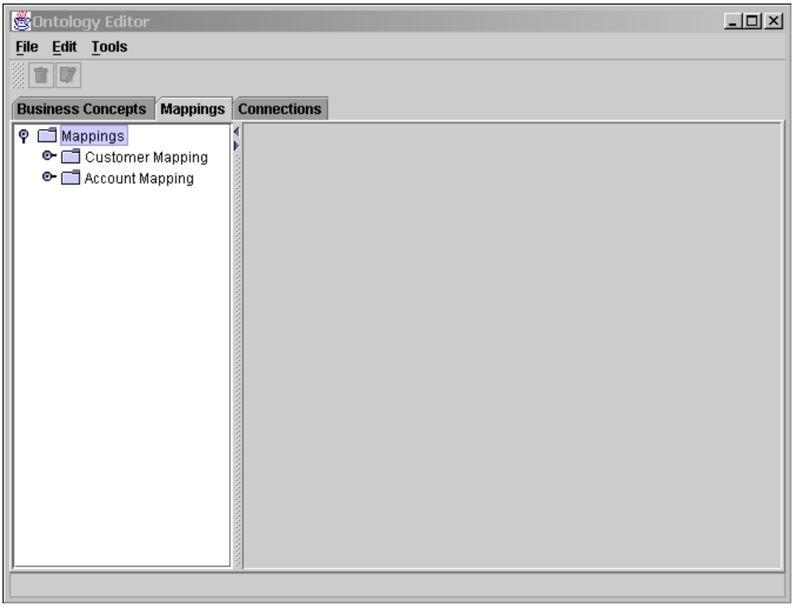
Every Business Concept and its child elements are represented in the Mappings tab of the Ontology Editor with “Mapping” appended to Business Concept name. You can click to expand the Business Concept Mappings to display the component elements. To create a Mapping, you must first created at least one Connection to external data and one Business Concept.

The Mapping Wizard is your interface for manipulating Mappings. The goal is to associate the Elements on the left-hand-side (that is, specific to Business Concepts) with Source Elements on the right-hand-side (that is, external data).

Because of Knowledge Broker’s new support for Structured Business Concepts (see *Flat and Structured Business Concepts on page 98*), if you create an SBC or use a pre-defined schema, you will not be able to use the Mapping Wizard to create Mappings and Joins for this Concept. Instead, you must enter a user-generated custom mapping (see *Creating User-Defined Queries on page 238*). For the Mapping Wizard to be enabled, you must create a flat Business Concept by specifying the business concept elements directly in the Concept Wizard.

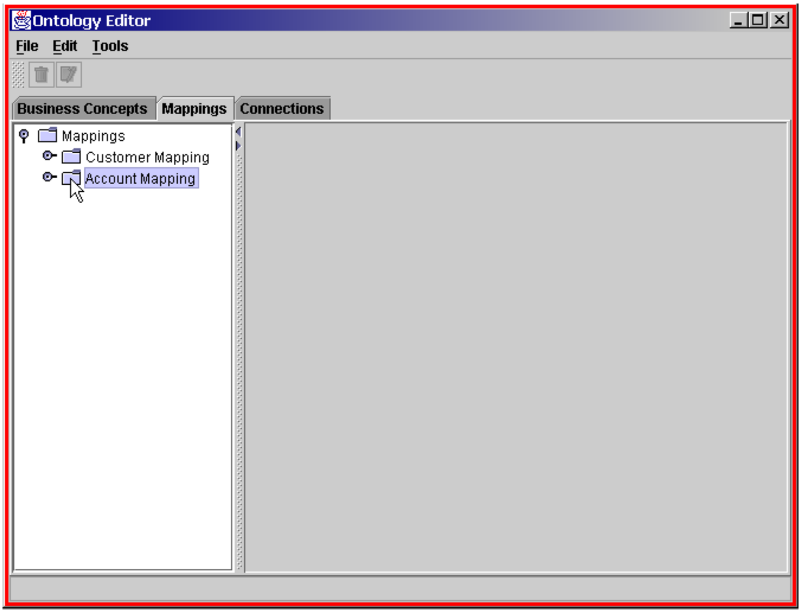
1 Open the Ontology Editor and click the Mappings tab. The left-hand panel displays the Mapping Elements for each Business Concept.

Figure 7-69. Ontology Editor - Mappings Active



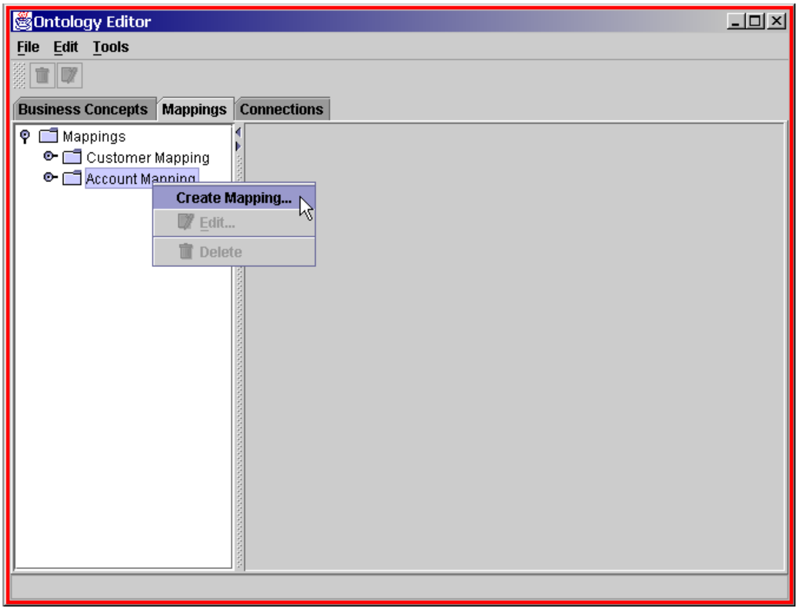
2 Select a Business Concept you wish to Map. You can do this by either clicking on one of the top-level Business Concept Mapping objects, or by using the cursor keys to cycle through them. The selected Business Concept Mapping object highlights.

Figure 7-70. Ontology Editor - Business Concept Mapping Selected



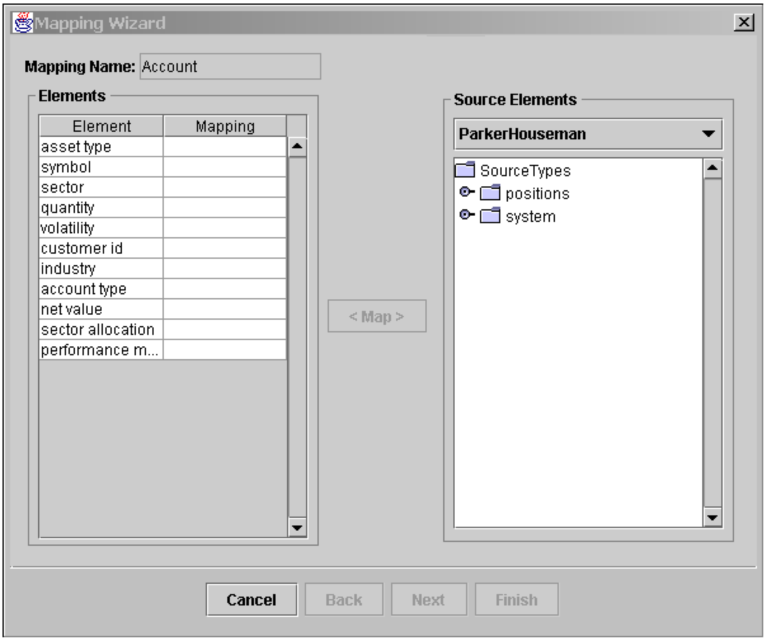
3 Option-click the selected Business Concept Mapping. The Create Mapping... alone highlights (the Edit... and Delete options are unavailable because you have not created any Business Concept Mapping here yet).

Figure 7-71. Ontology Editor - Create Mapping... Selected



4 When you release the mouse button, the Mapping Wizard Displays. The list of Elements and Source Elements will vary according to those Connections and Business Concepts defined in the Ontology Editor.

Figure 7-72. Mapping Wizard



Mapping Wizard

Mapping Name: Account

Elements

Element	Mapping
asset type	
symbol	
sector	
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

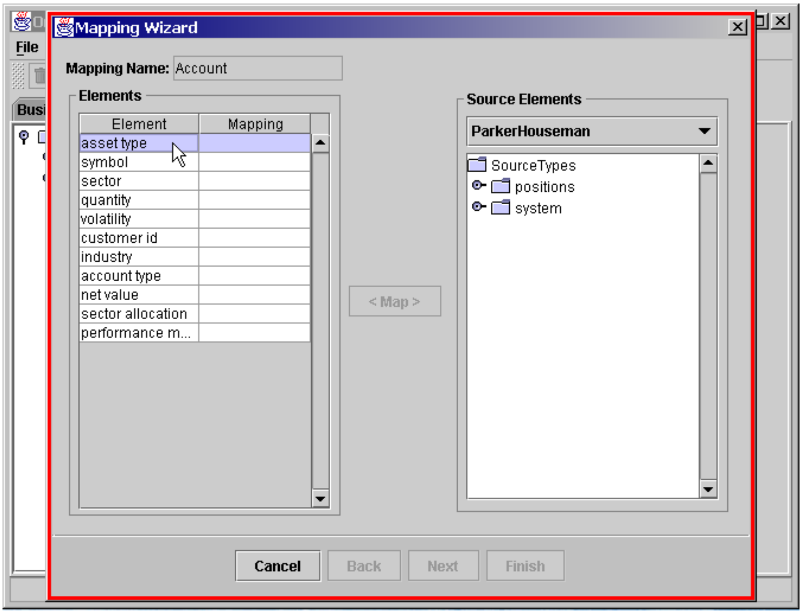
Source Elements

ParkerHouseman

- SourceTypes
  - positions
  - system

5 Click one of the left-hand Elements for which you wish you create a Mapping. You can use the cursor keys to cycle through the Elements.

Figure 7-73. Mapping Wizard - Element Selected



Mapping Name: Account

Elements

Element	Mapping
asset type	
symbol	
sector	
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

Source Elements

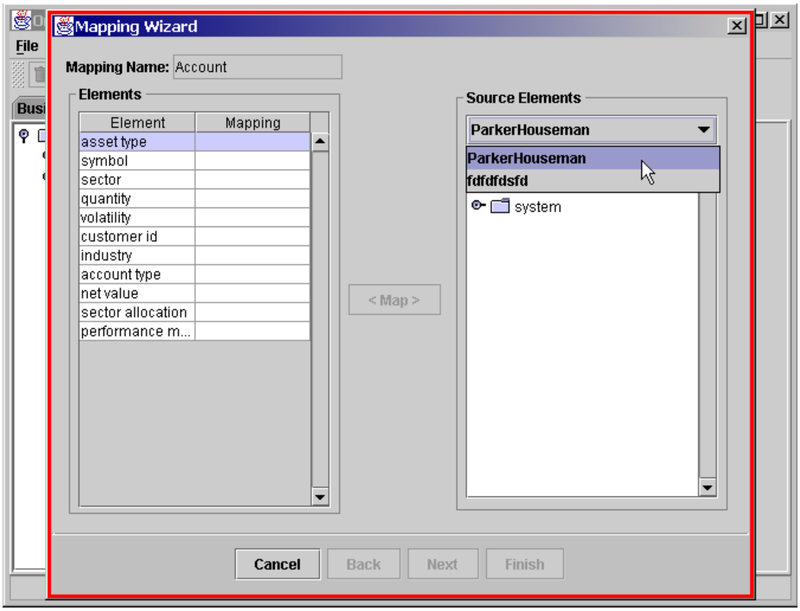
ParkerHouseman

- SourceTypes
  - positions
  - system

6 You must now select from which Connection you would like to Map. Without deselecting the left-hand Business Concept Element, click the drop-down menu below

"Source Elements" within the right-hand panel. A list of available Connections displays. Highlight your desired Connection.

Figure 7-74. Mapping Wizard - Selecting Connection



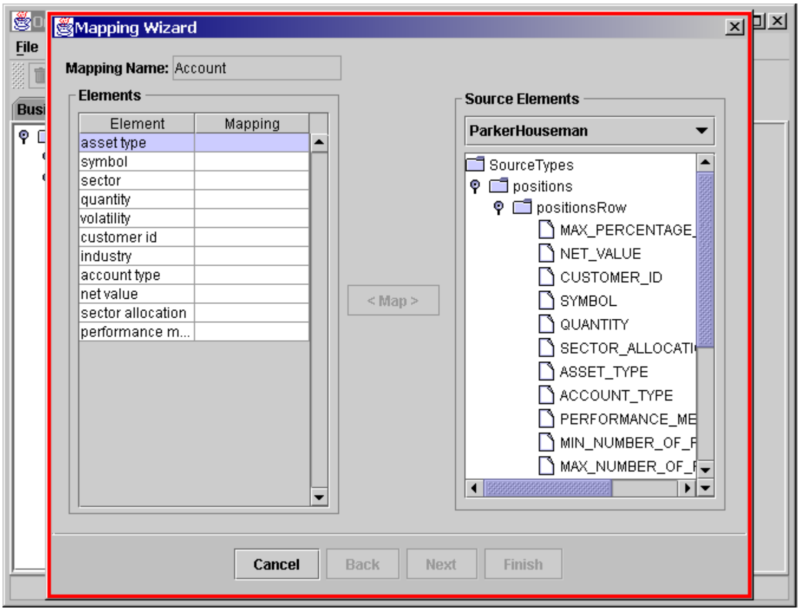
Mapping Name: Account

Element	Mapping
asset type	
symbol	
sector	
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

Source Elements: ParkerHouseman

7 After releasing the mouse button, your selected Connection displays its Source types. The exact display is a function of the selected Connection and its associated Schema. You should drill down the Element hierarchy until the individual Elements display.

Figure 7-75. Mapping Wizard Source Elements Displayed



Mapping Name: Account

Elements

Element	Mapping
asset type	
symbol	
sector	
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

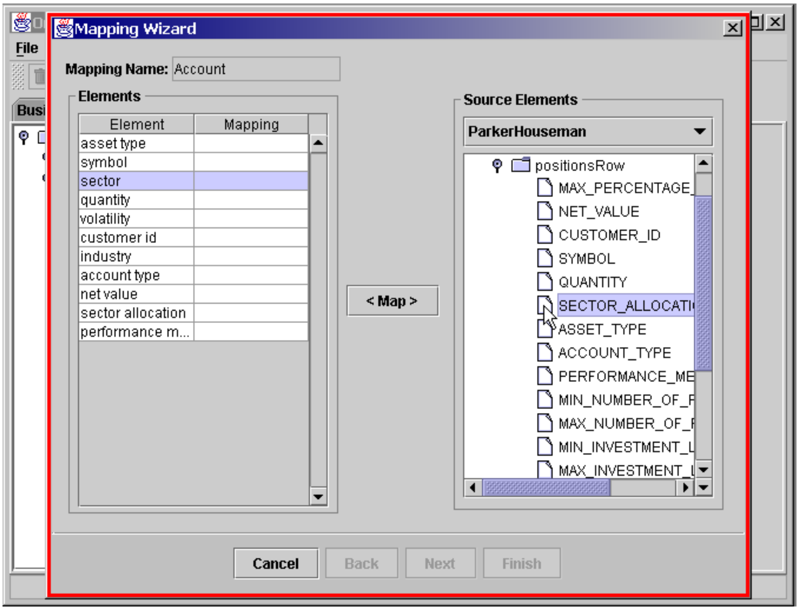
Source Elements

ParkerHouseman

- SourceTypes
  - positions
    - positionsRow
      - MAX_PERCENTAGE_
      - NET_VALUE
      - CUSTOMER_ID
      - SYMBOL
      - QUANTITY
      - SECTOR_ALLOCATI
      - ASSET_TYPE
      - ACCOUNT_TYPE
      - PERFORMANCE_ME
      - MIN_NUMBER_OF_F
      - MAX_NUMBER_OF_F

8 Now select those element or elements you wish to map from external datasource Elements to your internal Business Concept Elements. For example, select Element “sector” and Source Element SECTOR_ALLOCATION.

Figure 7-76. Mapping Wizard - Both Elements Selected



Mapping Name: Account

Elements

Element	Mapping
asset type	
symbol	
sector	
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

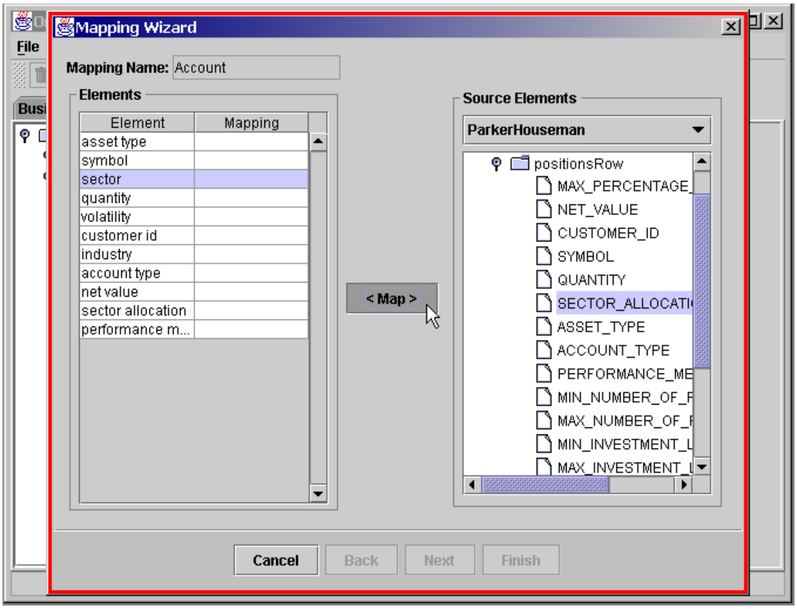
Source Elements

ParkerHouseman

- positionsRow
  - MAX_PERCENTAGE_
  - NET_VALUE
  - CUSTOMER_ID
  - SYMBOL
  - QUANTITY
  - SECTOR_ALLOCATI
  - ASSET_TYPE
  - ACCOUNT_TYPE
  - PERFORMANCE_ME
  - MIN_NUMBER_OF_F
  - MAX_NUMBER_OF_F
  - MIN_INVESTMENT_L
  - MAX_INVESTMENT_L

9 To Map these two selected Elements (and hence establish a one-to-one relationship), click the <Map> button.

Figure 7-77. Mapping Wizard - Map Button Selected



Mapping Wizard

Mapping Name: Account

Elements

Element	Mapping
asset type	
symbol	
sector	
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

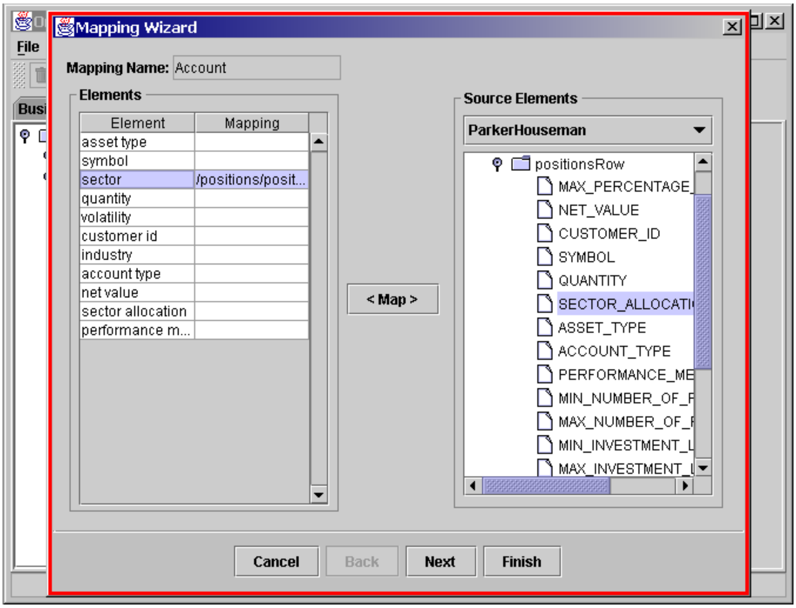
Source Elements

ParkerHouseman

positionsRow
MAX_PERCENTAGE_
NET_VALUE
CUSTOMER_ID
SYMBOL
QUANTITY
SECTOR_ALLOCATI
ASSET_TYPE
ACCOUNT_TYPE
PERFORMANCE_ME
MIN_NUMBER_OF_F
MAX_NUMBER_OF_F
MIN_INVESTMENT_L
MAX_INVESTMENT_L

10 The display now updates to reflect your Mapping. The "Mapping" column in the left-hand-side Elements panel now contains a hierarchical-style XPath fragment that describes the nature of the Mapping relation for the example Element.

Figure 7-78. Mapping Wizard - Mapping Accomplished

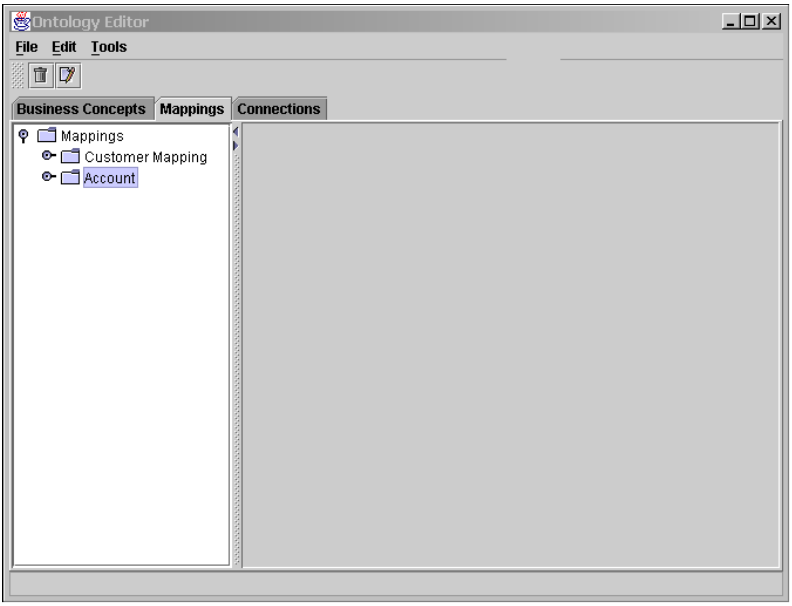


Mapping Name: Account

Element	Mapping
asset type	
symbol	
sector	/positions/posit...
quantity	
volatility	
customer id	
industry	
account type	
net value	
sector allocation	
performance m...	

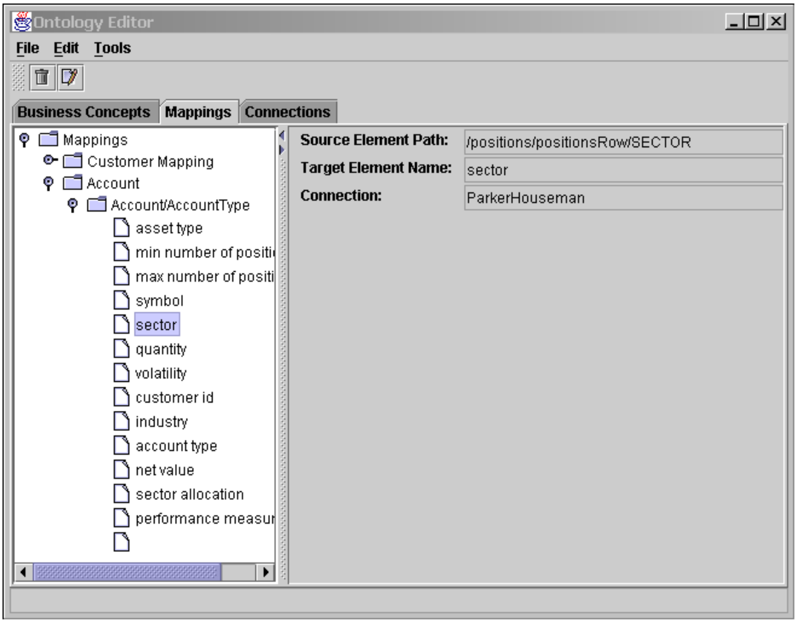
11 Click the Finish button to exit the Map Wizard. The Ontology Editor display updates to reflect the new Mapping, which is highlighted. The Delete and Edit... buttons and option-click context menu items remain active while the Mapping object is selected.

Figure 7-79. Ontology Editor - Mapping Created and Selected



12 Drill down to the Element level by using the cursor and clicking, or by using the cursor keys. When you highlight one of the Elements, its Mapping characteristics display within the right-hand panel.

Figure 7-80. Ontology Editor - Mapping Characteristics



Source Element Path: /positions/positionsRow/SECTOR
Target Element Name: sector
Connection: ParkerHouseman

Creating Joins

Within the Mapping Editor, you can join Elements from one Business Concept to Source Elements from different Connections. This is a Heterogeneous Join. However, to enable Knowledge Broker to “calibrate” elements from different Connections, you must define a Join between two Source Elements from different Connections that are fundamentally identical. This is similar to defining a Primary Key when you join different normalized tables in a relational database.

For example, You might have a Root Business Concept called “Customer” with associated SSN, Address, and Salary Elements. You Mapped the SSN Element to a “Social Security Number” Source Element and Address to a Source Element “Subscriber Address.” Both these Source Elements are from Connection1, but you Mapped Salary to an “Annual Salary” Source Element from Connection2.

You must calibrate Connection1 and Connection2. In this hypothetical example, you Join the Source Element “Social Security Number” from Connection1 to the Source

Element “Employee Taxpayer Number” in Connection2 because you know these two Elements contain equivalent information.

You create Joins in two ways

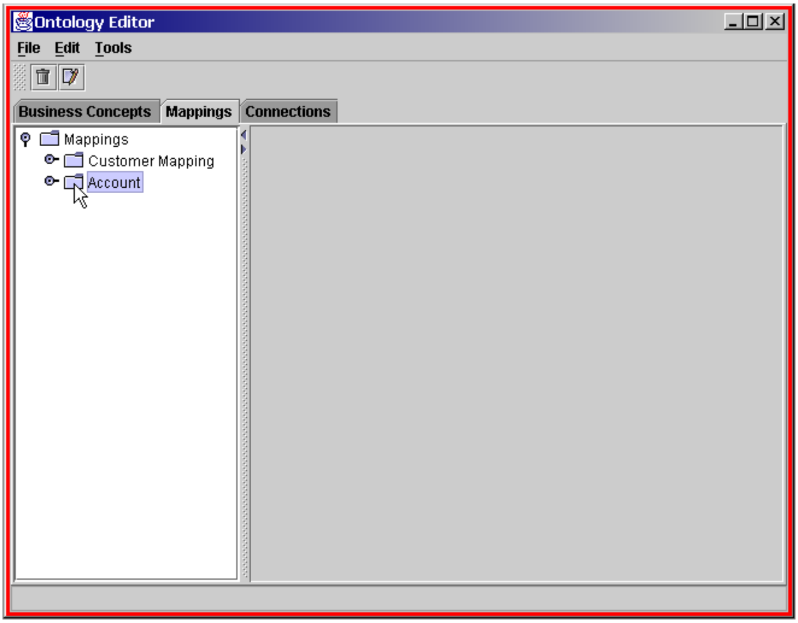
1 During the last stage of the Mapping Wizard. In the Creating a Mapping example above, you clicked the Finish button to exit the Mapping Wizard in Step 11. You could have instead clicked the Next button to continue to the Join Phase.

2 By editing a Mapped Business Concept.

Editing a Mapped Business Concept to Add a Join

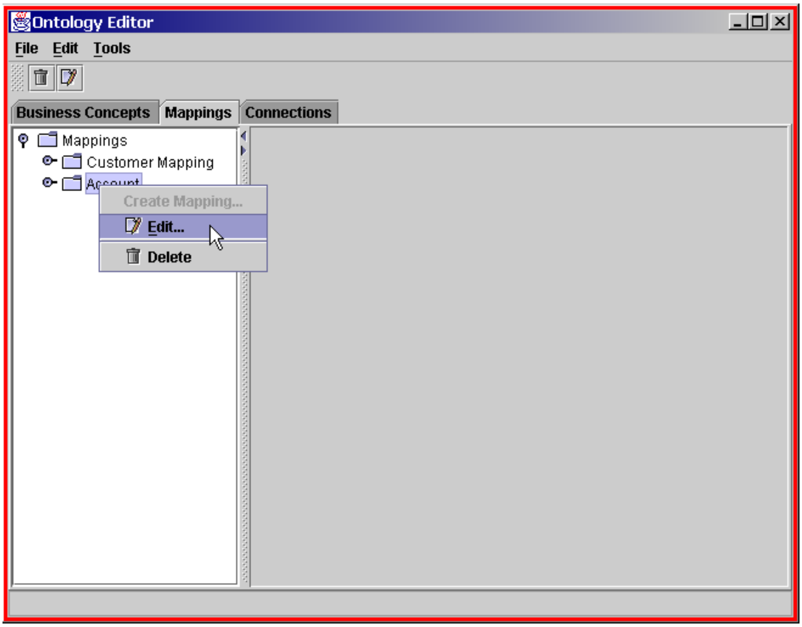
1 Open the Ontology Editor. Select a Mapped Business Concept.

Figure 7-81. Ontology Editor - Mapped Business Concept Selected



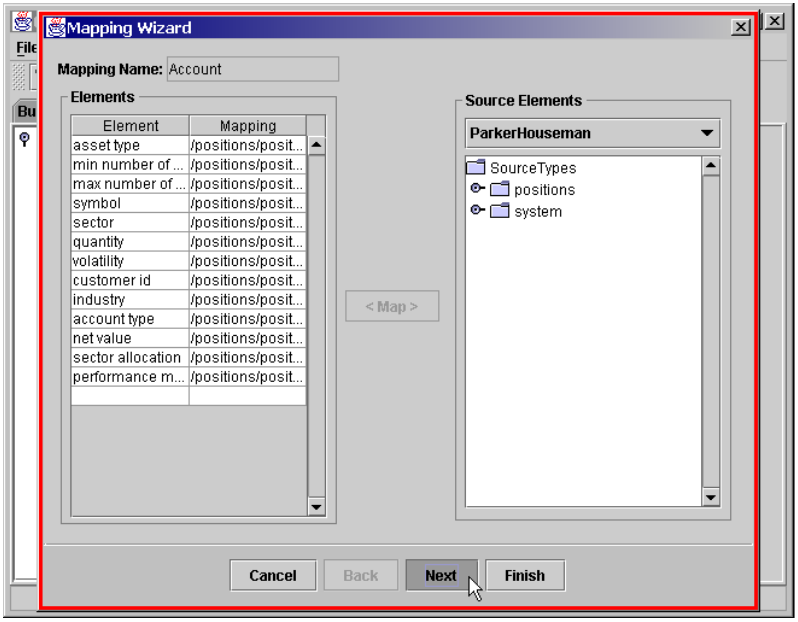
2 Option-click the Mapped Business Concept to select the Edit... function.

Figure 7-82. Ontology Editor - Edit Mapped Business Concept



3 The Mapping Wizard displays. Element Mappings previously completed display XPath information in the left-hand panel. Click the Next button.

Figure 7-83. Mapping Wizard Displays Previously Mapped Element XPaths



Mapping Name: Account

Elements

Element	Mapping
asset type	/positions/posit...
min number of ...	/positions/posit...
max number of ...	/positions/posit...
symbol	/positions/posit...
sector	/positions/posit...
quantity	/positions/posit...
volatility	/positions/posit...
customer id	/positions/posit...
industry	/positions/posit...
account type	/positions/posit...
net value	/positions/posit...
sector allocation	/positions/posit...
performance m...	/positions/posit...

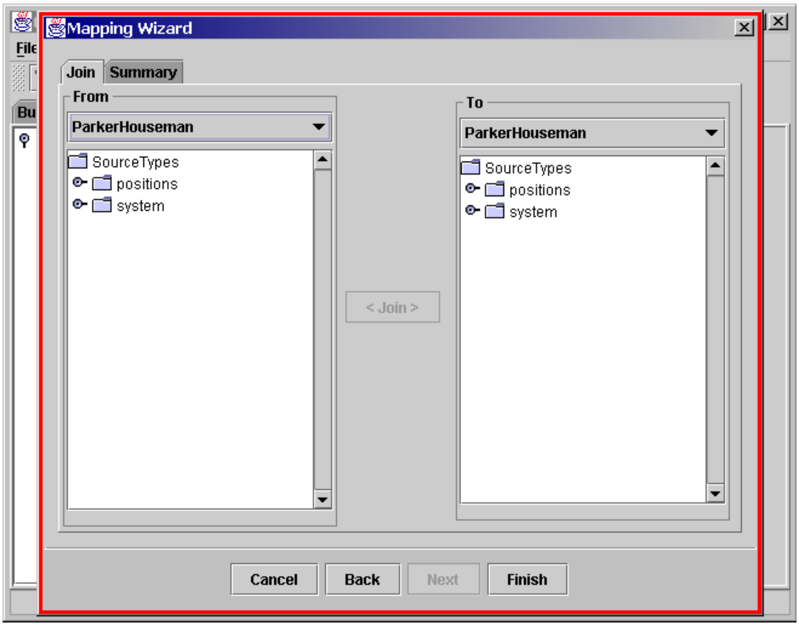
Source Elements

ParkerHouseman

- SourceTypes
  - positions
  - system

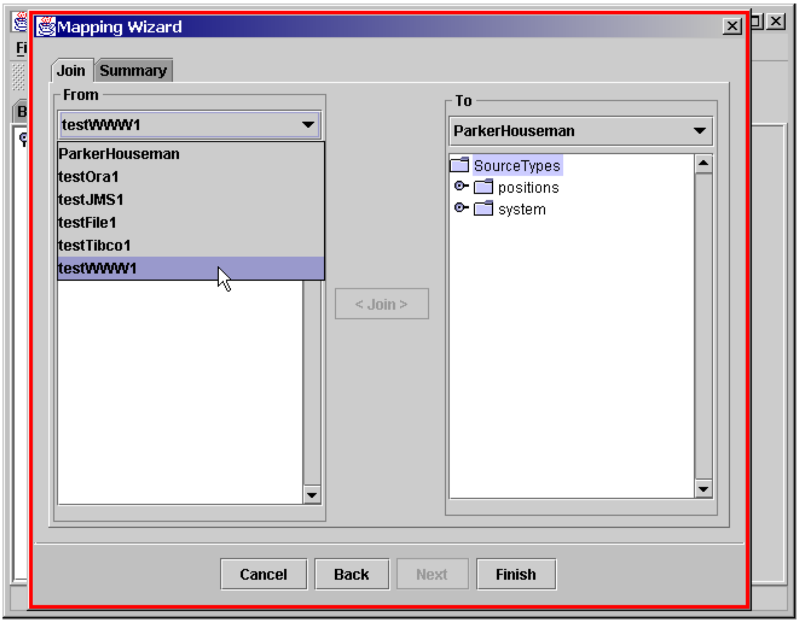
4 The Join function screen displays. The left-hand "From" panel and the right-hand "To" panel display a drop-down menu.

Figure 7-84. Mapping Wizard - Join Screen



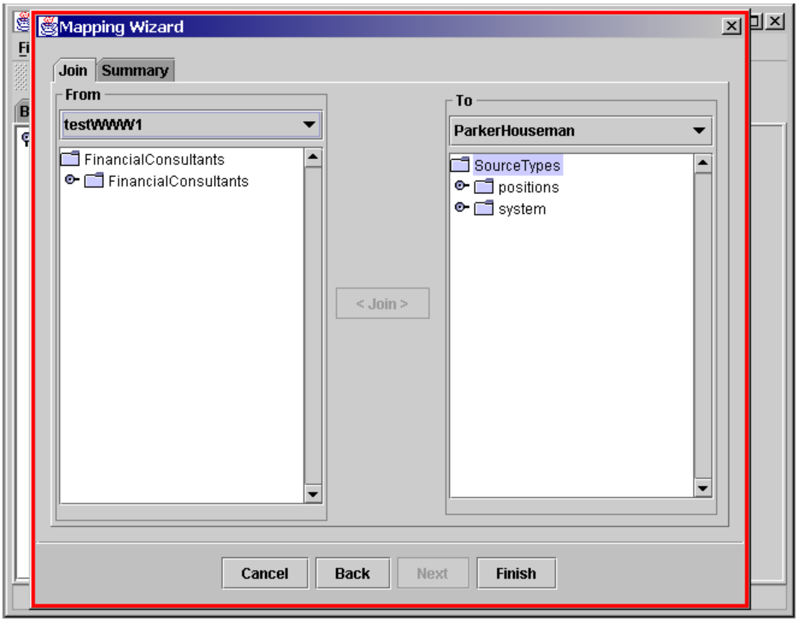
5 Open the left-hand "From" drop-down menu by using the mouse button or the cursor keys. A list of the available, previously defined Connections appears. Highlight the first Connection you want to Join.

Figure 7-85. Mapping Wizard - Join Screen "From" Connection Schema Selection



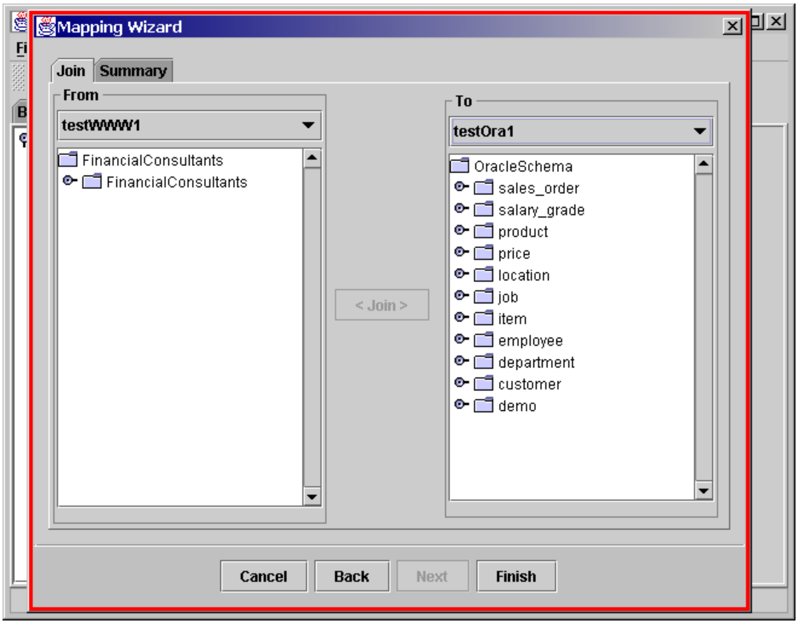
6 Select this Connection by clicking the mouse button or pressing the Enter key. The Schema associated with the selected Connection displays.

Figure 7-86. Mapping Wizard - Join Screen “From” Connection Schema Selected



7 Repeat this procedure to select your second Connection to be Joined on the right-hand “To” side.

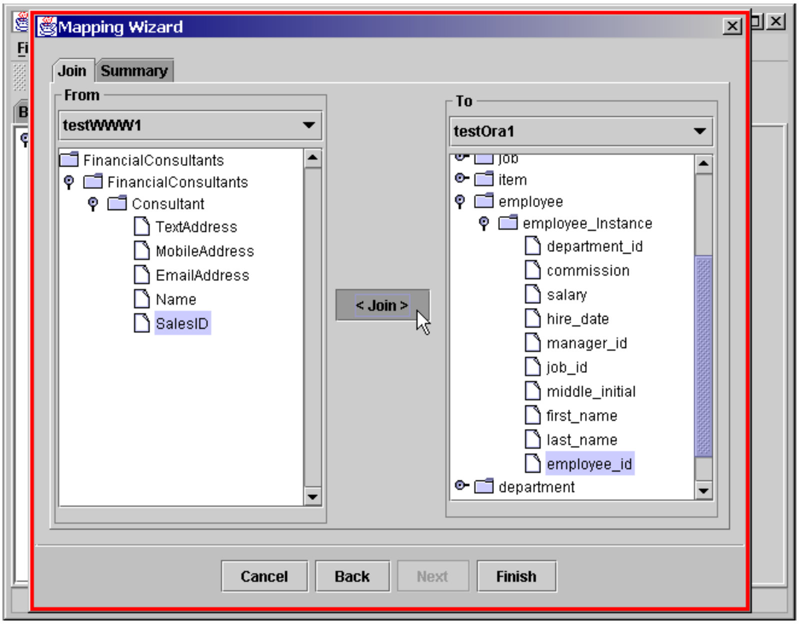
Figure 7-87. Mapping Wizard - Join Screen “To” Connection Schema Selected



8 Expand the left-hand “From” and right-hand “To” Connection Schemas until you identify the key Element to be Joined. In this example, the left-hand `SalesID` Source Element will be Joined with the right-hand `employee_id` Source Element. This will

produce a concordance between these two Connections that will calibrate the Business Concept. Highlight both Source Elements. Click the <Join> button.

Figure 7-88. Mapping Wizard - Joining the Source Elements



Mapping Wizard

Join | Summary

From: testWWW1

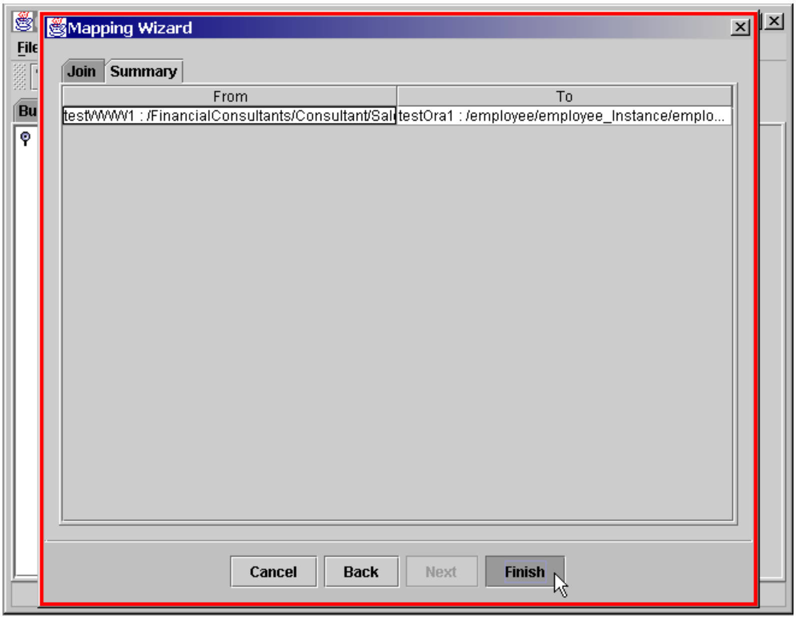
- FinancialConsultants
  - FinancialConsultants
    - Consultant
      - TextAddress
      - MobileAddress
      - EmailAddress
      - Name
      - SalesID

To: testOra1

- job
- item
- employee
  - employee_Instance
    - department_id
    - commission
    - salary
    - hire_date
    - manager_id
    - job_id
    - middle_initial
    - first_name
    - last_name
    - employee_id
- department

9 The two Source Elements have been Joined. Click the Summary button. The XPath/XQuery description of the Join displays. Click the Finish button to complete the Join process.

Figure 7-89. Mapping Wizard - Join Summary Screen



From	To
testWWW1 : /FinancialConsultants/Consultant/Sal	testOra1 : /employee/employee_Instance/emplo...

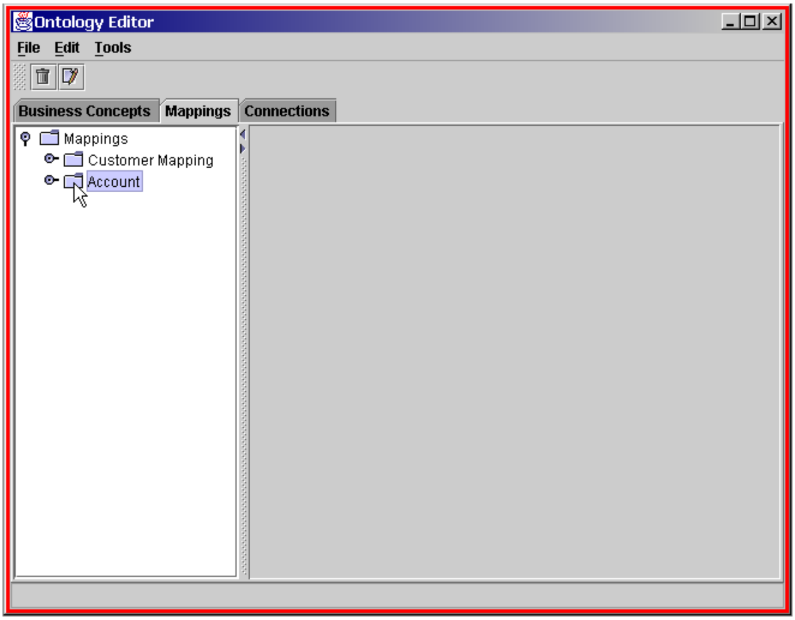
XQuery is a query definition and extraction language for XML documents that uses XPath for addressing. For more information, go here:

`http://www.w3.org/TR/xpath`

`http://www.w3.org/XML/Query`

10 The Ontology Editor displays. The Business Concept Mapping you Joined displays highlighted. You have completed a Join.

Figure 7-90. Ontology Editor - Mapped Business Concept Selected



Creating User-Defined Queries

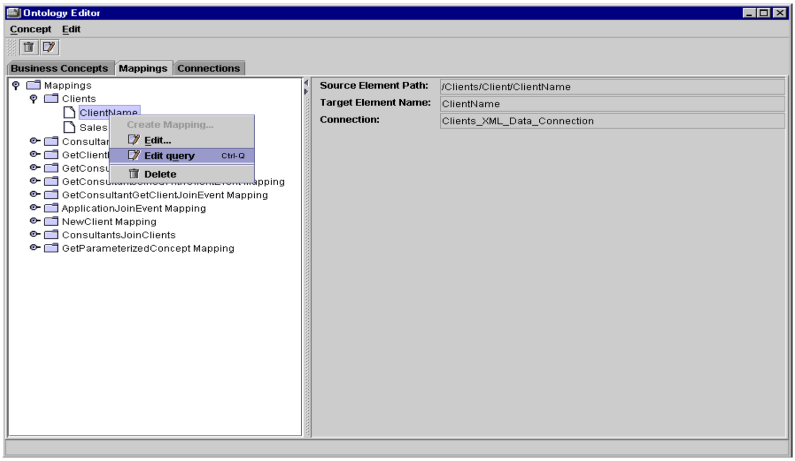
Mappings are actually structured queries that tell Knowledge Broker which external datasources to access, where and how to collect and extract data from them, and how and to which Business Concept Elements this data should be assigned. Knowledge Broker auto-generates queries based on your selections within the Mapping Wizard and stores them as XQuery statements. You use the Edit Query command to override the auto-generated XQuery with your own custom XQuery statements to create greater flexibility and precision. For information on writing these user-defined queries (UDQs) in Knowledge Broker, see *Writing XQueries on page 499*.

Using the Edit Query Command

1 Within the Ontology Editor, highlight a Mapping you wish to override.

2 Use option-click to display the context menu for this Mapping. Select **Edit Query** (or press Ctrl-Q):

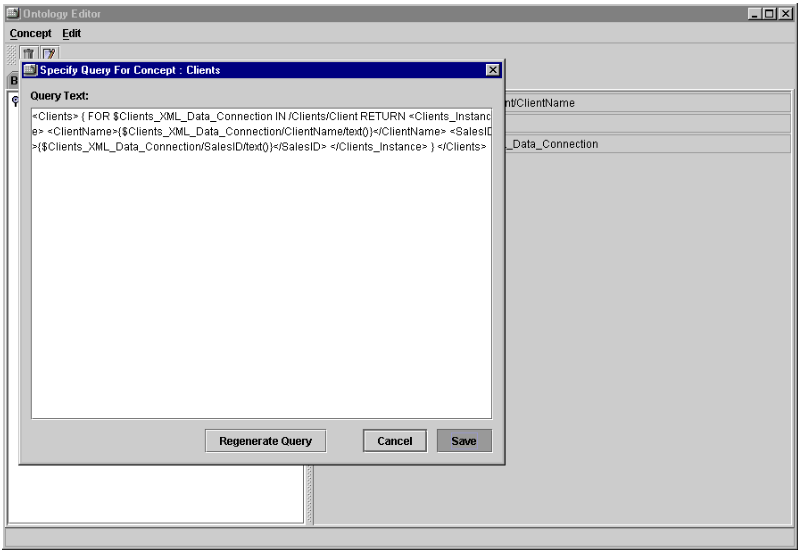
Figure 7-91. Ontology Editor - Selecting Mapping Override



Source Element Path: /Clients/Client/ClientName
Target Element Name: ClientName
Connection: Clients_XML_Data_Connection

3 The Specify Query dialog displays. You can edit the XQuery code directly by typing within the panel, or you can paste in XQuery from external sources. When you have finished entering your custom XQuery, click the **Save** button.

Figure 7-92. Ontology Editor - Entering Mapping Override XQuery Code



```
<Clients> { FOR $Clients_XML_Data_Connection IN /Clients/Client RETURN <Clients_Instanc
e> <ClientName>{$Clients_XML_Data_Connection/ClientName/text()}</ClientName> <SalesID
>{$Clients_XML_Data_Connection/SalesID/text()}</SalesID> </Clients_Instance> } </Clients>
```

To restore the original, auto-generated XQuery, click the **Regenerate Query** button.

XQuery is a query definition and extraction language for XML documents that uses XPath for addressing. For more information, go here:

```
http://www.w3.org/TR/xpath
http://www.w3.org/XML/Query
```

Chapter

8

Creating Interactions

Interactions enable advanced Knowledge Broker users to access low-level data access and query manipulations. They provide a flexibility and power not available using Mappings and Connections.

Introducing Interactions

Interactions are a “raw” method for interacting with Knowledge Broker. Using Interactions, you can fine-tune Knowledge Broker’s data connections, mappings, join, and data extraction operations.

Interactions enable you to directly define an Interaction Spec, that is, a script or flow that Knowledge Broker executes on demand. Knowledge Broker’s user interface generates these flows but its flexibility, and their power, is limited compared with user-created Interactions.

Types of Interactions

There are four available Interaction Types:

a Query

b Delete

c Insert

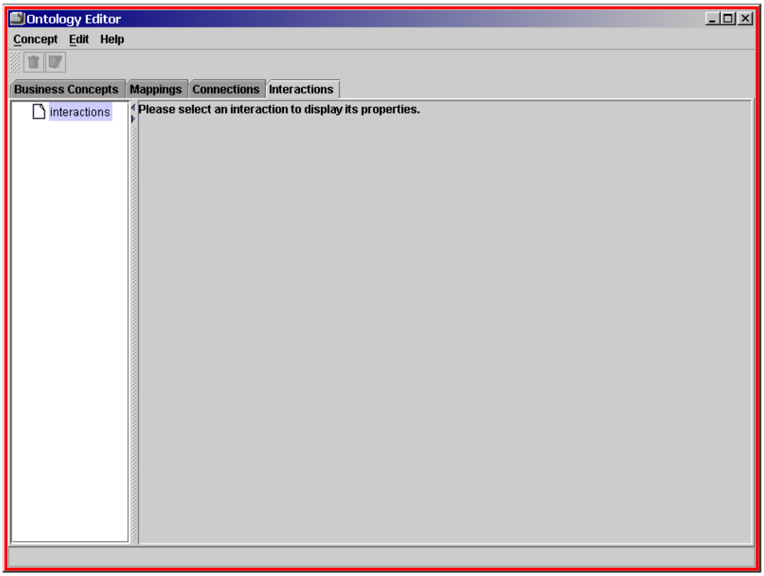
d Update

Creating an Interaction

1 Start the Ontology Editor, if it is not currently open.

2 Select the **Interactions** tab.

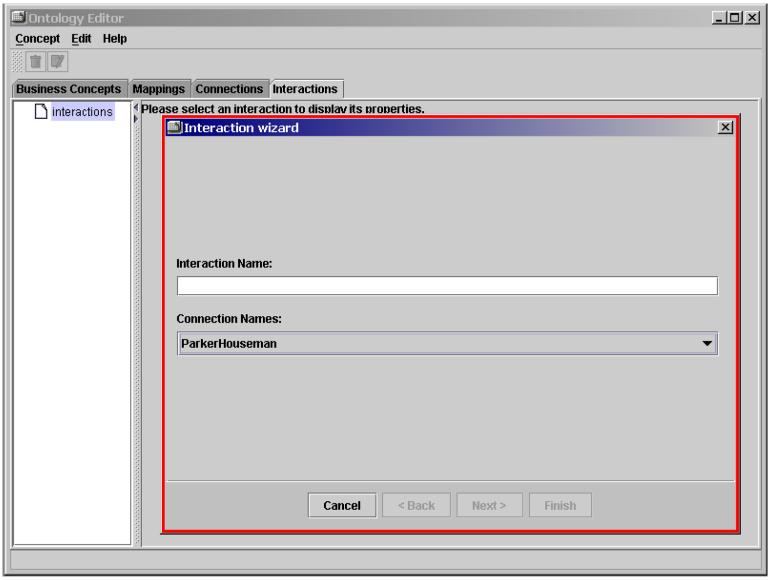
Figure 8-1. Interactions Editor - Initial Display



Please select an interaction to display its properties.

3 Select **Concept > New > New Interaction.** The Interaction Wizard displays.

Figure 8-2. Interaction Wizard - Initial Display



Please select an interaction to display its properties.

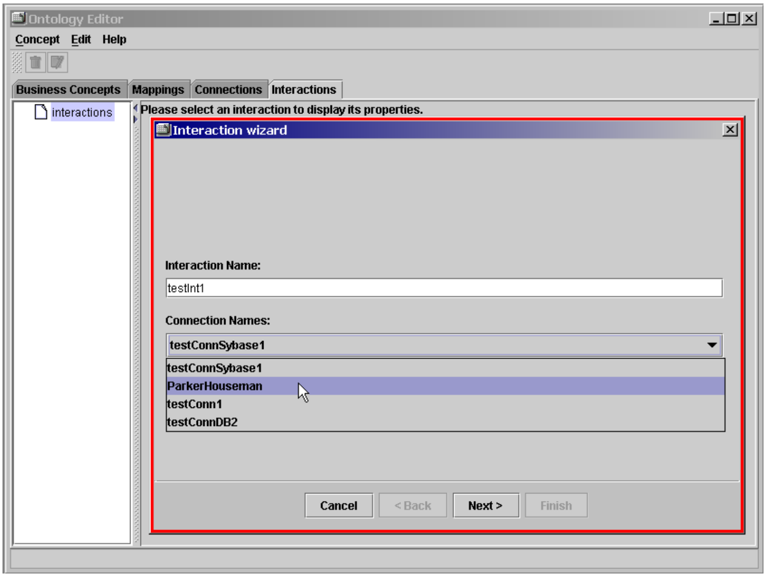
Interaction Name:

Connection Names: ParkerHouseman

4 Click in the **Interaction Name:** field and type an identifying name for the Interaction.

5 Every Interaction must be associated with a Connection. Click the **Connection Names:** drop-down menu to display a list of available connections.

Figure 8-3. Interaction Wizard - Available Connections



Please select an interaction to display its properties.

Interaction Name:

testInt1

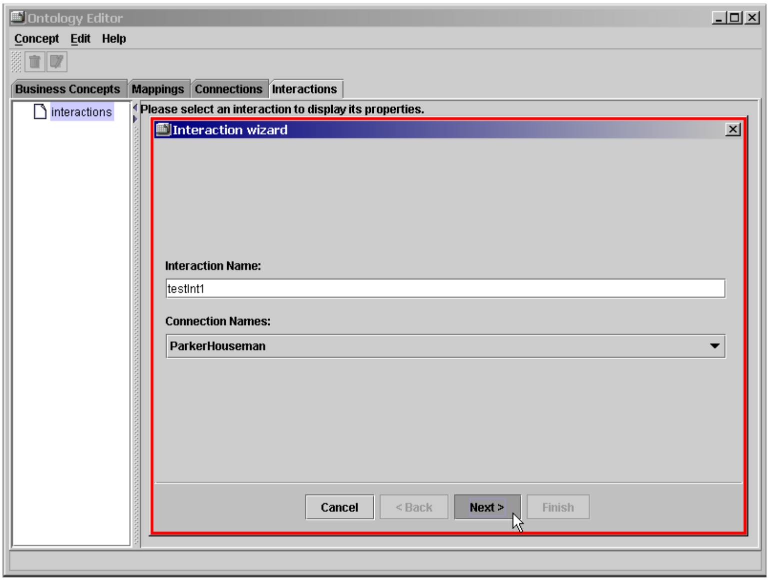
Connection Names:

testConnSybase1

testConnSybase1
ParkerHouseman
testConn1
testConnDB2

6 Select the desired Connection. Click the **Next >** button to continue.

Figure 8-4. Interaction WizardWizard - Connection Selected



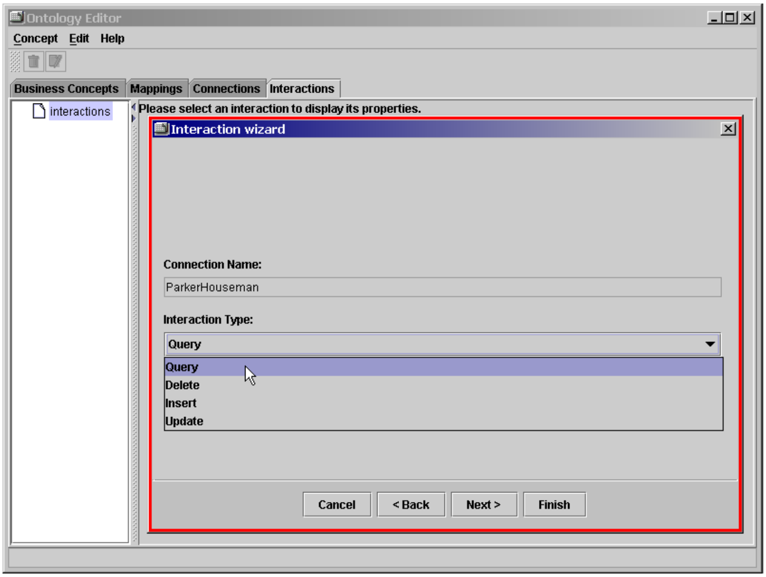
Please select an interaction to display its properties.

Interaction Name: testInt1

Connection Names: ParkerHouseman

7 The Interaction Wizard registers the chosen Connection and updates. Click the **Interaction Type:** drop-down menu to select which Interaction Type to associate with this Interaction.

Figure 8-5. Interaction Wizard - Available Interaction Types



Please select an interaction to display its properties.

Connection Name: ParkerHouseman

Interaction Type: Query

Query
Delete
Insert
Update

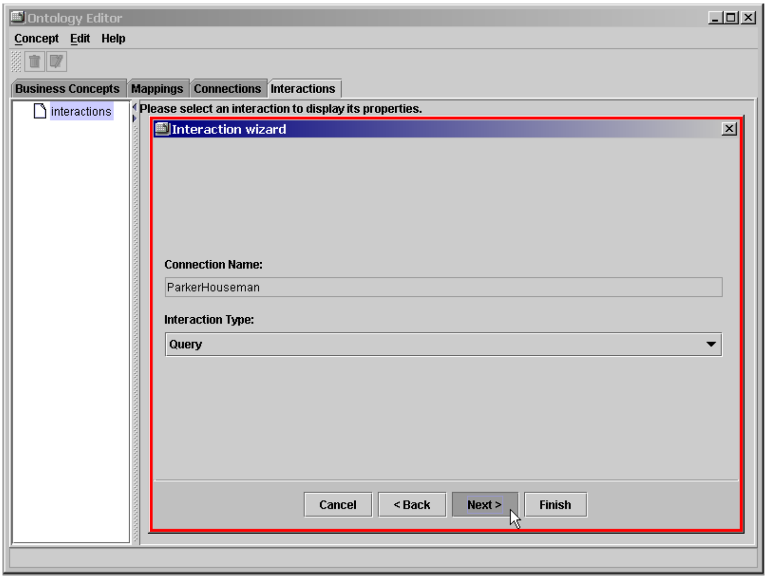
There are four available Interaction Types:

- **a** Query
- **b** Delete
- **c** Insert
- **d** Update

The subsequent Interaction properties screen varies depending on the selected Interaction Type.

8 Select the desired Interaction Type. Click the **Nex t>** button to continue.

Figure 8-6. Interaction Wizard - Interaction Type Query Selected



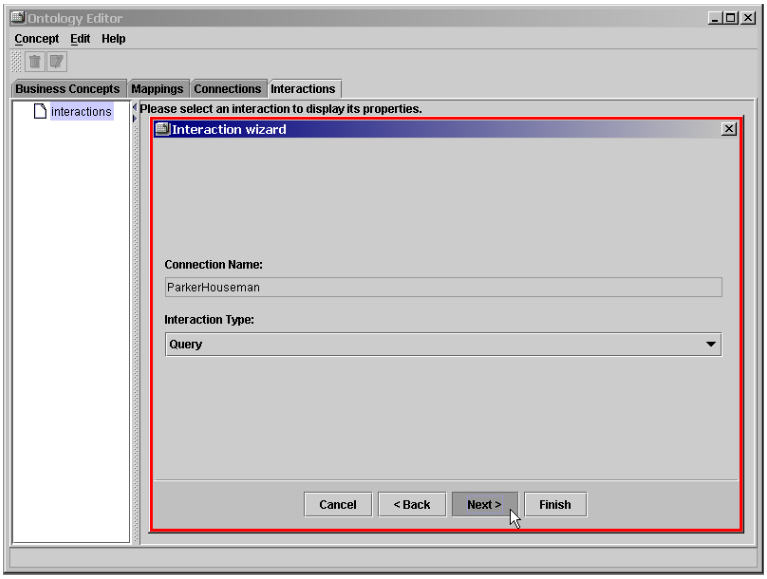
Please select an interaction to display its properties.

Connection Name:
ParkerHouseman

Interaction Type:
Query

9 The Interaction Wizard display updates to reflect the selected Interaction Type:

Figure 8-7. Interaction Wizard - Interaction Properties - Query Selected

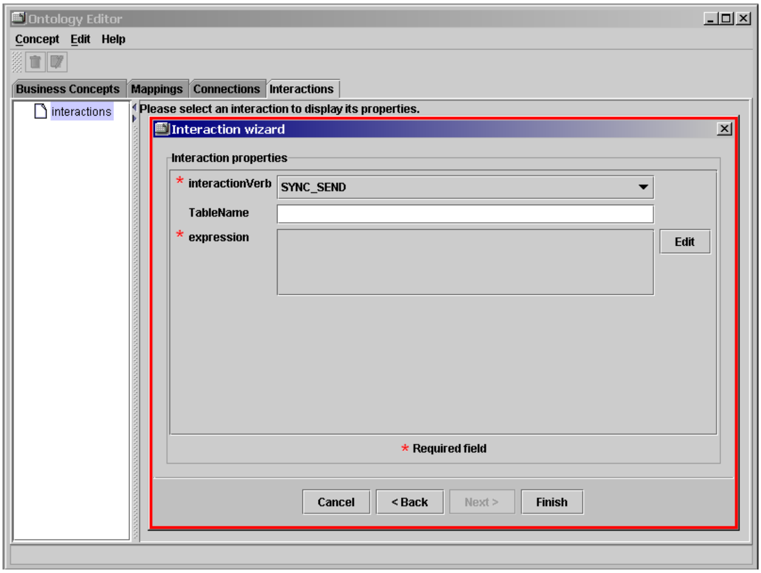


Please select an interaction to display its properties.

Connection Name: ParkerHouseman

Interaction Type: Query

Figure 8-8. Interaction Wizard - Interaction Properties - Delete Selected



Please select an interaction to display its properties.

Interaction properties

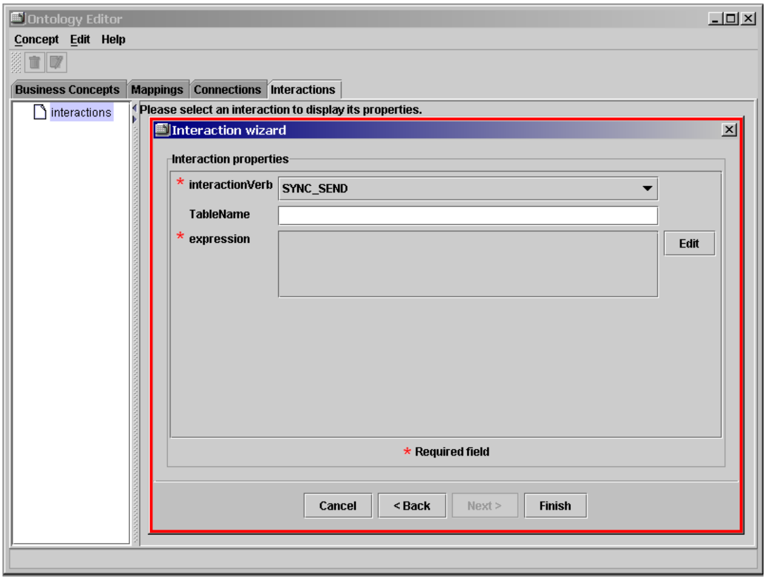
* interactionVerb: SYNC_SEND

TableName:

* expression:

* Required field

Figure 8-9. Interaction Wizard - Interaction Properties - Insert Selected



Please select an interaction to display its properties.

Interaction properties

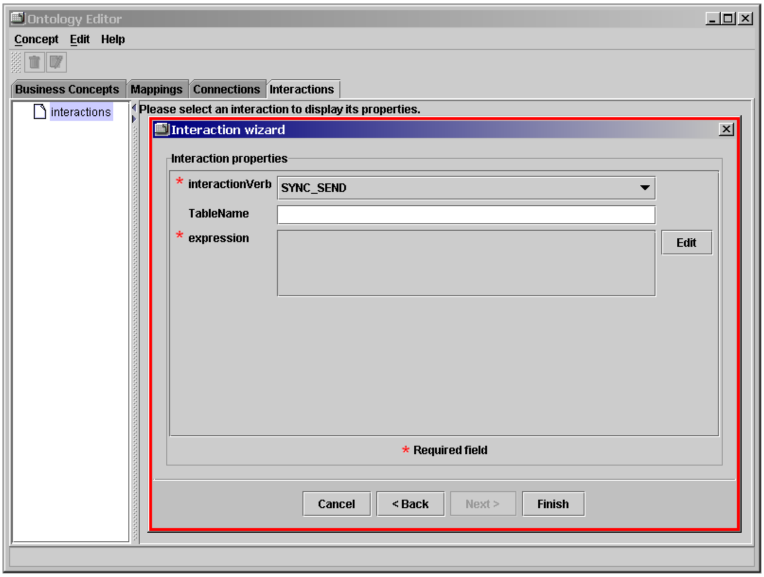
* interactionVerb: SYNC_SEND

TableName:

* expression:

* Required field

Figure 8-10. Interaction Wizard - Interaction Properties - Update Selected



Please select an interaction to display its properties.

Interaction properties

* interactionVerb: SYNC_SEND

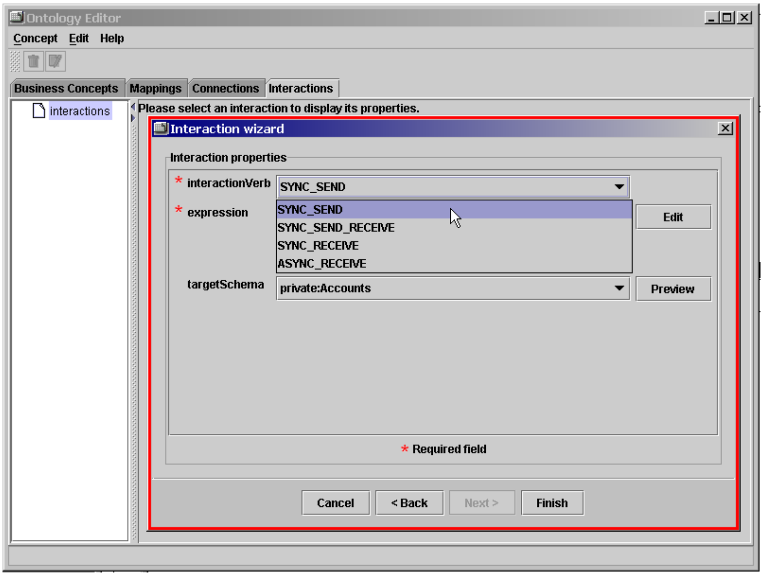
TableName:

* expression:

* Required field

10 The Interaction Wizard now displays the Interaction properties. Click the interactionVerb to display the list of available J2EE EIS interaction modes:

Figure 8-11. Interaction Wizard - Interaction Properties - Interaction Verbs



Please select an interaction to display its properties.

* interactionVerb: SYNC_SEND

SYNC_SEND
SYNC_SEND_RECEIVE
SYNC_RECEIVE
ASYNC_RECEIVE

* expression

targetSchema: private:Accounts

* Required field

There are four available Interaction Verbs:

a SYNC_SEND

b SYNC_SEND_RECEIVE

c SYNC_RECEIVE

d ASYNC_RECEIVE

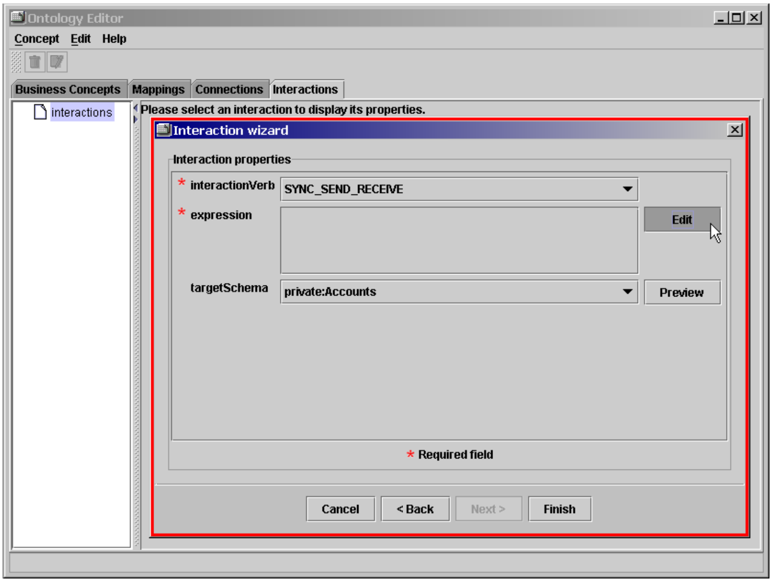
The Interaction Verbs describe the mode of interaction of Knowledge Broker with the underlying data source referenced by the active Connection.

The list of available Interaction Verbs generally corresponds to those described in the J2EE Connector Architecture Common Client Interface (CCI).

You can find details about the Interaction Verbs here:
`http://java.sun.com/j2ee/apidocs-1_0-fr/api/javax/resource/cci/InteractionSpec.html`

11 All four Verbs execute an XQuery expression. To input XQuery, click the **Edit** button:

Figure 8-12. Interaction Wizard - Interaction Properties - Select Edit Expression



Please select an interaction to display its properties.

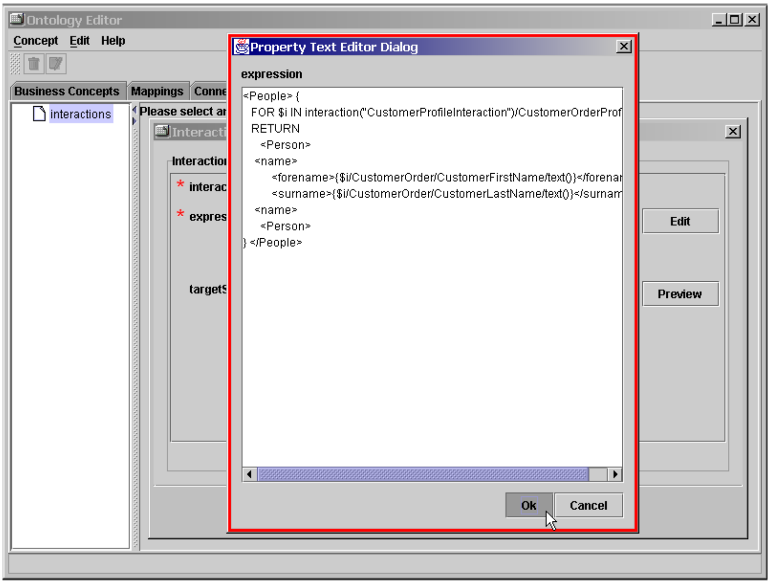
* interactionVerb SYNC_SEND_RECEIVE

* expression

targetSchema private:Accounts

12 The **Property Text Editor Dialog** displays. Type or paste a suitable XQuery into this window. When you are finished creating your expression, click the **OK** button.

Figure 8-13. Interaction Wizard - Interaction Properties - Entering XQuery Expression



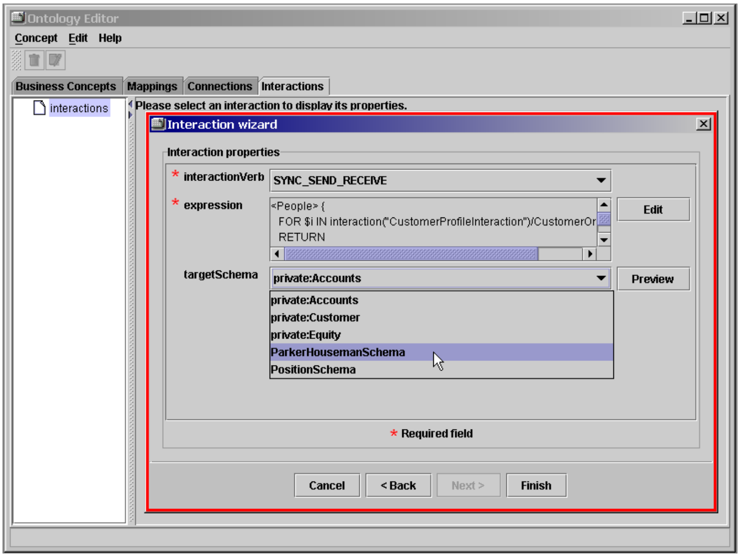
```
<People> {
  FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProf
  RETURN
    <Person>
   <name>
       <forename>{$i/CustomerOrder/CustomerFirstName/text()}</forenar
       <surname>{$i/CustomerOrder/CustomerLastName/text()}</surnam
   <name>
    <Person>
} </People>
```

For details about writing XQuery for Knowledge Broker, see *Writing XQueries on page 499*.

13 The Interaction Wizard updates, displaying the new expression contents.

14 For Query Interactions, the screen displays a targetSchema drop-down menu. Click this to optionally select the desired targetSchema against which the XQuery executes. For all other Interactions, optionally enter a TableName against which the XQuery executes.

Figure 8-14. Interaction Wizard - Interaction Properties - Query - targetSchema



Please select an interaction to display its properties.

Interaction properties

* interactionVerb SYNC_SEND_RECEIVE

* expression

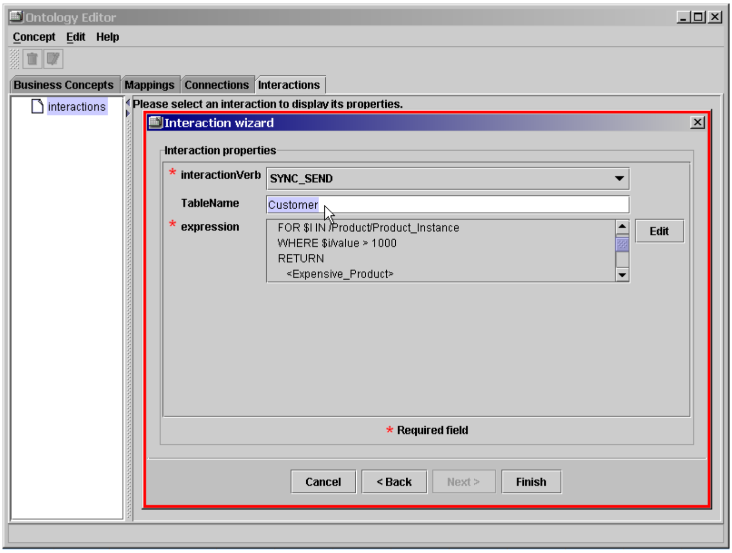
```
<People> {
 FOR $i IN interaction("CustomerProfileInteraction")/CustomerOr
 RETURN
```

targetSchema private:Accounts

private:Accounts
private:Customer
private:Equity
ParkerHousemanSchema
PositionSchema

* Required field

Figure 8-15. Interaction Wizard - Interaction Properties - TableName



Please select an interaction to display its properties.

Interaction properties

* interactionVerb SYNC_SEND

TableName Customer

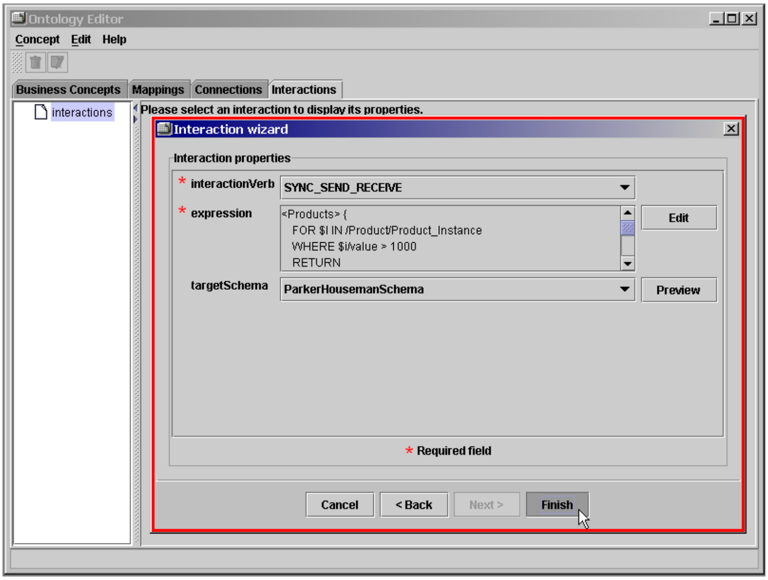
* expression

```
 FOR $i IN /Product/Product_Instance
 WHERE $i/value > 1000
 RETURN
   <Expensive_Product>
```

* Required field

15 Click the **Finish** button to complete the Interaction Wizard.

Figure 8-16. Interaction Wizard - Finished



Please select an interaction to display its properties.

Interaction properties

* interactionVerb SYNC_SEND_RECEIVE

* expression

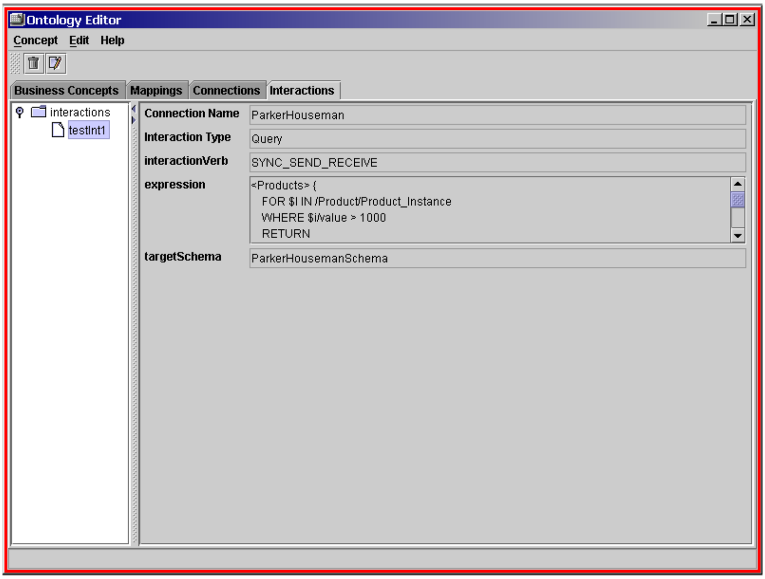
```
<Products> {
  FOR $I IN /Product/Product_Instance
  WHERE $i/value > 1000
  RETURN
```

targetSchema ParkerHousemanSchema

* Required field

16 The Ontology Editor displays with the newly created Interaction highlighted and its details displayed within the right-hand panel.

Figure 8-17. Ontology Editor - Created Interaction Displayed



Connection Name ParkerHouseman

Interaction Type Query

interactionVerb SYNC_SEND_RECEIVE

expression
```
<Products> {
   FOR $I IN /Product/Product_Instance
   WHERE $i/value > 1000
   RETURN
```

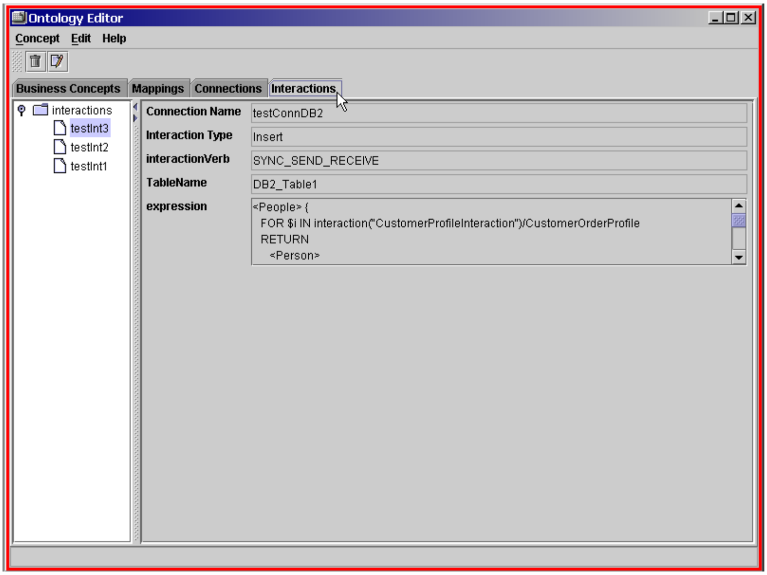
targetSchema ParkerHousemanSchema

Modifying an Interaction

1 Start the Ontology Editor, if it is not currently open.

2 Select the **Interactions** tab. The Interactions Editor displays the existing Interactions.

Figure 8-18. Modify Interaction - Initial Display

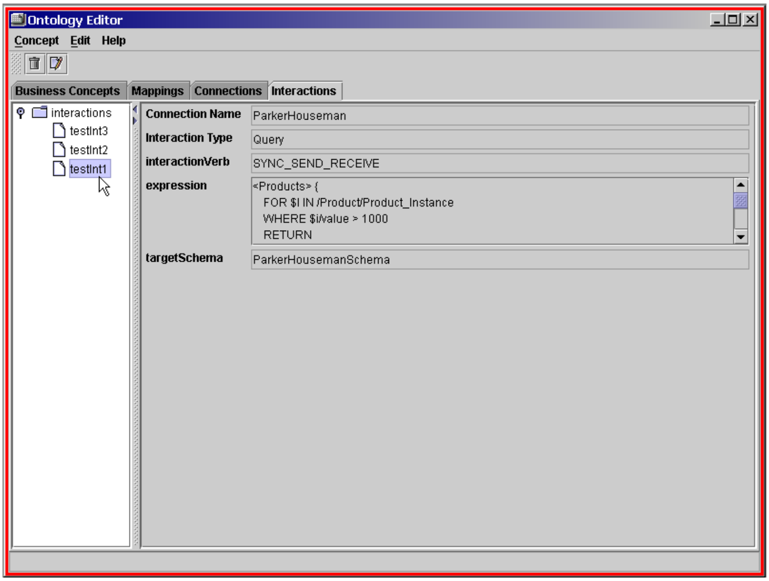


Connection Name testConnDB2
Interaction Type Insert
interactionVerb SYNC_SEND_RECEIVE
TableName DB2_Table1
expression

```
<People> {
  FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProfile
  RETURN
    <Person>
```

3 Select the Interaction to modify. The right-hand panel updates to display the details of the selected Interaction.

Figure 8-19. Modify Interaction - Interaction Selected



Connection Name: ParkerHouseman

Interaction Type: Query

interactionVerb: SYNC_SEND_RECEIVE

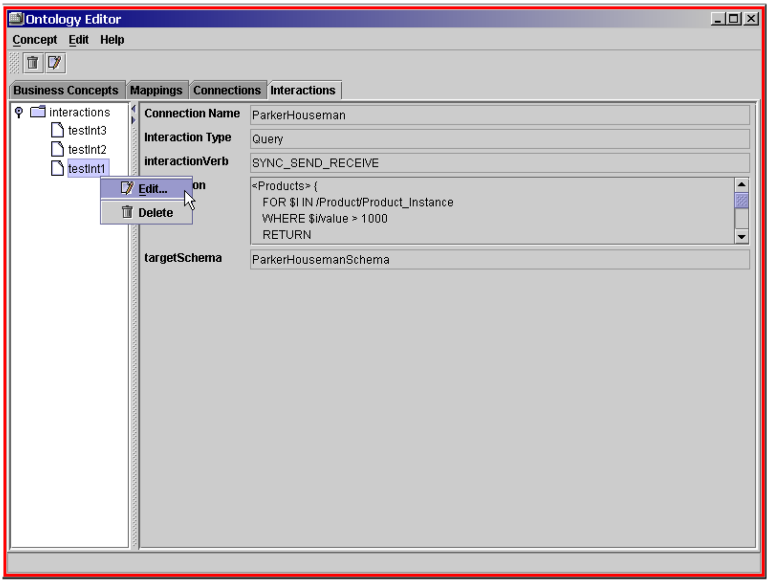
expression:

```
<Products> {
  FOR $I IN /Product/Product_Instance
  WHERE $i/value > 1000
  RETURN
```

targetSchema: ParkerHousemanSchema

4 Option-click the selected Interaction and select **Edit...**

Figure 8-20. Modify Interaction - Interaction Edit Selected



Connection Name: ParkerHouseman

Interaction Type: Query

interactionVerb: SYNC_SEND_RECEIVE

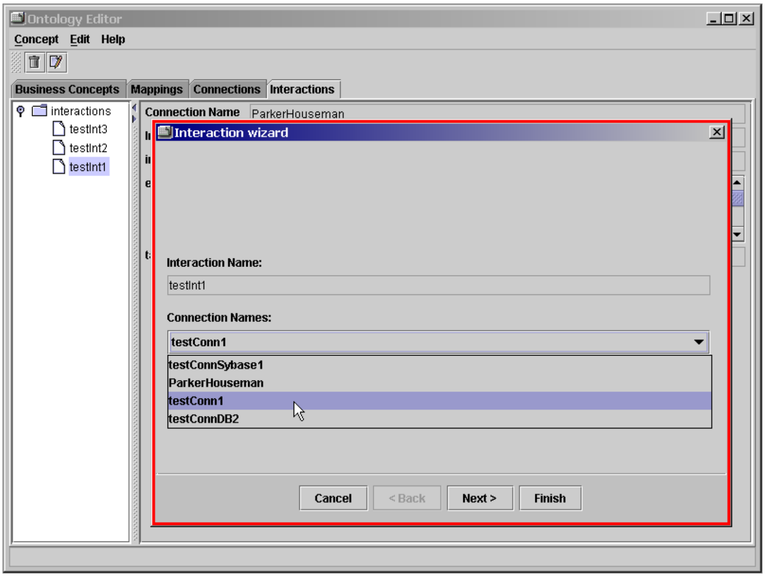
on:

```
<Products> {
  FOR $I IN /Product/Product_Instance
  WHERE $i/value > 1000
  RETURN
```

targetSchema: ParkerHousemanSchema

5 The Interaction Wizard displays. On this initial screen you can change the Interaction's Connection Name by clicking the **Connection Names:** drop-down menu.

Figure 8-21. Modify Interaction - Connection Changed



Connection Name ParkerHouseman

Interaction Name:

testInt1

Connection Names:

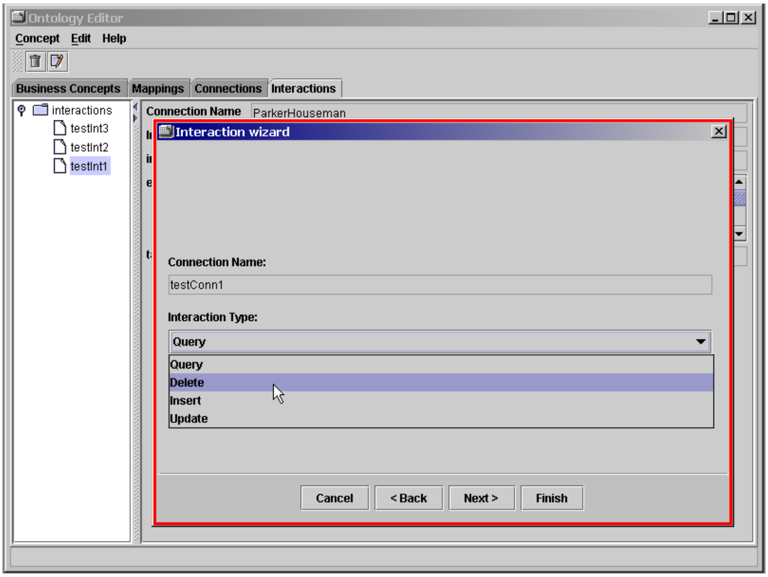
testConn1

testConnSybase1
ParkerHouseman
testConn1
testConnDB2

6 Click the **Next** > button to continue.

7 The Interaction Wizard proceeds. On this screen you can change the Interaction Type by clicking the **Interaction Type:** drop-down menu.

Figure 8-22. Modify Interaction - Interaction Type Changed



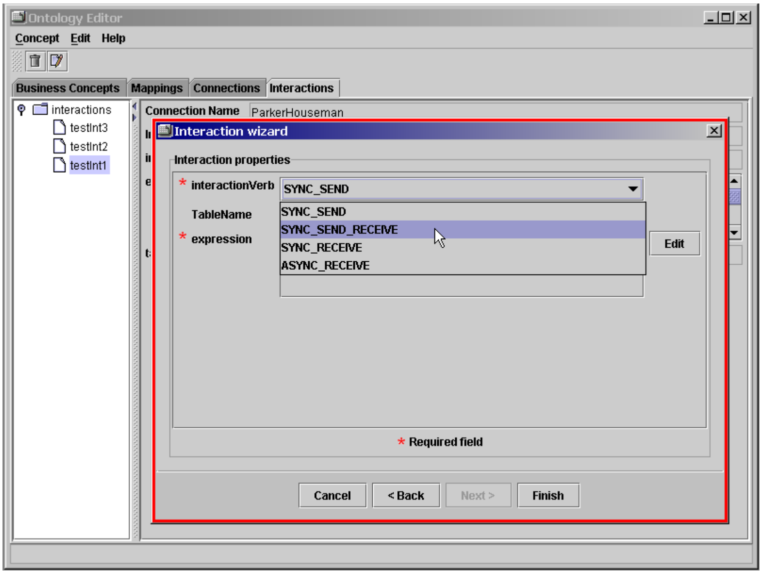
Connection Name:
testConn1

Interaction Type:
Query

8 Click the **Next >** button to continue.

9 The Interaction Wizard proceeds. On this screen you can change the Interaction Verb by clicking the **interactionVerb** drop-down menu.

Figure 8-23. Modify Interaction - Interaction Verb Changed



If you selected Query Interaction Type, you can select a different Target Schema, if desired.

If you selected any Interaction Type except Query, you can enter a different TableName, if desired.

You can also update or replace the XQuery by clicking the **Edit** button.

10 When you have completed all your desired modifications, click the **Finish** button.

11 The Interactions Editor appears, displaying your modifications in the right-hand panel.

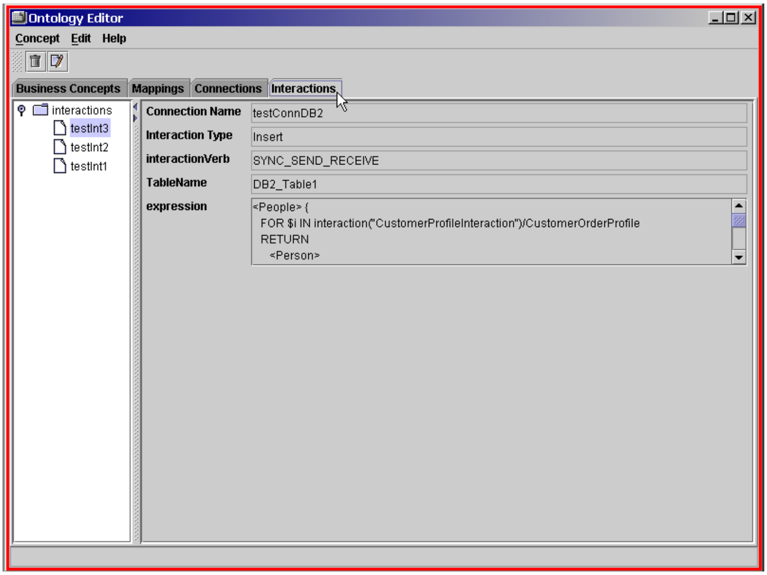
Modifying an Interaction Query

As an alternative to optionally modifying all the properties of an Interaction, you can quickly modify just the XQuery expression.

1 Start the Ontology Editor, if it is not currently open.

2 Select the **Interactions** tab. The Interactions Editor displays the existing Interactions.

Figure 8-24. Modify Interaction Query - Initial Display



Connection Name: testConnDB2

Interaction Type: Insert

interactionVerb: SYNC_SEND_RECEIVE

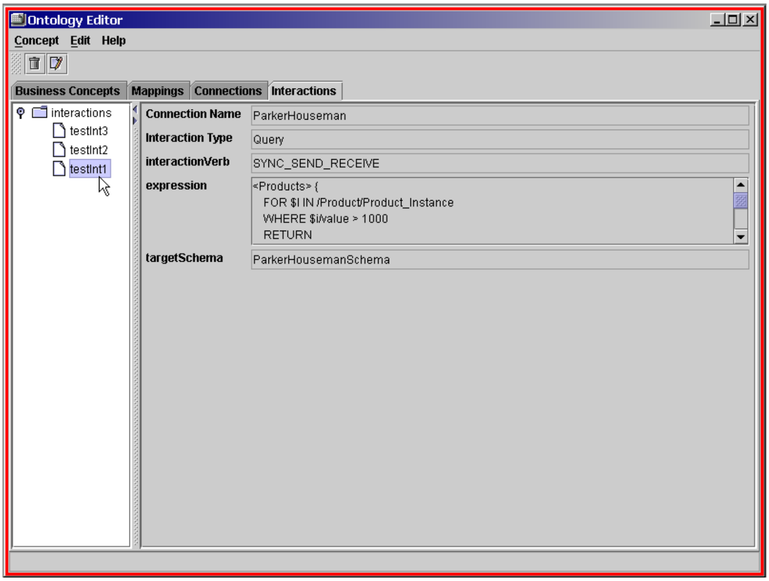
TableName: DB2_Table1

expression:

```
<People> {
  FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProfile
  RETURN
    <Person>
```

3 Select the Interaction to modify. The right-hand panel updates to display the details of the selected Interaction.

Figure 8-25. Modify Interaction Query - Interaction Selected



Connection Name: ParkerHouseman

Interaction Type: Query

interactionVerb: SYNC_SEND_RECEIVE

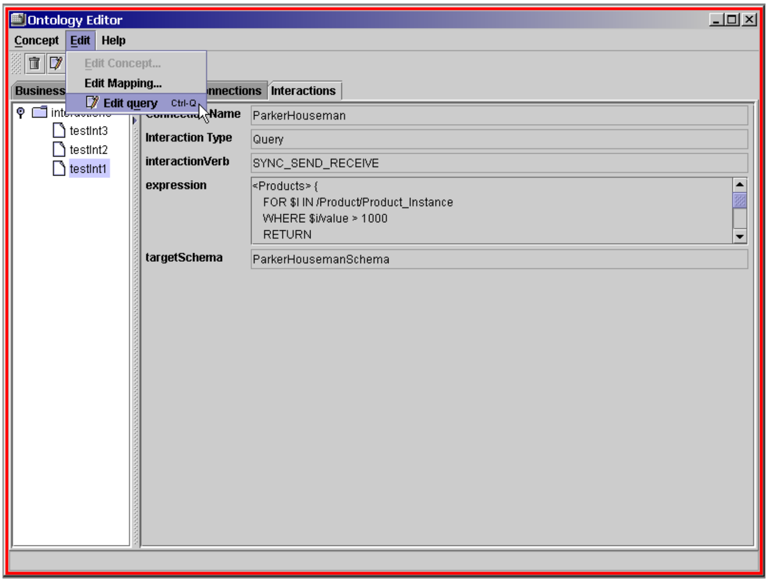
expression:

```
<Products> {
  FOR $I IN /Product/Product_Instance
  WHERE $i/value > 1000
  RETURN
```

targetSchema: ParkerHousemanSchema

4 Select **Edit > Edit query.** The Interaction Wizard displays

Figure 8-26. Modify Interaction Query - Edit Query Selected



ConnectionName: ParkerHouseman

Interaction Type: Query

interactionVerb: SYNC_SEND_RECEIVE

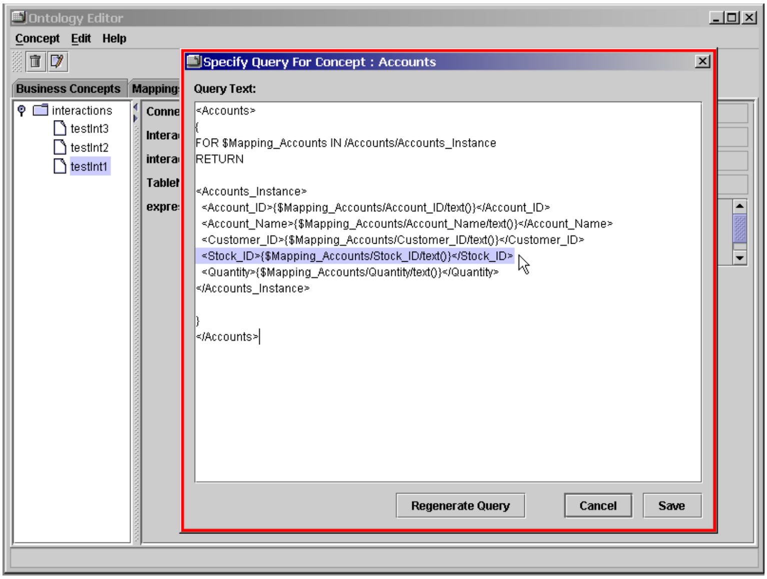
expression:

```
<Products> {
  FOR $I IN /Product/Product_Instance
  WHERE $i/value > 1000
  RETURN
```

targetSchema: ParkerHousemanSchema

5 The Specify Query dialog displays.

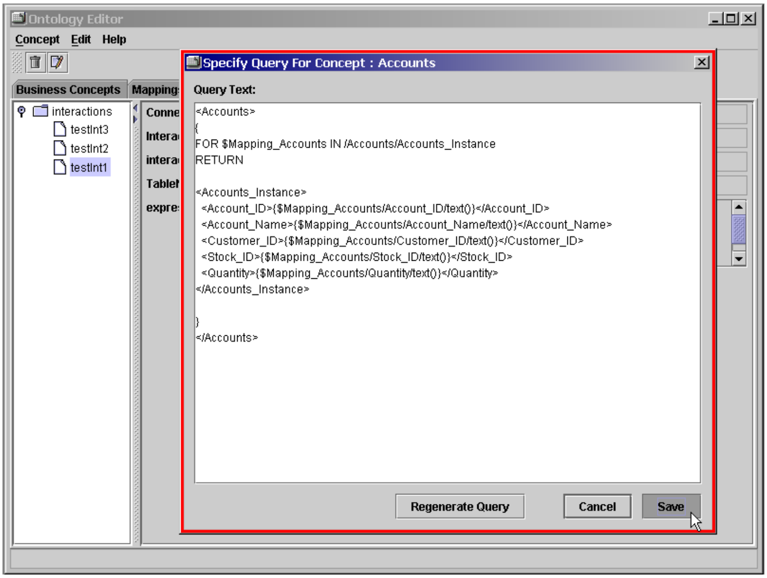
Figure 8-27. Modify Interaction Query - Specify Query Dialog



```
<Accounts>
{
FOR $Mapping_Accounts IN /Accounts/Accounts_Instance
RETURN

<Accounts_Instance>
  <Account_ID>{$Mapping_Accounts/Account_ID/text()}</Account_ID>
  <Account_Name>{$Mapping_Accounts/Account_Name/text()}</Account_Name>
  <Customer_ID>{$Mapping_Accounts/Customer_ID/text()}</Customer_ID>
  <Stock_ID>{$Mapping_Accounts/Stock_ID/text()}</Stock_ID>
  <Quantity>{$Mapping_Accounts/Quantity/text()}</Quantity>
</Accounts_Instance>

}
</Accounts>
```

6 You can type or paste modifications to the XQuery directly into this dialog. When you have finished, click the **Save** button.

Figure 8-28. Modify Interaction Query - Specify Query Dialog



Query Text:

```
<Accounts>
{
FOR $Mapping_Accounts IN /Accounts/Accounts_Instance
RETURN

<Accounts_Instance>
  <Account_ID>{$Mapping_Accounts/Account_ID/text()}</Account_ID>
  <Account_Name>{$Mapping_Accounts/Account_Name/text()}</Account_Name>
  <Customer_ID>{$Mapping_Accounts/Customer_ID/text()}</Customer_ID>
  <Stock_ID>{$Mapping_Accounts/Stock_ID/text()}</Stock_ID>
  <Quantity>{$Mapping_Accounts/Quantity/text()}</Quantity>
</Accounts_Instance>


}
</Accounts>
```

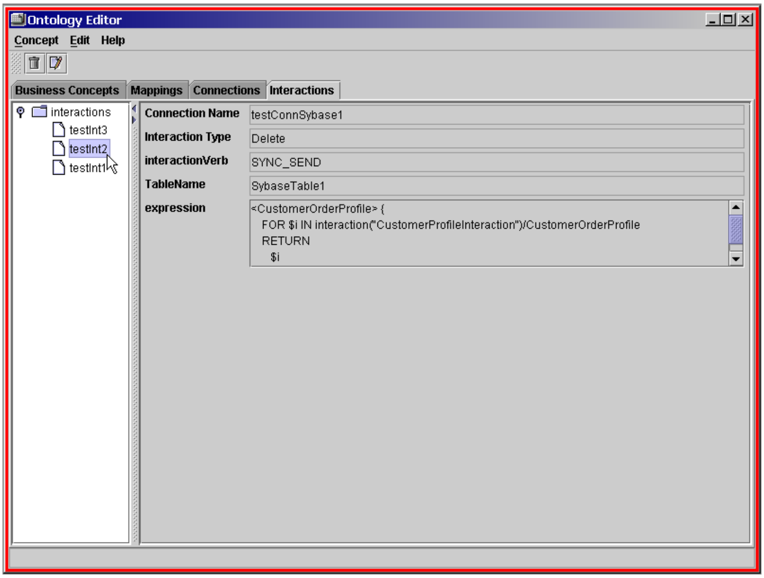
7 The Interactions Editor displays with the details of your modified XQuery displayed within the right-hand panel.

Deleting an Interaction

1 Start the Ontology Editor, if it is not currently open.

2 Select which Interaction to delete.

Figure 8-29. Delete Interaction - Interaction Selected



Connection Name testConnSybase1

Interaction Type Delete

interactionVerb SYNC_SEND

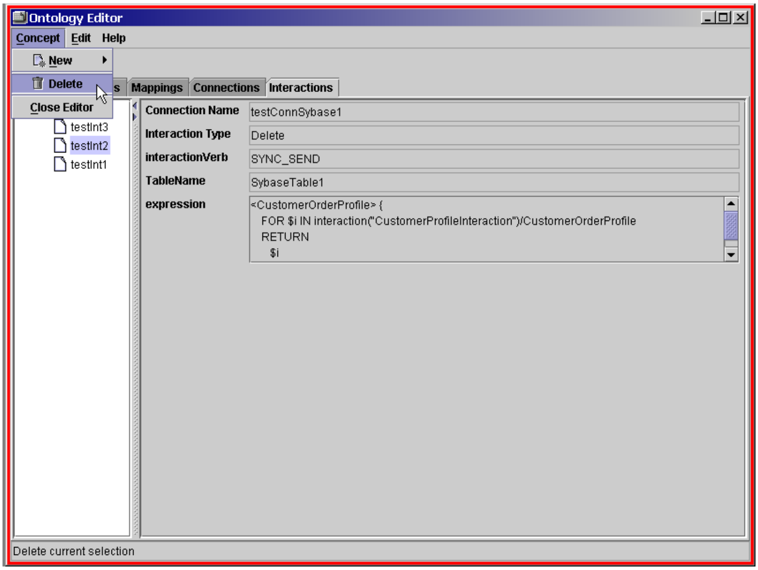
TableName SybaseTable1

expression

```
<CustomerOrderProfile> {
  FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProfile
  RETURN
    $i
```

3 Select **Concept > Delete.**

Figure 8-30. Delete Interaction - Delete Selected



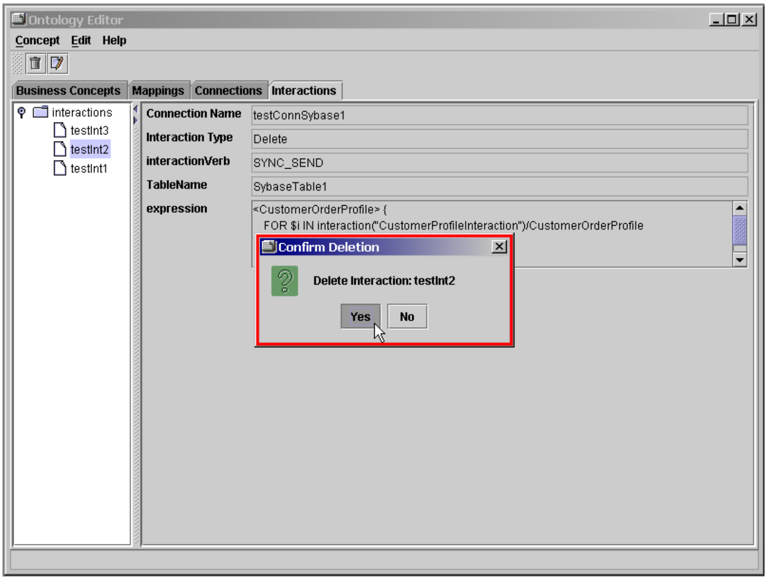
Connection Name	testConnSybase1
Interaction Type	Delete
interactionVerb	SYNC_SEND
TableName	SybaseTable1
expression	

```
<CustomerOrderProfile> {
  FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProfile
  RETURN
    $i
```

Delete current selection

4 A dialog appears, asking you to confirm your choice. Click the **Yes** button to delete, or **No** to cancel the deletion.

Figure 8-31. Delete Interaction - Confirm Delete



Connection Name testConnSybase1

Interaction Type Delete

interactionVerb SYNC_SEND

TableName SybaseTable1

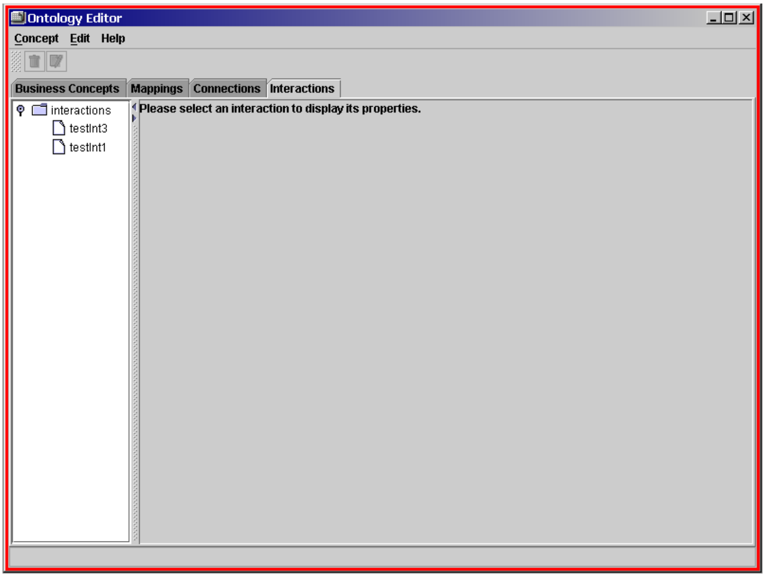
expression

```
<CustomerOrderProfile> {
  FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProfile
```

Delete Interaction: testInt2

5 The Interactions Editor display updates, removing the deleted Interaction.

Figure 8-32. Delete Interaction - Interaction Deleted



Please select an interaction to display its properties.

Chapter

9

Creating Agents and Applications

Agents and Applications enable Business Users to convert their static Business Concepts and models into active business processes. Using Agents and Applications, you can create dynamic models that publish and subscribe to datasources and message feeds, listen for triggers during specific date or time ranges, and perform complex data manipulation and share their results with other Agents or Applications or external datasources or message feeds.

Introducing Applications and Agents

Applications communicate with external datasources, message feeds, and third-party applications. Their data and rowset manipulation abilities are limited.

Applications are composed of one or more "trigger" blocks of Application Language script, and may also contain agents.

Agents are composed of sequentially executed Agent Language scripts. Agents can "nest" other Agents, that is, invoke other Agents from within their own code. Where Applications are high-level, Agents deliver low-level functionality. Agents feature a powerful set of data and rowset manipulation commands.

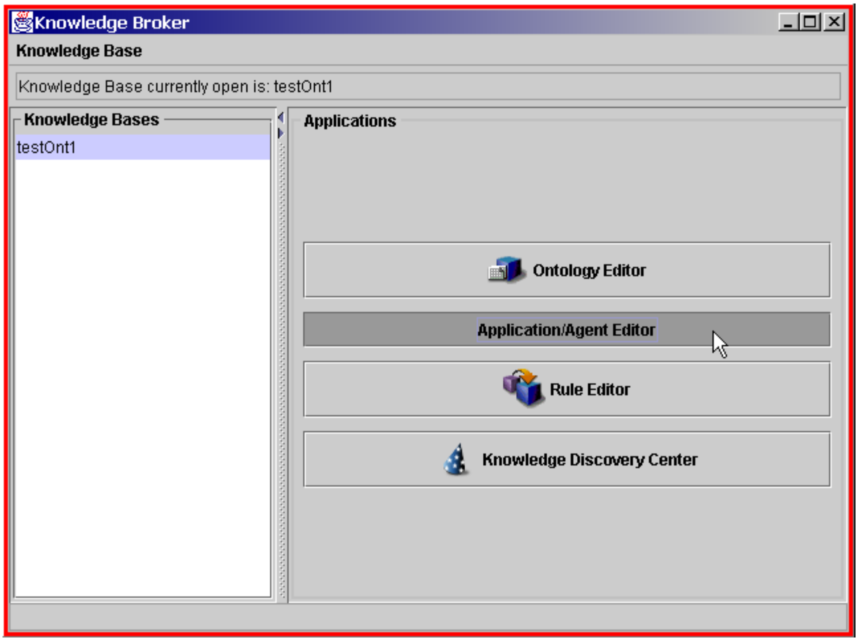
Creating and Deleting Applications

Creating an Application

You create Applications using the Application/Agent Editor.

1 Ensure you have opened a Knowledge Base. From the Launcher, highlight the Application/Agent Editor button and click or press the Space Bar to select it.

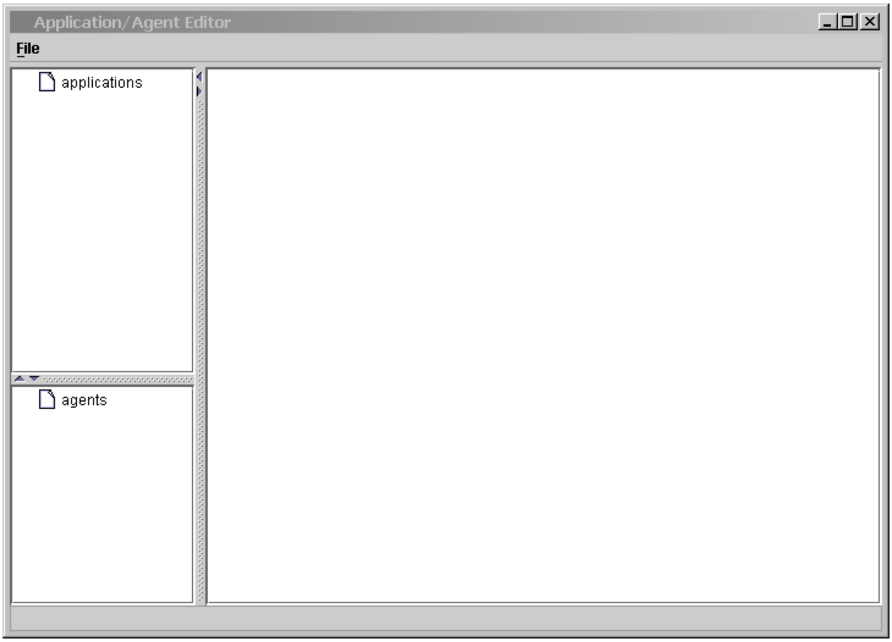
Figure 9-1. Starting the Application Agent Editor



Knowledge Base currently open is: testOnt1

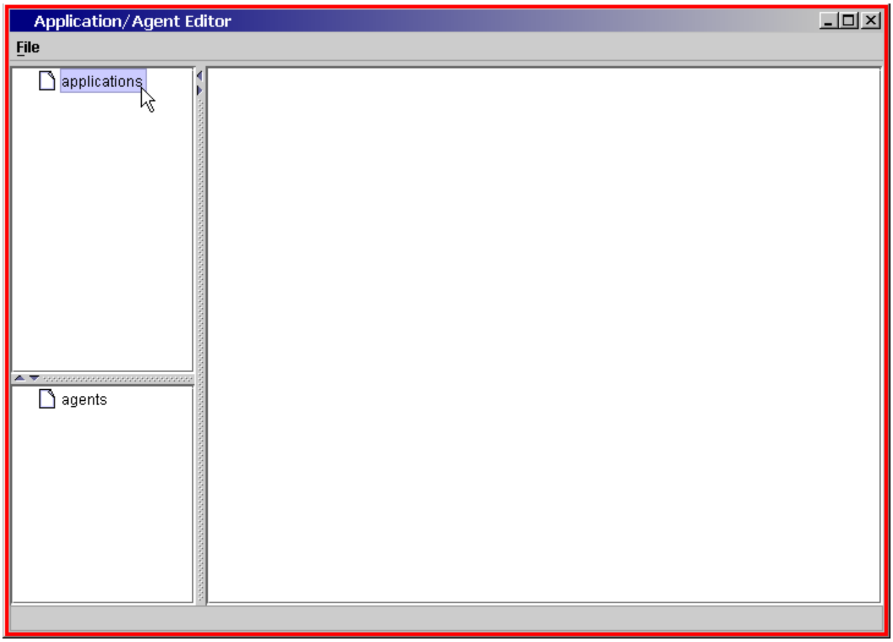
2 The Application/Agent Editor displays. Initially, it displays no entries because no Applications or Agents have been created.

Figure 9-2. Application/Agent Editor Initial Display



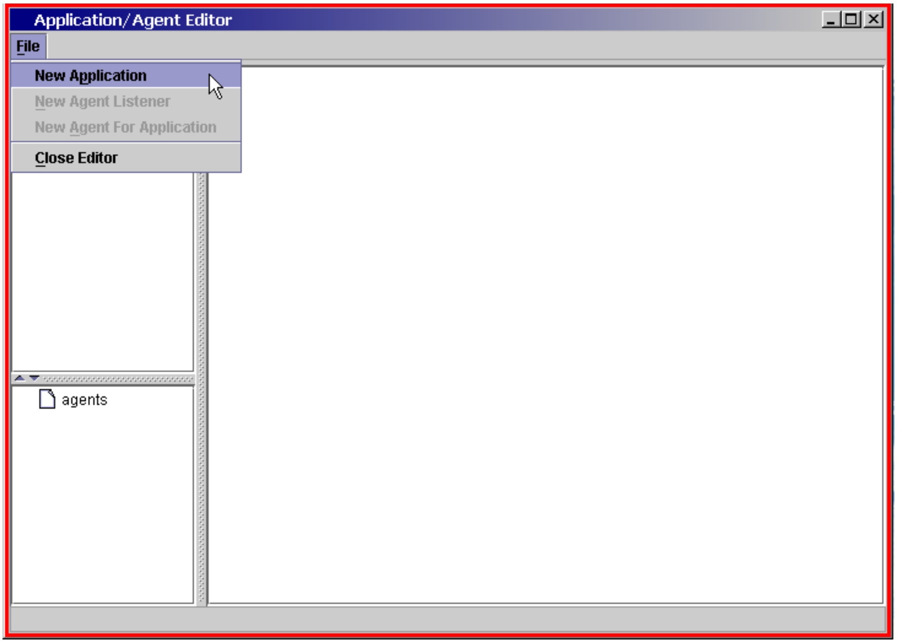
3 Use the cursor to highlight “application” in the upper-left panel.

Figure 9-3. Application/Agent Editor - application Selected



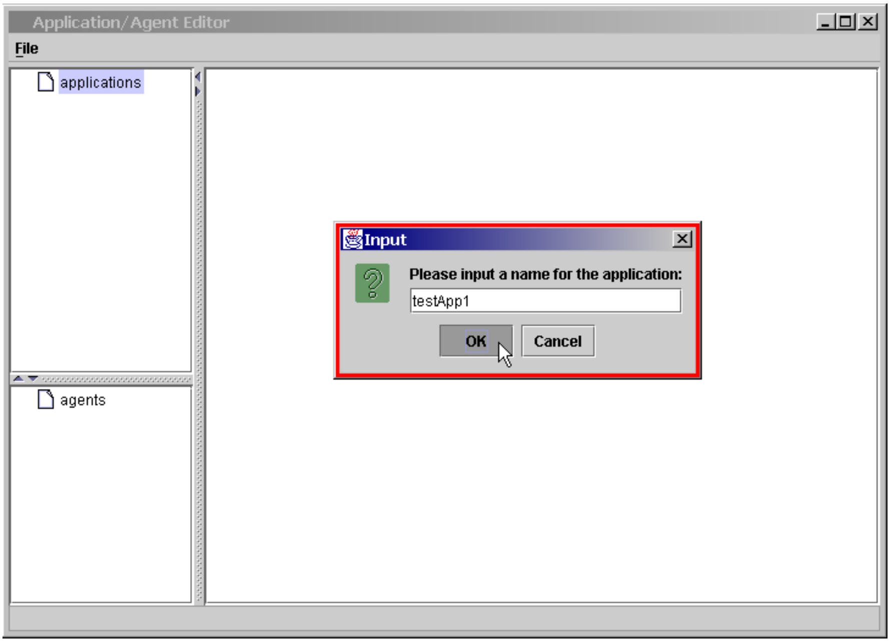
4 Select **File > New Application** (or use option-click).

Figure 9-4. Application/Agent Editor - New Application Selected



5 A dialog displays requesting you to name your new Application. Type your desired name and select the OK button.

Figure 9-5. Application/Agent Editor - Naming the New Application

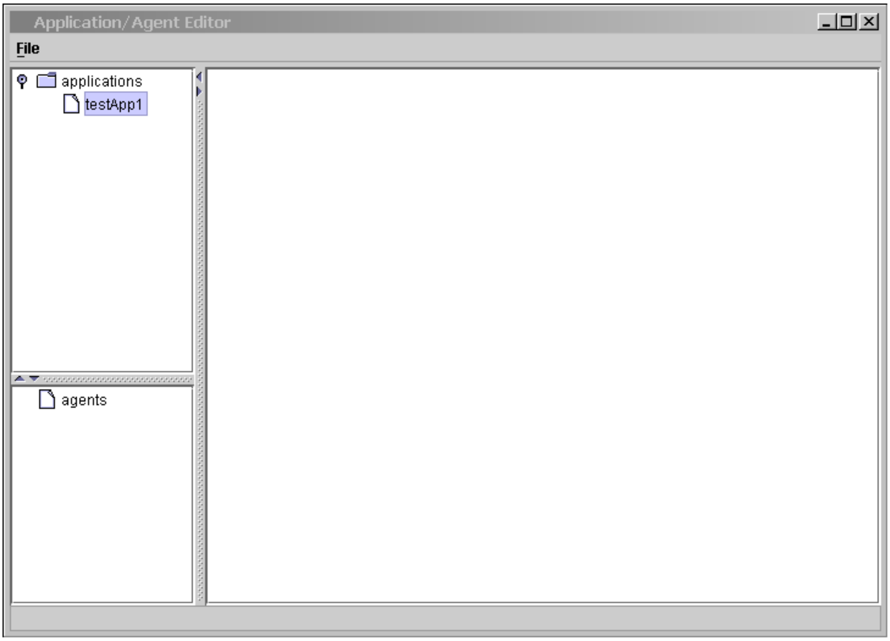


Please input a name for the application:

testApp1

6 The Application/Agent Editor now displays the new, blank Application.

Figure 9-6. Application/Agent Editor - New Application Created



Deleting an Application

1 Open the Application/Agent Editor, if it is not currently open.

2 Select the desired Application.

3 **Option Click > Delete**. You have now deleted an Application.

Triggering Applications

Applications are distinguished by the type of "trigger" Block that executes. Blocks of Application functionality can be set to listen for an event during a specific time period, to simply execute during a specific time period, or to simply execute on demand. The three types of Blocks are:

1 Listen

2 Start Process

3 Start Process on Demand

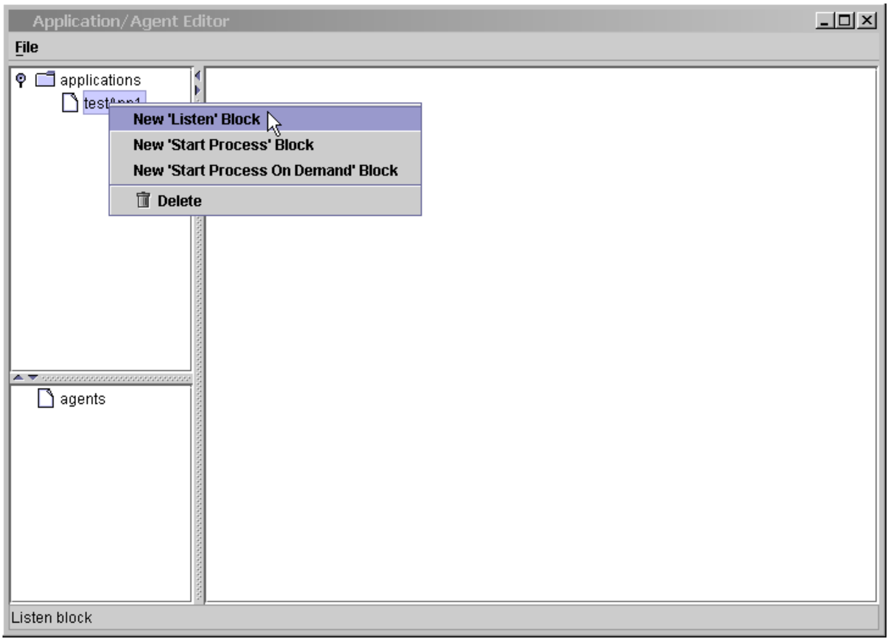
Each Application Block can contain multiple "rows" of Application Block code elements. Application Blocks execute these rows sequentially from the start row (top) to the end row (bottom).

Adding a Listen Block

This trigger listens for an Event Business Concept according to a user-defined schedule. When both of these conditions are satisfied, it executes.

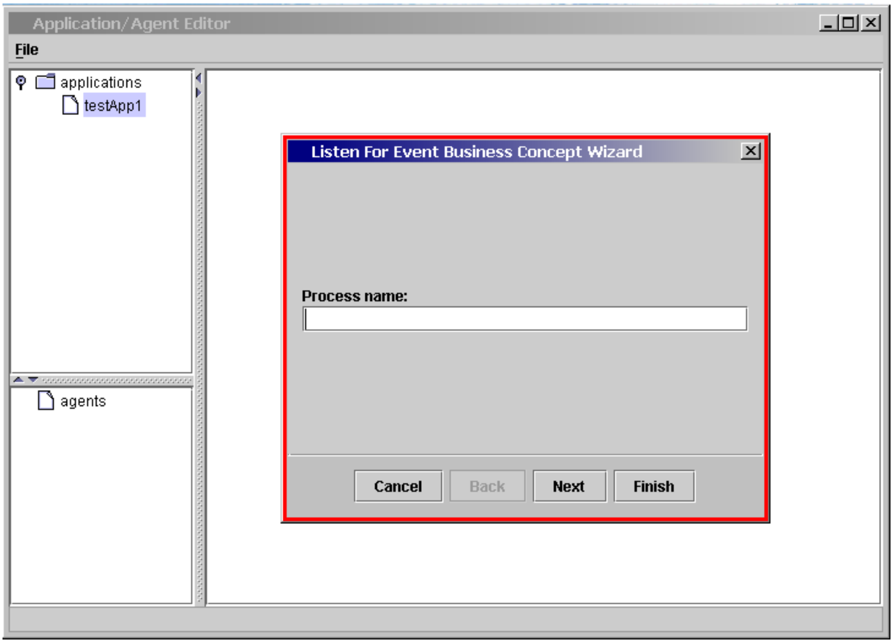
1 Open the Application/Agent Editor. Select the Application to which you want to add a Listen Block and option-click. The context menu displays.

Figure 9-7. Application/Agent Editor - Listen Block - Adding



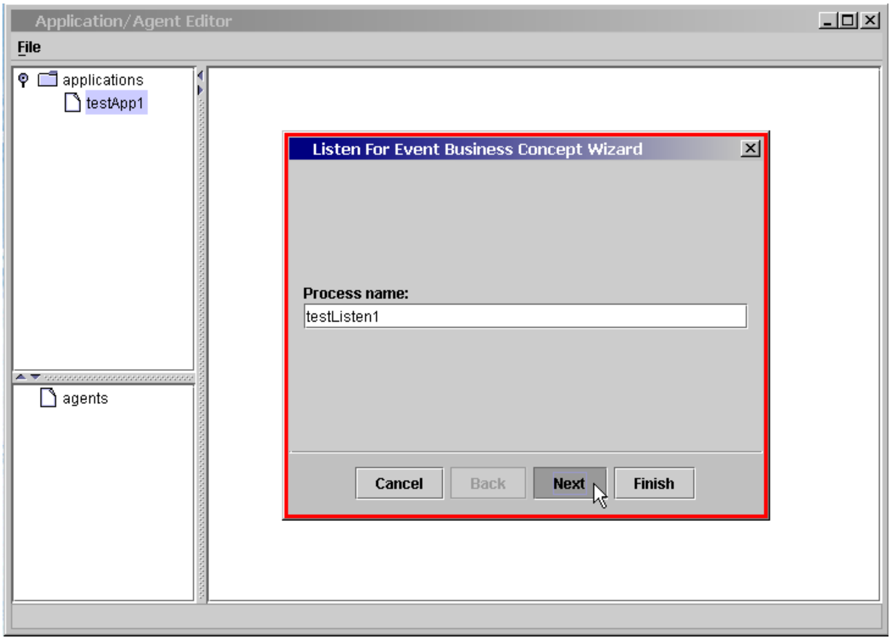
2 Select the New ‘Listen’ Block. The Listen for Event Business Concept Wizard displays.

Figure 9-8. Application/Agent Editor - Listen for Event Business Concept Wizard Displays



3 Type a name for the Listen Block and click the Next button to continue.

Figure 9-9. Listen for Event Business Concept Wizard - Naming the Listen Block

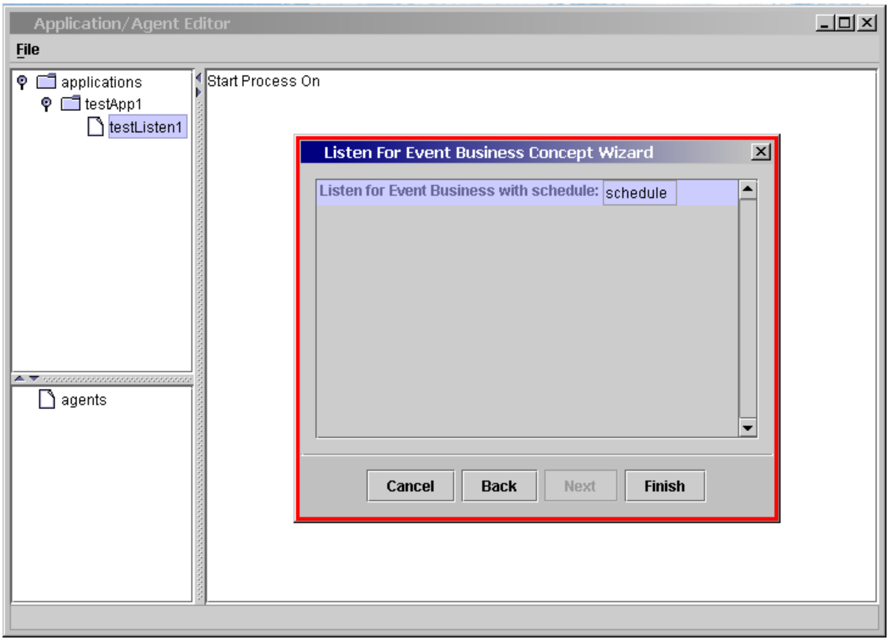


Process name:
testListen1

4 The Wizard now displays the first line of the Listen Block:

```
Listen for Event Business with schedule:schedule.
```

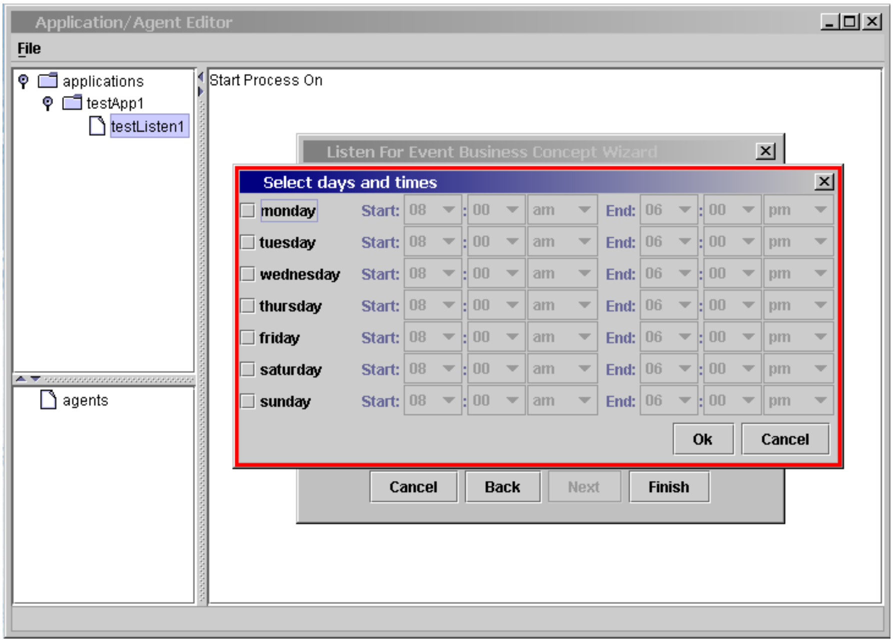
Figure 9-10. Listen for Event Business Concept Wizard - Initial Schedule Block Code



Listen for Event Business with schedule: schedule

5 Select the **`schedule`** pop-up text. A daily schedule displays.

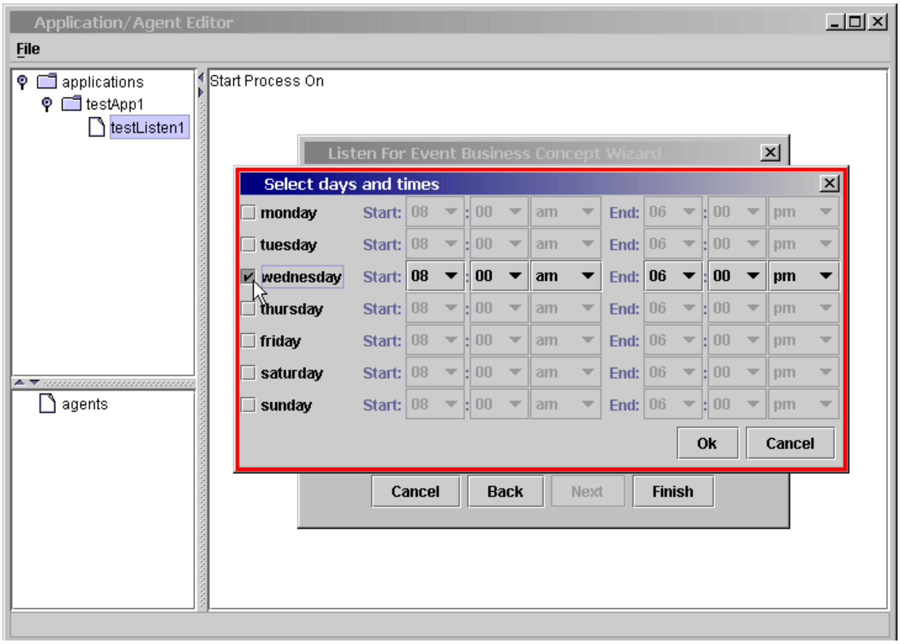
Figure 9-11. Listen for Event Business Concept Wizard - Schedule Popup



□ monday	Start:	08	00	am	End:	06	00	pm
□ tuesday	Start:	08	00	am	End:	06	00	pm
□ wednesday	Start:	08	00	am	End:	06	00	pm
□ thursday	Start:	08	00	am	End:	06	00	pm
□ friday	Start:	08	00	am	End:	06	00	pm
□ saturday	Start:	08	00	am	End:	06	00	pm
□ sunday	Start:	08	00	am	End:	06	00	pm

6 Use the cursor to place a checkmark next to a day of the week for which you want the Listen Block to be active. That row highlights.

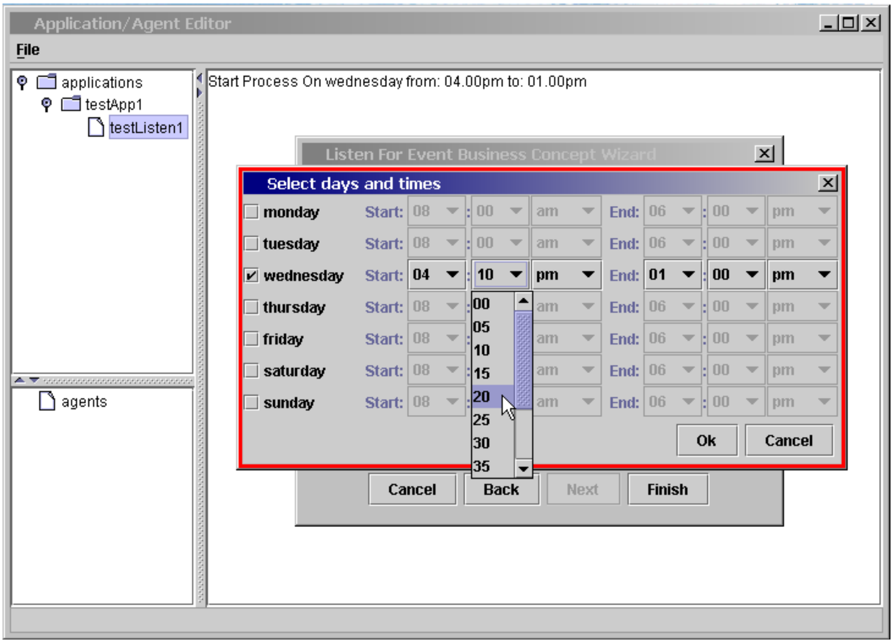
Figure 9-12. Listen for Event Business Concept Wizard - Day Selected



□ monday	Start:	08	00	am	End:	06	00	pm
□ tuesday	Start:	08	00	am	End:	06	00	pm
☑ wednesday	Start:	08	00	am	End:	06	00	pm
□ thursday	Start:	08	00	am	End:	06	00	pm
□ friday	Start:	08	00	am	End:	06	00	pm
□ saturday	Start:	08	00	am	End:	06	00	pm
□ sunday	Start:	08	00	am	End:	06	00	pm

7 Use the cursor to specify a Start time for your Listen Block.

Figure 9-13. Listen for Event Business Concept Wizard - Start Time Selected

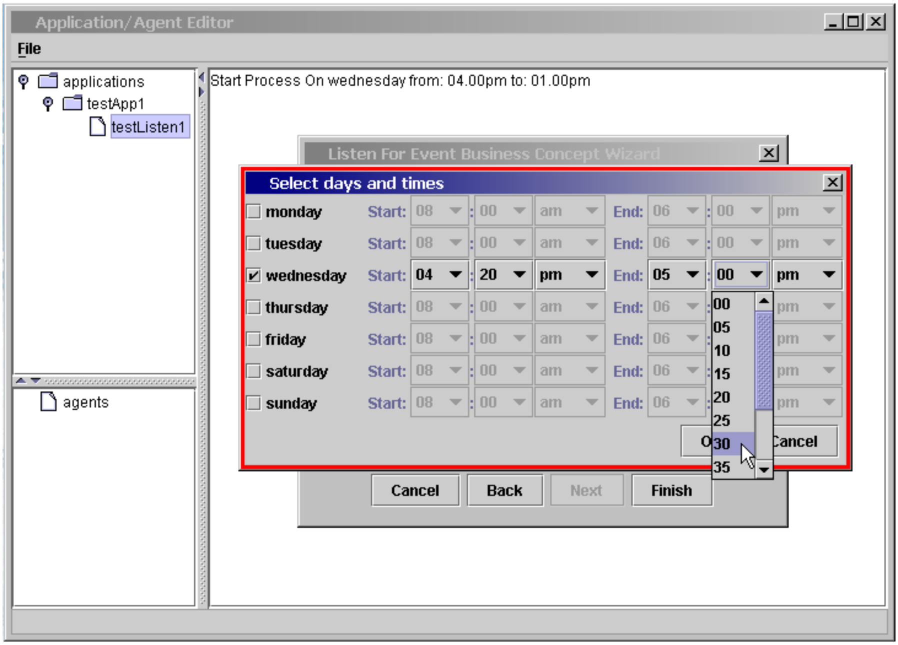


Start Process On wednesday from: 04.00pm to: 01.00pm

☐ monday	Start:	08	00	am	End:	06	00	pm
☐ tuesday	Start:	08	00	am	End:	06	00	pm
☑ wednesday	Start:	04	10	pm	End:	01	00	pm
☐ thursday	Start:	08		am	End:	06	00	pm
☐ friday	Start:	08		am	End:	06	00	pm
☐ saturday	Start:	08		am	End:	06	00	pm
☐ sunday	Start:	08		am	End:	06	00	pm

8 Use the cursor to specify an End time for your Listen Block.

Figure 9-14. Listen for Event Business Concept Wizard - End Time Selected



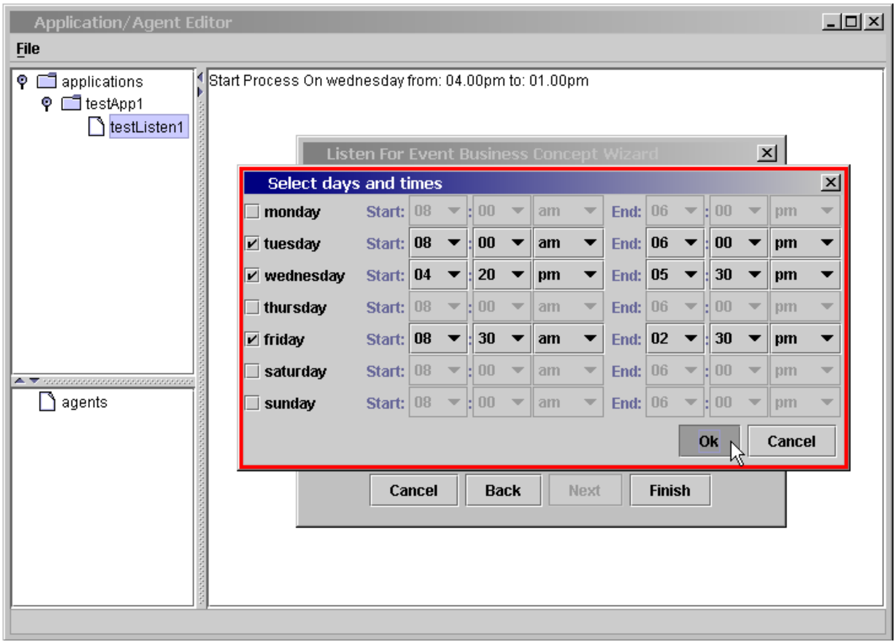
Start Process On wednesday from: 04.00pm to: 01.00pm

☐ monday	Start:	08	00	am	End:	06	00	pm
☐ tuesday	Start:	08	00	am	End:	06	00	pm
☑ wednesday	Start:	04	20	pm	End:	05	00	pm
☐ thursday	Start:	08	00	am	End:	06		pm
☐ friday	Start:	08	00	am	End:	06		pm
☐ saturday	Start:	08	00	am	End:	06		pm
☐ sunday	Start:	08	00	am	End:	06		pm

For the marked day, the Listen Block will activate only during the selected time period.

9 You can select more days and time periods using a similar procedure. When you have finished defining your Listen Schedule, click the Ok button.

Figure 9-15. Listen for Event Business Concept Wizard - Schedule Completed

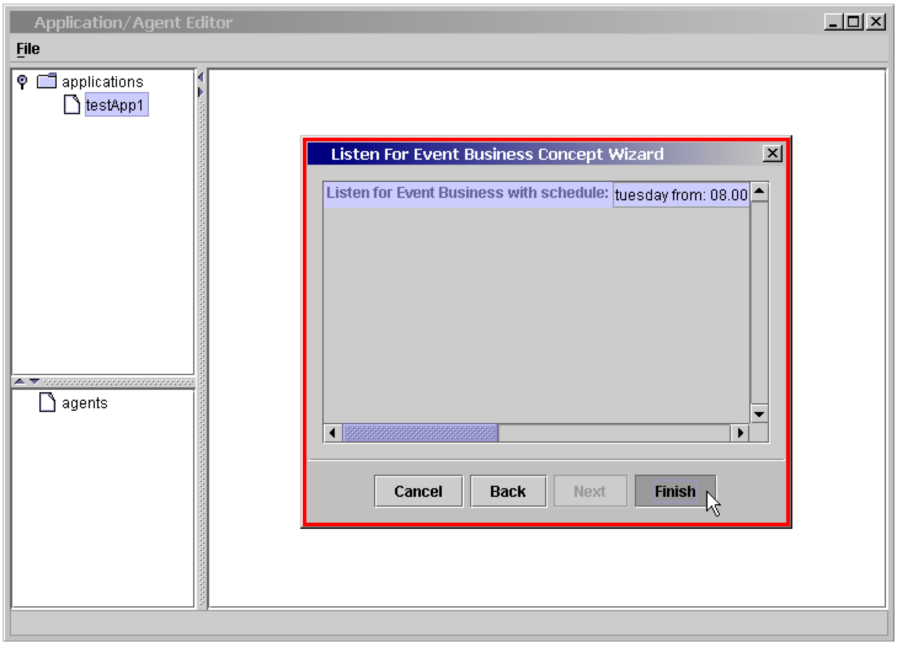


Start Process On wednesday from: 04.00pm to: 01.00pm

☐	monday	Start:	08	:	00	am	End:	06	:	00	pm
☑	tuesday	Start:	08	:	00	am	End:	06	:	00	pm
☑	wednesday	Start:	04	:	20	pm	End:	05	:	30	pm
☐	thursday	Start:	08	:	00	am	End:	06	:	00	pm
☑	friday	Start:	08	:	30	am	End:	02	:	30	pm
☐	saturday	Start:	08	:	00	am	End:	06	:	00	pm
☐	sunday	Start:	08	:	00	am	End:	06	:	00	pm

10 The Listen For Event Business Concept Wizard displays. Your Listen Block now has a schedule. Click the Finish button.

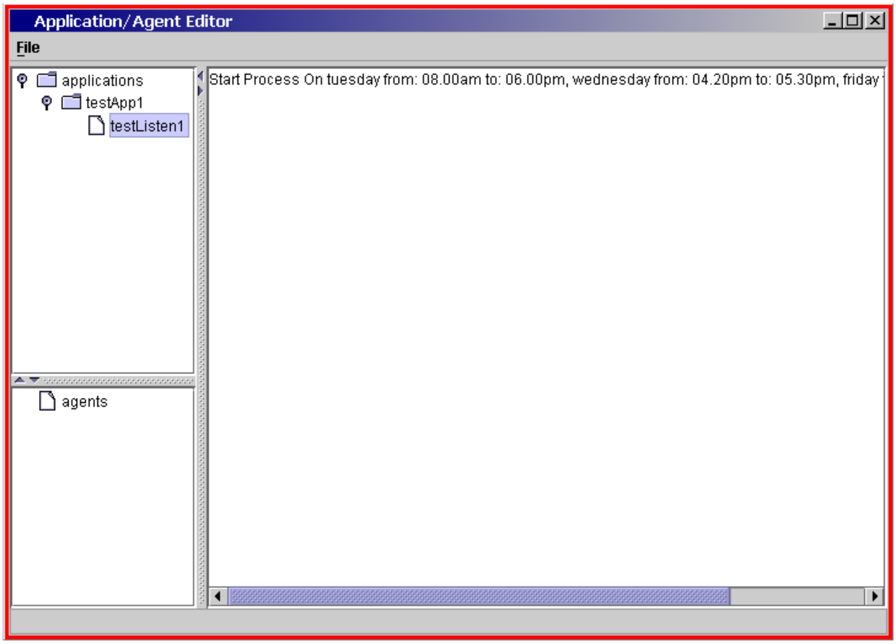
Figure 9-16. Listen for Event Business Concept Wizard - Schedule Displayed



Listen for Event Business with schedule: tuesday from: 08.00

11 The Application/Agent Editor displays. The Application Block you edited remains selected. The right-hand panel now displays the Listen Block code, including your defined schedule.

Figure 9-17. Application/Agent Editor - Listen Block Added



Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

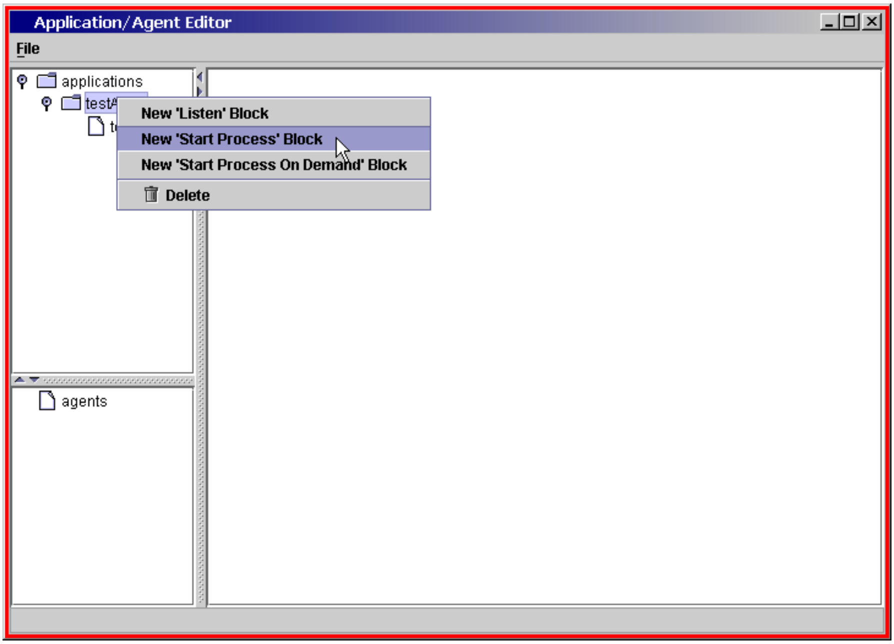
You have successfully added a new Listen Block.

Adding a Start Process Block

This trigger executes during a specific user-defined schedule. It does not list for an Event Business Concept.

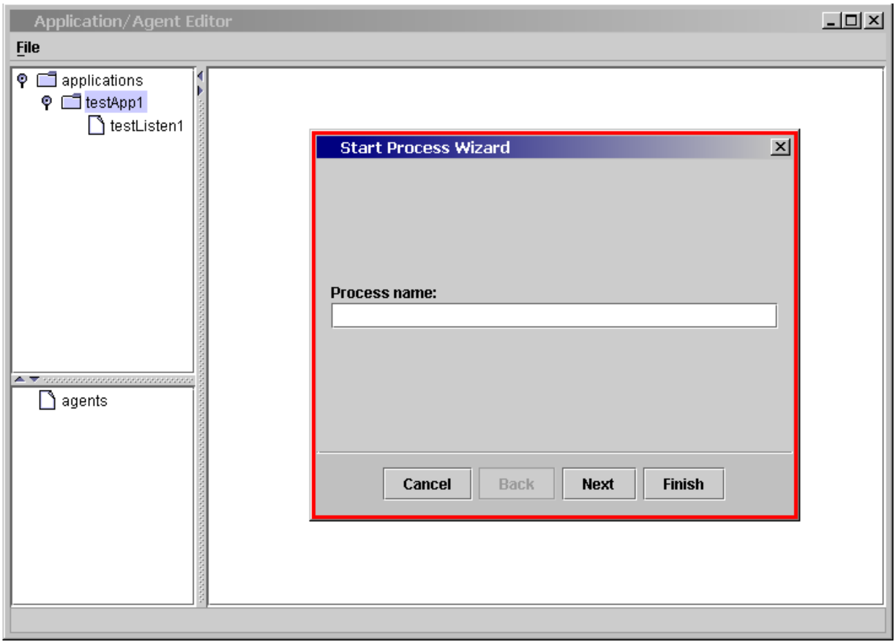
1 Open the Application/Agent Editor. Select the Application to which you want to add a Listen Block and option-click. The context menu displays.

Figure 9-18. Application/Agent Editor - Start Process Block - Adding



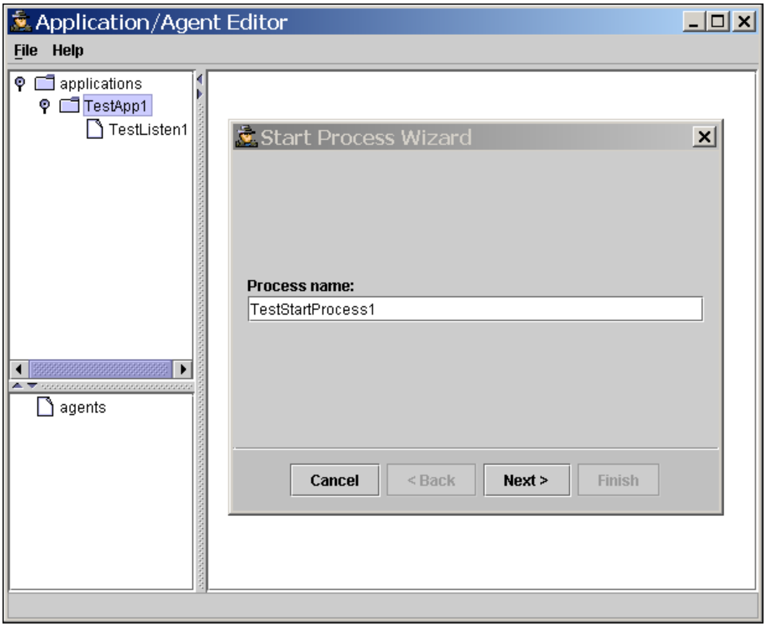
2 Select the New 'Start Process' Block. The Start Process Wizard displays.

Figure 9-19. Application/Agent Editor - Start Process Wizard Displays



3 Type a name for the Start Process Block and click the Next button to continue.

Figure 9-20. Start Process Wizard - Naming the Start Process Block

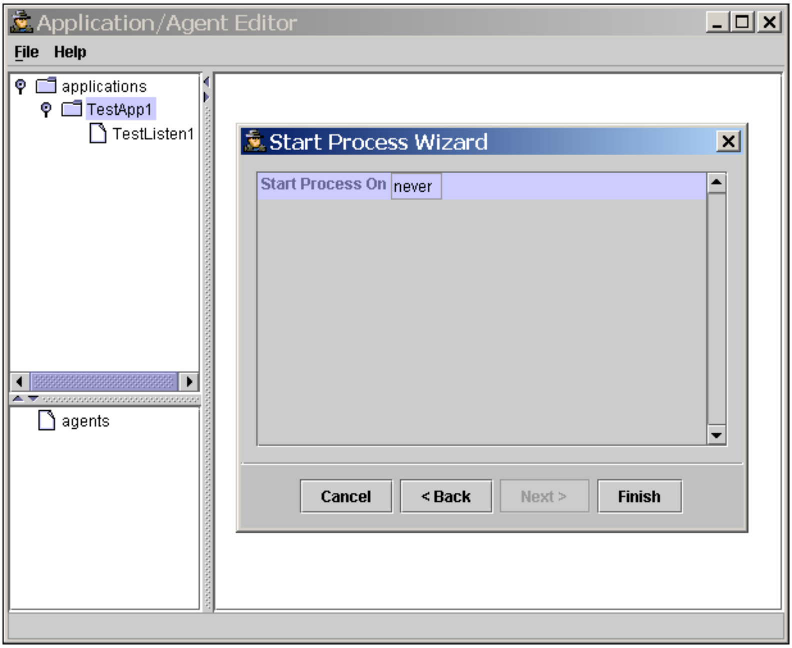


Process name: TestStartProcess1

4 The Wizard now displays the first line of the Start Process Block:

```
Start Process On:never.
```

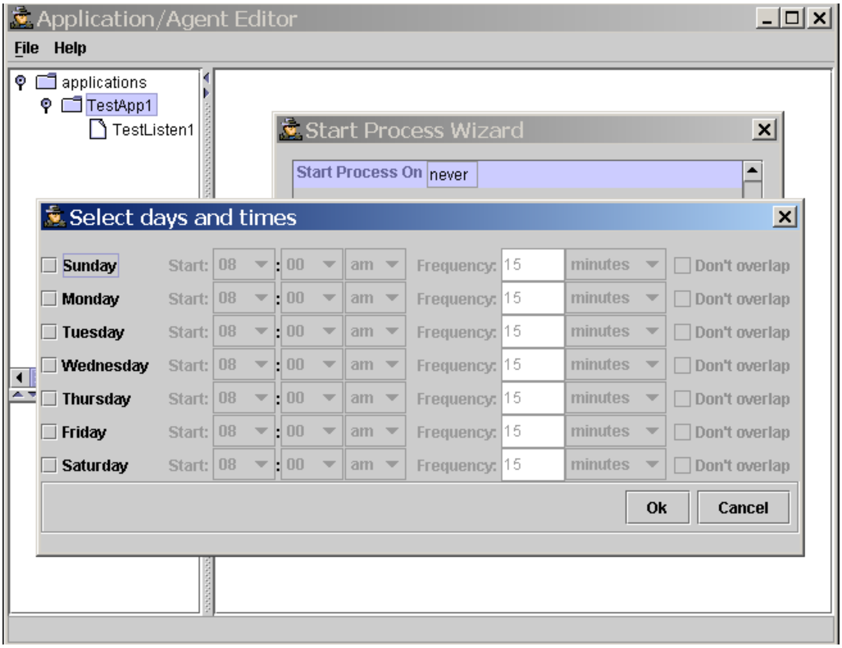
Figure 9-21. Start Process Wizard - Initial Schedule Block Code



Start Process On never

5 Select the schedule pop-up text. A daily schedule displays.

Figure 9-22. Start Process Wizard - Schedule Popup

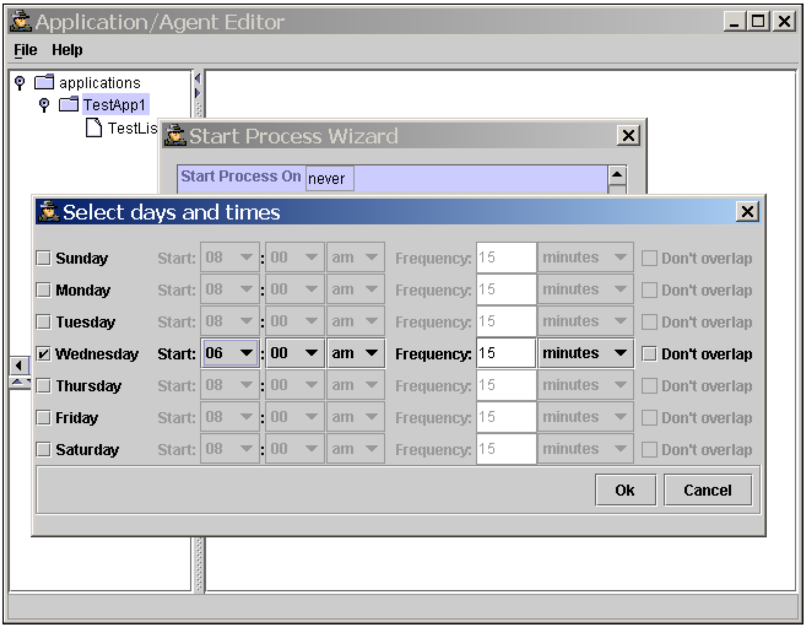


Start Process On never

□ Sunday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Monday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Tuesday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Wednesday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Thursday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Friday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Saturday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap

6 Use the cursor to place a checkmark next to a day of the week for which you want the Start Process Block to be active. That row highlights.

Figure 9-23. Start Process Wizard - Day Selected

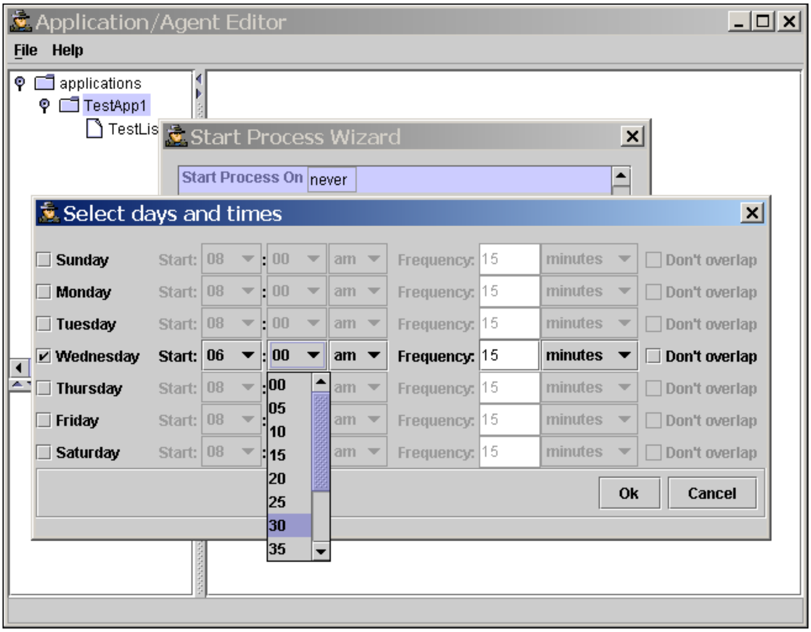


Start Process On never

□ Sunday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Monday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Tuesday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
☑ Wednesday	Start:	06	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Thursday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Friday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap
□ Saturday	Start:	08	: 00	am	Frequency: 15	minutes	□ Don't overlap

7 Use the cursor to specify a Start time for your Start Process Block.

Figure 9-24. Start Process Wizard - Start Time Selected

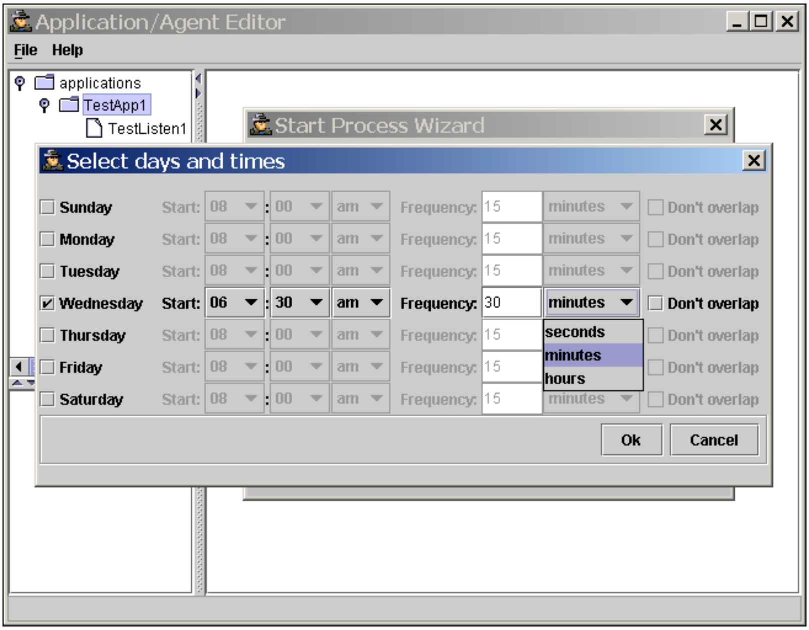


Start Process On: never

☐ Sunday	Start:	08	:	00	am	Frequency:	15	minutes	☐ Don't overlap
☐ Monday	Start:	08	:	00	am	Frequency:	15	minutes	☐ Don't overlap
☐ Tuesday	Start:	08	:	00	am	Frequency:	15	minutes	☐ Don't overlap
☑ Wednesday	Start:	06	:	00	am	Frequency:	15	minutes	☐ Don't overlap
☐ Thursday	Start:	08	:		am	Frequency:	15	minutes	☐ Don't overlap
☐ Friday	Start:	08	:		am	Frequency:	15	minutes	☐ Don't overlap
☐ Saturday	Start:	08	:		am	Frequency:	15	minutes	☐ Don't overlap

8 Use the cursor to specify a frequency for your Start Process Block.

Figure 9-25. Start Process Wizard - Frequency Selected



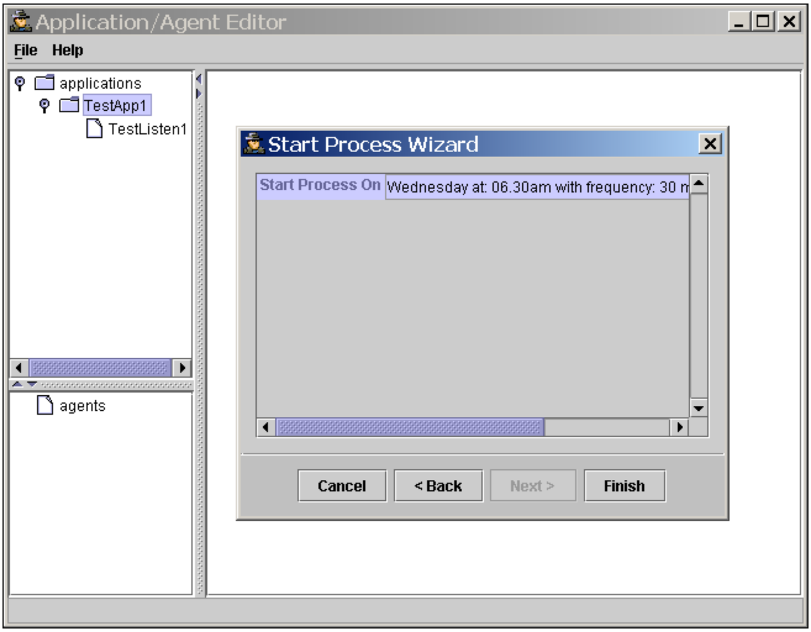
☐ Sunday	Start:	08	: 00	am	Frequency:	15	minutes	☐ Don't overlap
☐ Monday	Start:	08	: 00	am	Frequency:	15	minutes	☐ Don't overlap
☐ Tuesday	Start:	08	: 00	am	Frequency:	15	minutes	☐ Don't overlap
☑ Wednesday	Start:	06	: 30	am	Frequency:	30	minutes	☐ Don't overlap
☐ Thursday	Start:	08	: 00	am	Frequency:	15		☐ Don't overlap
☐ Friday	Start:	08	: 00	am	Frequency:	15		☐ Don't overlap
☐ Saturday	Start:	08	: 00	am	Frequency:	15	minutes	☐ Don't overlap

For the marked day, the Start Process Block will activate only during the specified time periods.

9 You can select more days and frequencies using a similar procedure. When you have finished defining your Start Process Schedule, click the OK button.

10 The Start Process Wizard displays. Your New Start Process now has a schedule. Click the Finish button.

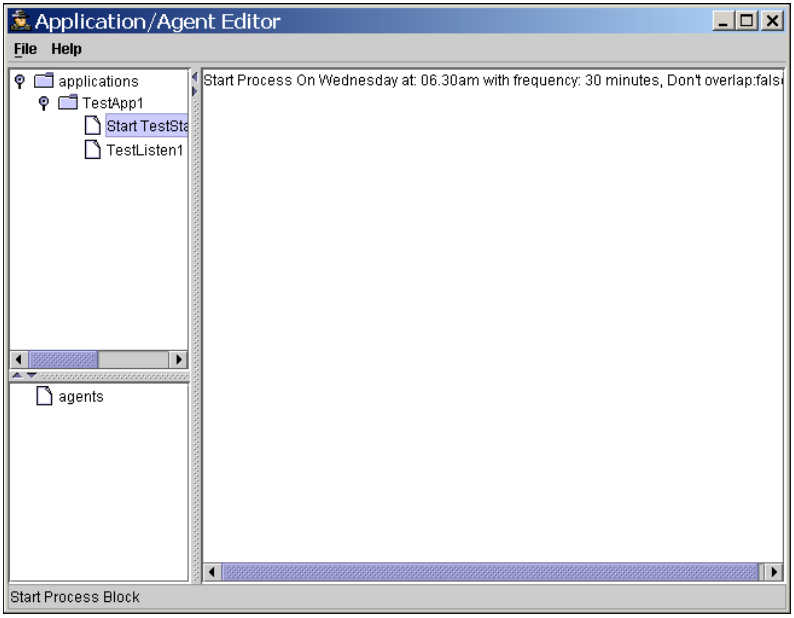
Figure 9-26. Start Process Wizard - Schedule Displayed



Start Process On Wednesday at: 06.30am with frequency: 30 m

11 The Application/Agent Editor displays. The Application Block you edited remains selected. The right-hand panel now displays the Start Process Block code, including your defined schedule.

Figure 9-27. Application/Agent Editor - Start Process Added



Start Process On Wednesday at: 06.30am with frequency: 30 minutes, Don't overlap:fals

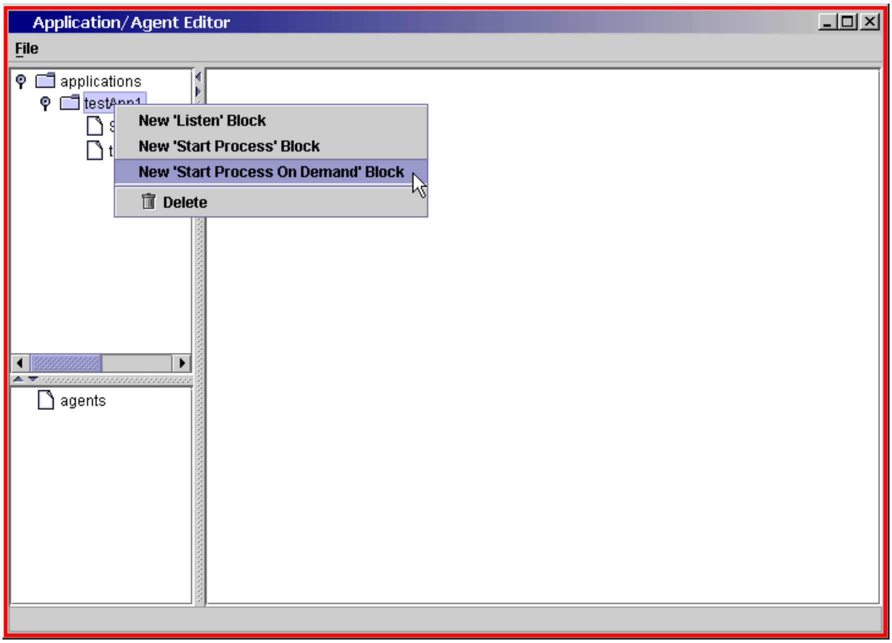
You have successfully added a new Start Process Block.

Adding a Start Process on Demand Block

This trigger executes on demand, without any scheduling constraints or Event Business Concept preconditions.

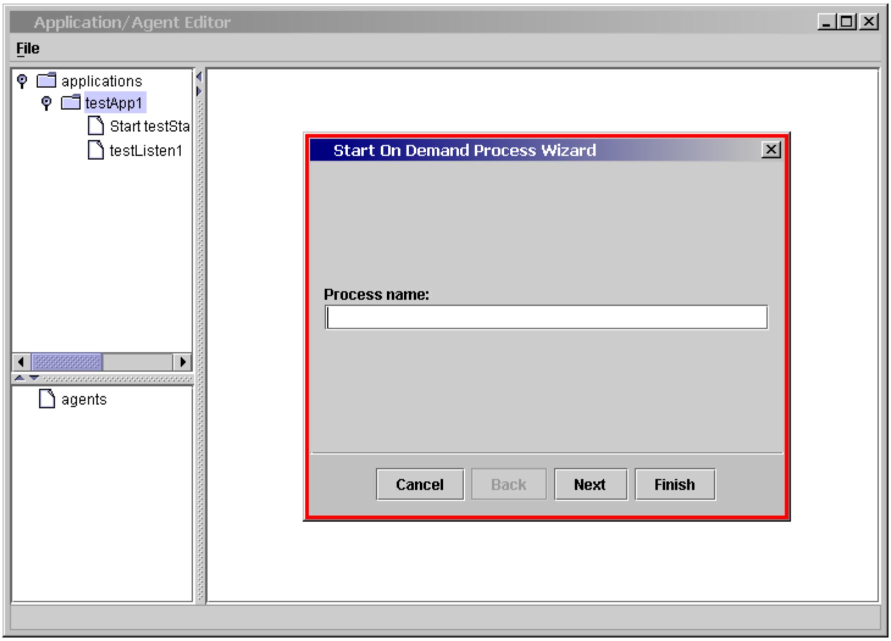
1 Open the Application/Agent Editor. Select the Application to which you want to add a Start Process on Demand and option-click. The context menu displays.

Figure 9-28. Start Process On Demand Block - Adding



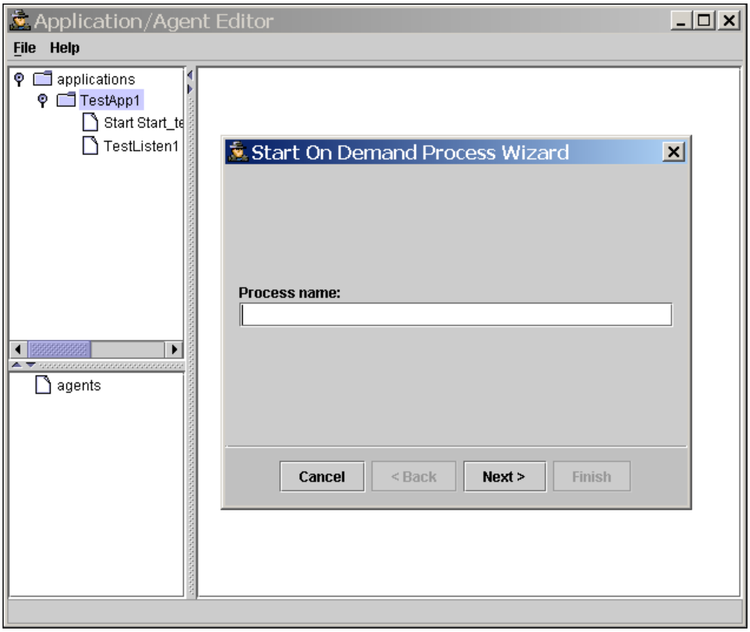
2 Select the New ‘Start Process’ Block. The Start Process Wizard displays.

Figure 9-29. Start Process On Demand Wizard Displays



3 Type a name for the Start Process Block and Click the Next button to continue.

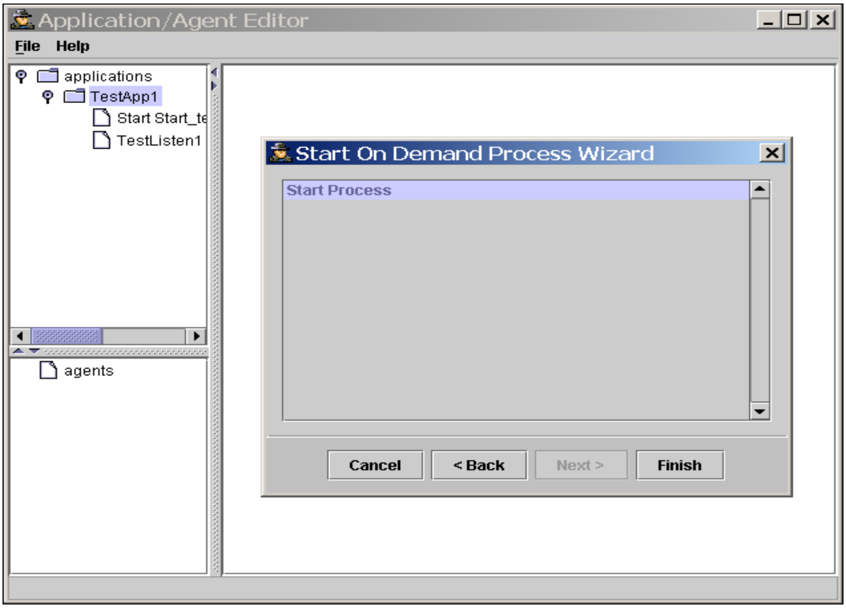
Figure 9-30. Start Process on Demand Wizard - Naming



Process name:

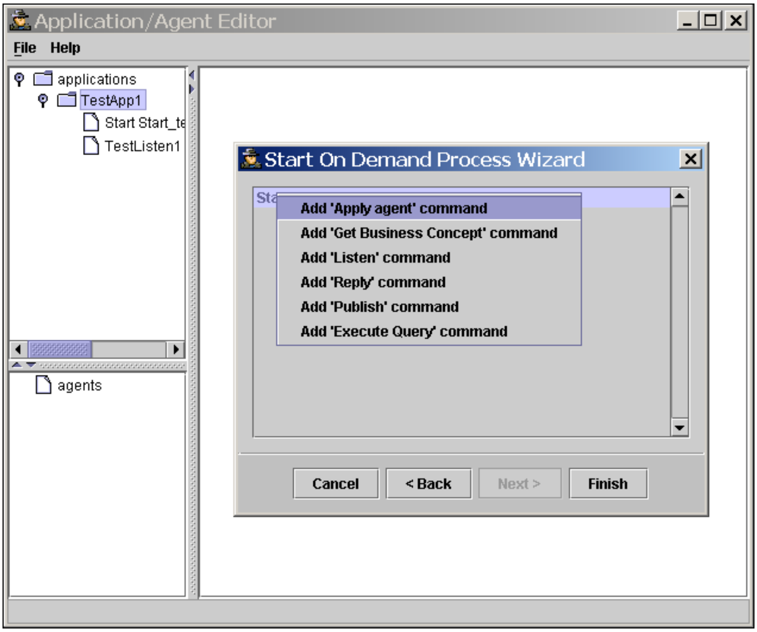
4 The Application Block Code Element Editor displays.

Figure 9-31. Start Process on Demand Wizard - Element Editor



Any pre-existing Application Block code elements display. The active element row displays highlighted. Decide which row you want to be immediately above your new Application Block code element. Option-click on this row to display the context menu.

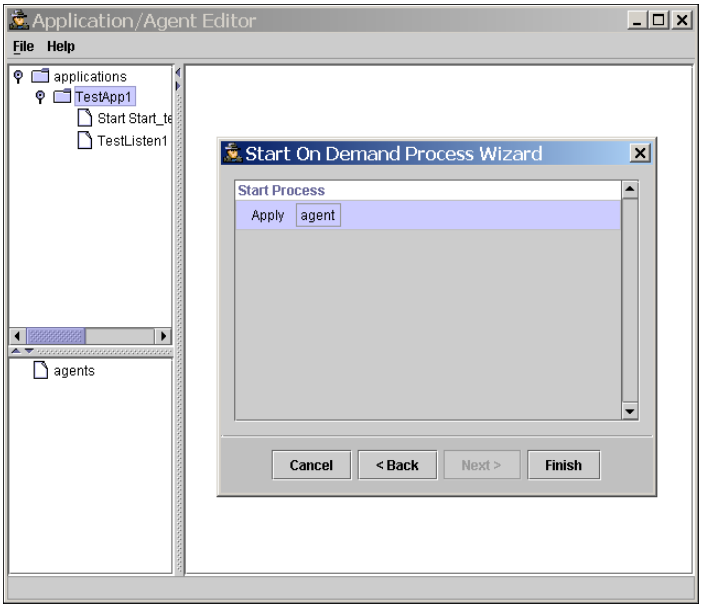
Figure 9-32. Start Process on Demand Wizard - Element Editor2



5 Use the cursor to select which type of Application Block element you want and click or option-click to add your choice.

6 The new Application Block element appears directly below whichever Application Block code element was active when you option-clicked in Step 5.

Figure 9-33. Start Process on Demand - New Application Block Code Element Appears Below



Start Process

Apply agent

To add as many Application Block code elements as required, repeat this option-click context menu action.

7 When you have finished adding Application Block Code, select the Finish button.

8 The Application/Agent Editor displays. The Application Block you modified remains selected. The right-hand panel now displays the new Application Block code.

You have now finished adding a new Start On Demand Process Block.

Modifying Applications

The Application Wizards enable you to add or modify Application Block code for specific Applications. There are five Application Block code elements:

1 Apply Agent

2 Get Business Concept

3 Listen

4 Reply

5 Publish

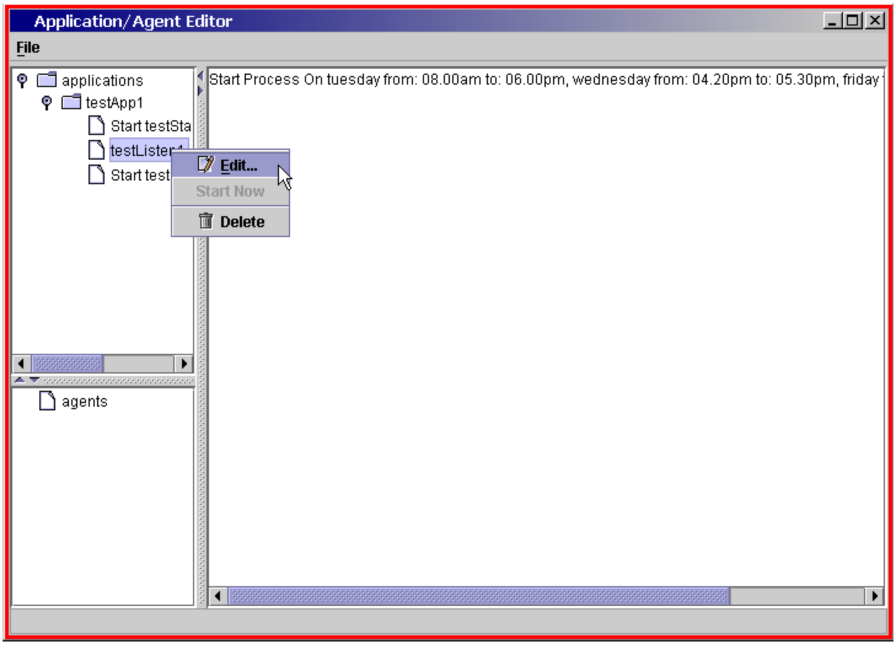
These elements enable you to add or modify Application triggers, invoke Agents, and push information out to or reply to external datasources.

Adding Application Block Code

For all Application Block code elements, the procedure is similar:

1 Open the Application/Agent Editor. Select the Application Block to which you want to add an Application Block code element and option-click. The context menu displays.

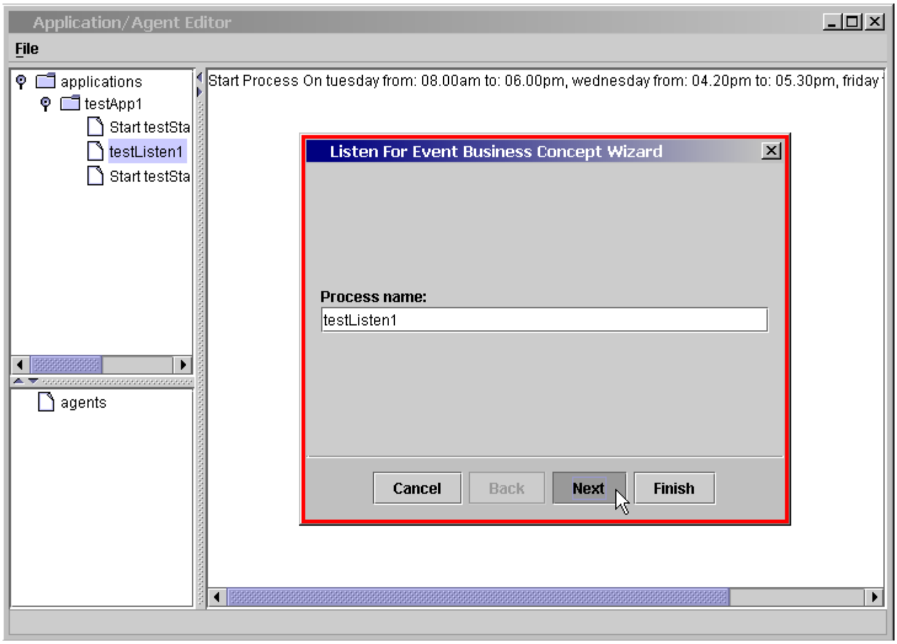
Figure 9-34. Application/Agent Editor - Adding Block Code Element - Edit Function



Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

2 Select the Edit... option. The appropriate Application Wizard displays. Select the Next button.

Figure 9-35. Adding Application Block Code - Appropriate Wizard Displays



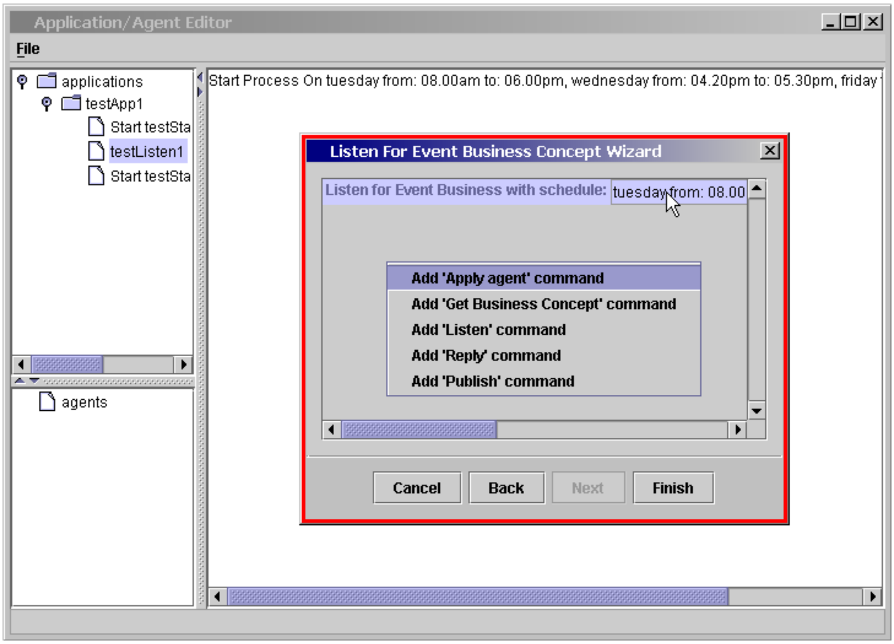
Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

Process name: testListen1

3 The Application Block Code Element Editor displays. Any pre-existing Application Block code elements display. The active element row displays highlighted. Decide

which row you want to be immediately above your new Application Block code element. Option-click on this row to display the context menu.

Figure 9-36. Adding Application Block Code - Popup Context Menu Displayed

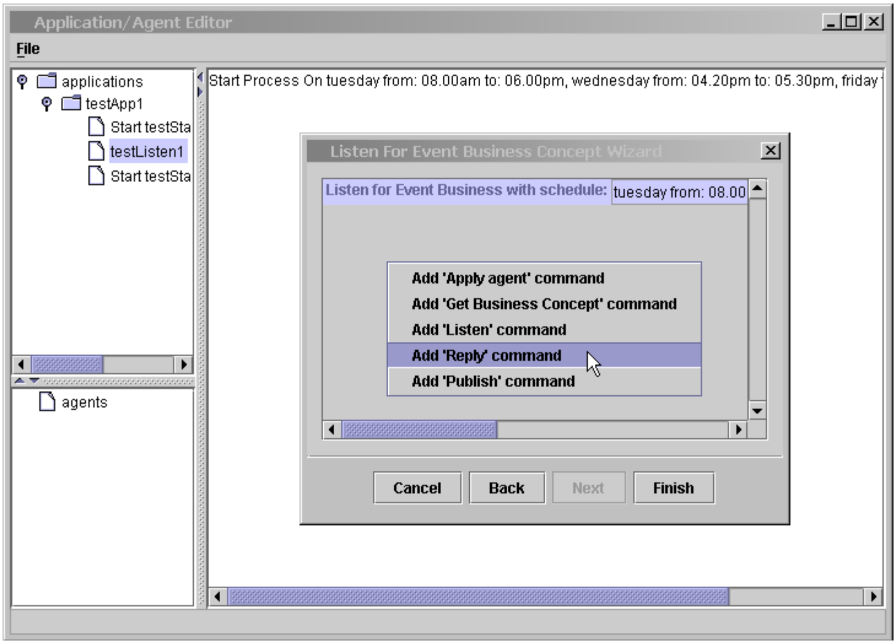


Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

Listen for Event Business with schedule: tuesday from: 08.00

4 Use the cursor to select which type of Application Block element you want and click or option-click to add your choice.

Figure 9-37. Adding Application Block Code - Desired Block Code Element Selected

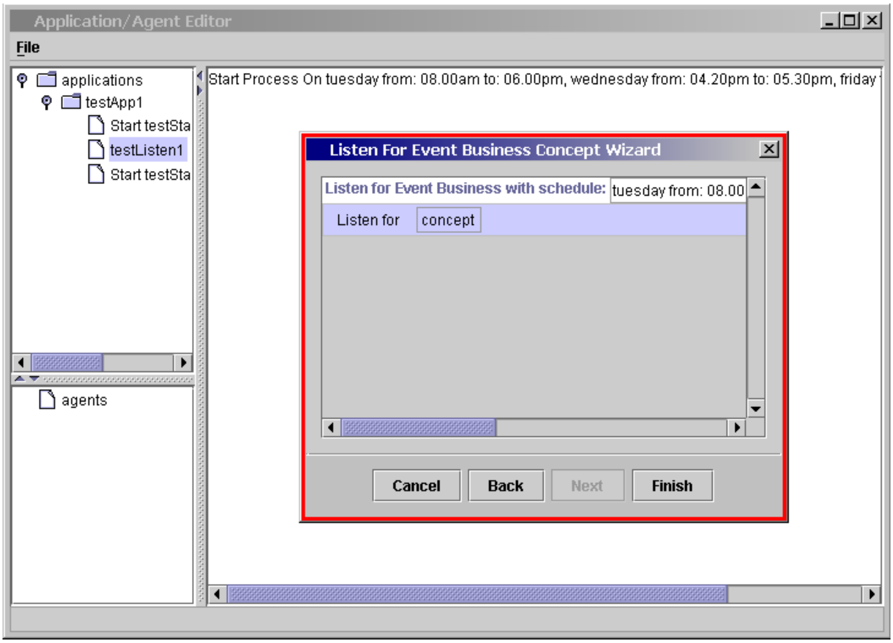


Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

Listen for Event Business with schedule: tuesday from: 08.00

5 The new Application Block element appears directly below whichever Application Block code element was active when you option-clicked in Step 3.

Figure 9-38. Adding Application Block Code - New Application Block Code Element Appears Below



Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

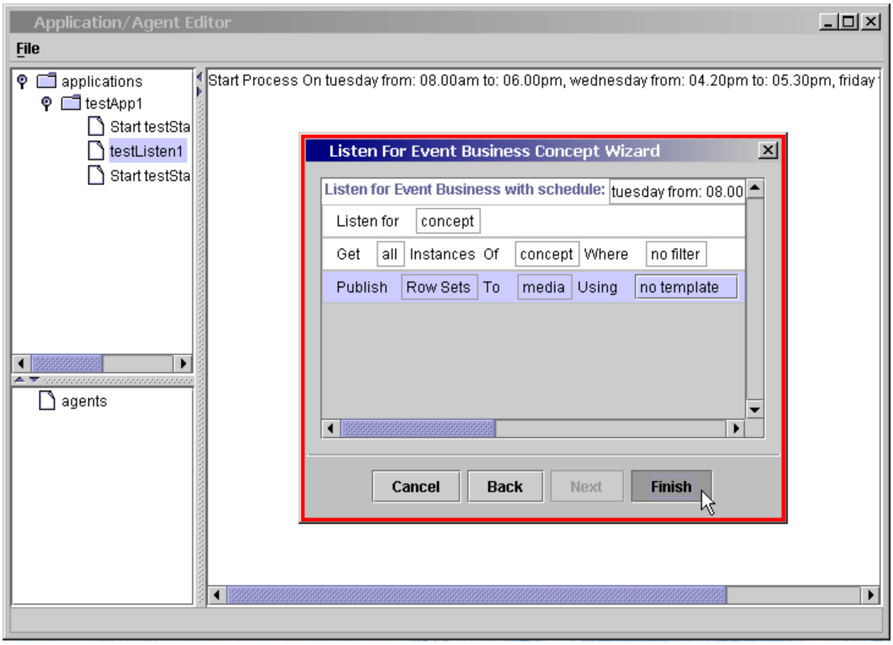
Listen for Event Business with schedule: tuesday from: 08.00

Listen for concept

To add as many Application Block code elements as required, repeat this option-click context menu action.

6 When you have finished adding Application Block Code, select the Finish button.

Figure 9-39. Adding Application Block Code - Finished Adding Application Block Code



Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday

Listen for Event Business with schedule: tuesday from: 08.00

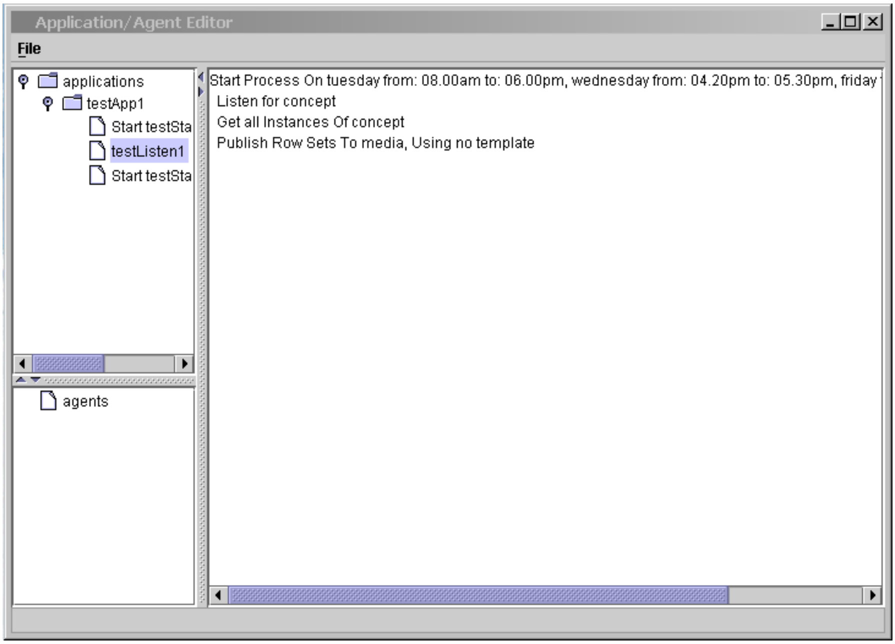
Listen for concept

Get all Instances Of concept Where no filter

Publish Row Sets To media Using no template

7 The Application/Agent Editor displays. The Application Block you modified remains selected. The right-hand panel now displays the new Application Block code.

Figure 9-40. Application/Agent Editor - Application Block Code Added



```
Start Process On tuesday from: 08.00am to: 06.00pm, wednesday from: 04.20pm to: 05.30pm, friday
 Listen for concept
 Get all Instances Of concept
 Publish Row Sets To media, Using no template
```

You have now finished adding new Application Block code.

Using Application Block Code

The Application Block code elements display within the Application Wizards with their active, editable components highlighted. Click within the highlighted region to activate the context-sensitive, pop-up edit functionality. Select the Finish button to exit the Application Wizards. Your changes display on the right-hand panel of the Application/ Agent Editor.

The commands execute sequentially, in a "flow."

Identifying Command Flow Result Tokens

When a command executes, it returns a token to Knowledge Broker that can then be used by a subsequent command in the "flow". The nature of the token varies according to the command. Some commands return a list of elements and associated values: these are Rowsets. Other commands return single values: these are Variables.

Understanding Apply Agent

The Apply Agent command enables you to call a named Agent. The format is:

```
Apply agent
```

Click the **agent** drop-down to display a list of available Agents. Click to select your desired Agent.

Specifies that any Rowset or Variable generated by the applied Agent can be used in the current flow.

Understanding Get Business Concept

The Get Business Concept command enables you to selectively convert the data from a Business Concept into a RowSet, that is, a selection of instance data. Knowledge Broker's components can further manipulate this RowSet. The format is:

```
Get all Instances of concept Where no filter, Including Attribute List
```

Click the **`all`** drop-down to display the Select number of instances dialog. Your choices here are

1. First *n* instances
2. All instances.

Click the **`concept`** widget to display the Select a Business Concept dialog. A list of available Business Concepts displays. Use the cursor to select your desired Business Concept. Select the OK button to confirm your choice.

Click the **`no filter`** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operations onto instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Click the **`Including`** drop-down widget to display a list of options. Click to select either *Including* or *Excluding*.

Click the **`Attribute List`** widget to display the Select Attributes dialog. A list of available Business Concepts and their attributes displays. Use the cursor and click the >> button to select your desired Attributes. Select the OK button to confirm your choice.

The rows are linked by the logical operators **`and`** and **`or`**. Because of arithmetical and logical commutativity and transitivity, the sequence order in which you perform Element comparisons and logical operations affects the Resultset, that is, the set of all instances returned that satisfy your constraints.

The Business Concept filter expressions calculate with standard left-to-right precedence, The expressions grouped inside parentheses operators are calculated first. That is, the entire expression expands from the deepest parentheses, working "up" until all parenthetical operations are complete. Then the expression is calculated from left-to-right.

Your expression displays in the bottom panel. When you are satisfied with your expression, select the Ok button to commit your filter.

Returns a Rowset.

Understanding Listen

The Listen Command enables you to listen for an Event Business Concept that will trigger this Application Block execution. The format is:

```
Listen for concept
```

Click the **`concept`** widget to display the Select a Business Concept dialog. A list of available Business Concepts displays. Use the cursor to select your desired Business Concept. Select the Ok button to confirm your choice.

Event only. Displays a list of available Event Business Concepts.

Understanding Reply

The Reply command enables you to return a named RowSet. You can manipulate the format of the RowSet using a named Extensible Stylesheet Language Transformations (XSLT). The format is:

```
Reply With RowSet Using no template
```

Click the **`RowSet`** widget to display the Select Rowset dialog. Business Concepts are the "root" RowSets while other displayed RowSets depend on other Application and Agent activity. Select the Ok button to confirm your choice.

Click the **`no template`** text field to type the name of a valid Connection name (except for an XML file where the file name is used).

Returns a Rowset.

Understanding Publish

The Publish command enables you to output RowSet result data from the Application to external datasources, message feeds, or third-party applications. The format is:

```
Publish RowSets To media Using no template
```

Click the **`RowSet`** widget to display the Select row sets dialog. Business Concepts are the "root" RowSets while other displayed RowSets depend on other Application and Agent activity. Select the Ok button to confirm your choice.

Click the **`media`** drop-down to display a list of your publishing media options. The options are:

1. xmlFile - Publish to an XML file
2. csvFile - Publish to CSV file
3. jmsMessage - Publish to a JMS message
4. tibrvMessage - Publish to a TIB/rv message
5. rdbmsTable - Publish to a relational database table

Click the **`no template`** text field to type the name of a valid Connection name (except for an XML file where the file name is used).

Returns a Rowset.

Understanding Execute Query

```
[RESULT] = Execute Query query With Schema Schema And Element Schema
Element
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`query`** widget to display the Select Query dialog box. The options are: Defined By Query and Defined By Variable. If you select Defined By Query you must type the Query definition in the text box. If you select Defined by Variable you must select the variable from the pull-down menu. Select the OK button to confirm your choice.

Click the **`Schema`** pull-down menu to display a list of schema options. Click to select the desired schema.

Click the **`Schema Element`** pull-down menu to display a list of business concepts. Click to select the desired business concept.

Returns a Rowset or a Variable.

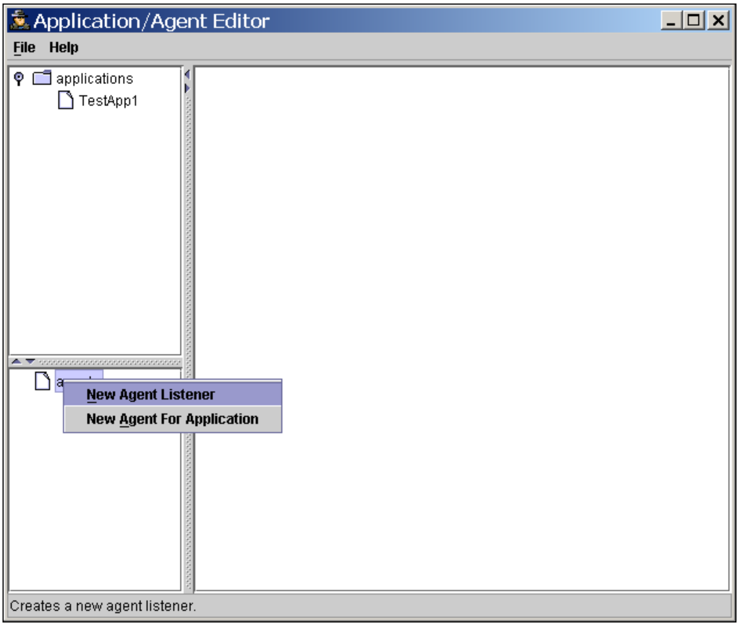
Creating and Deleting Agents

Creating an Agent

1 Open the Application/Agent Editor. Initially, it displays no entries because no Applications or Agents have been created.

2 Select Agent and option-click. The context menu displays.

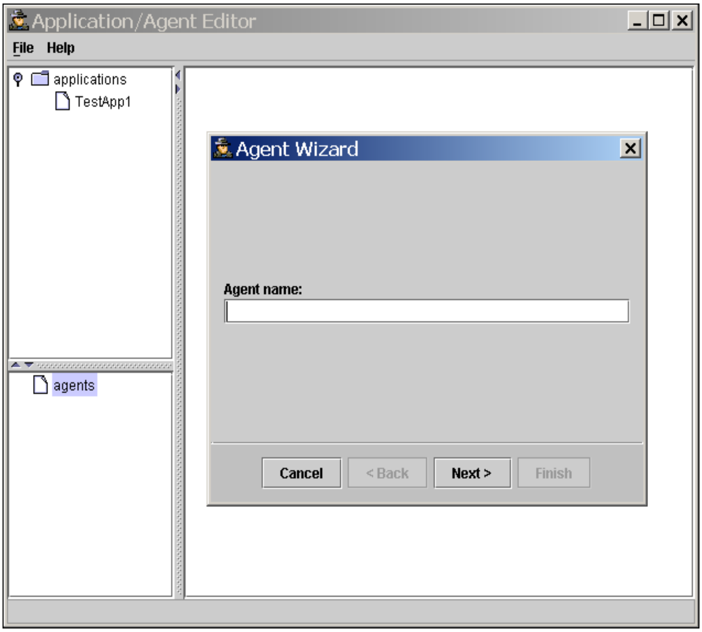
Figure 9-41. Application/Agent Editor - New Agent Listener - Adding Agent



Creates a new agent listener.

3 Select New Agent Listener or New Agent for Application . The Agent Wizard displays.

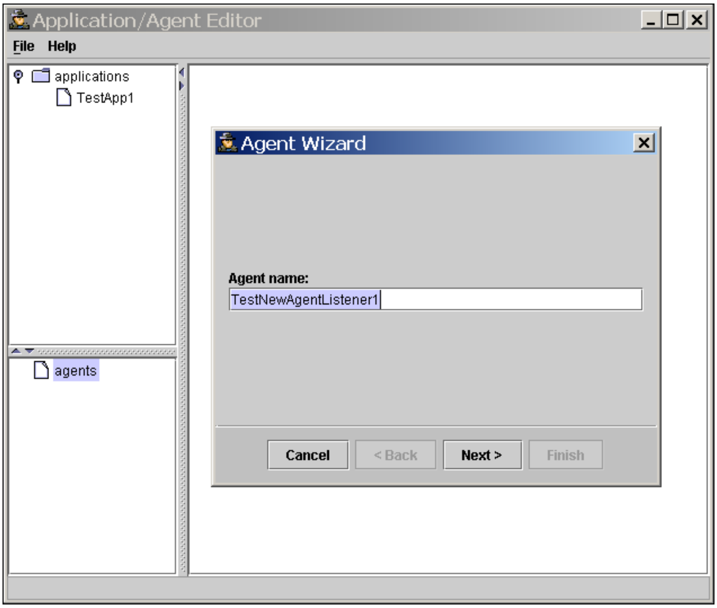
Figure 9-42. Application/Agent Editor - Agent Wizard Displays



Agent name:

4 Type a name for the New Agent and click Next.

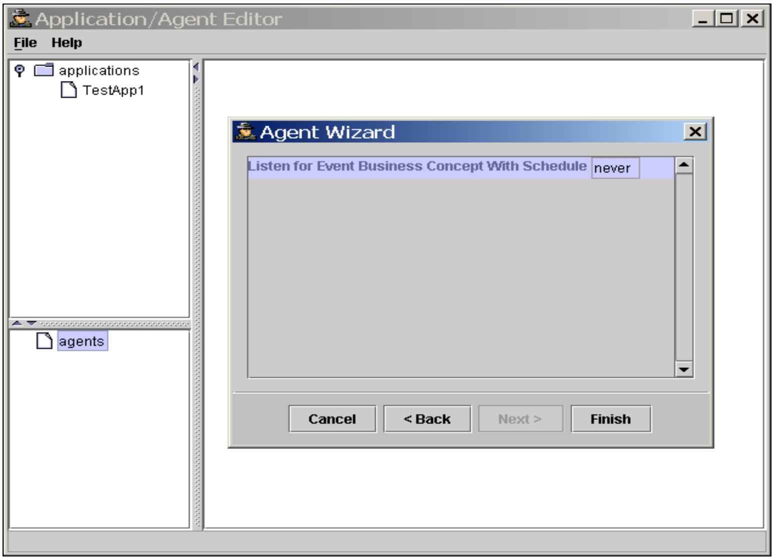
Figure 9-43. Application/Agent Editor - Agent Wizard Displays



Agent name: TestNewAgentListener1

5 The Wizard now displays the first line of the New Agent Block. Click Finish..

Figure 9-44. New Agent Listener Wizard - Initial Schedule Block Code



Listen for Event Business Concept With Schedule never

The Application/Agent Editor now displays the new, blank Agent.

Deleting an Agent

1 Open the Application/Agent Editor, if it is not currently open.

2 Select the desired Agent.

3 **Option Click > Delete**. You have now deleted An Agent.

Triggering Agents

Agents are triggered in one of two ways:

1 During a specified schedule

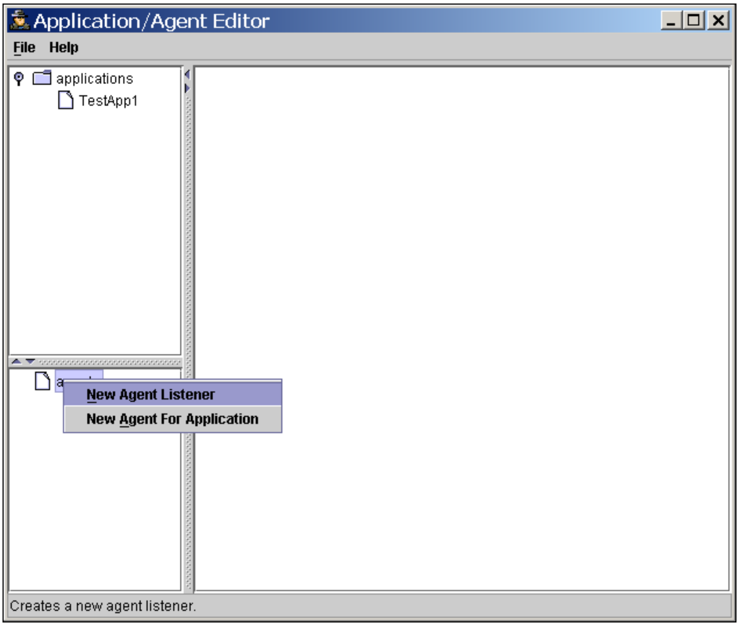
2 Called by an Application or Agent.

Creating a New Agent Listener

The Agent Listener command creates a named Agent that triggers during a specified schedule.

1 Open the Application/Agent Editor. Select Agent and option-click. The context menu displays.

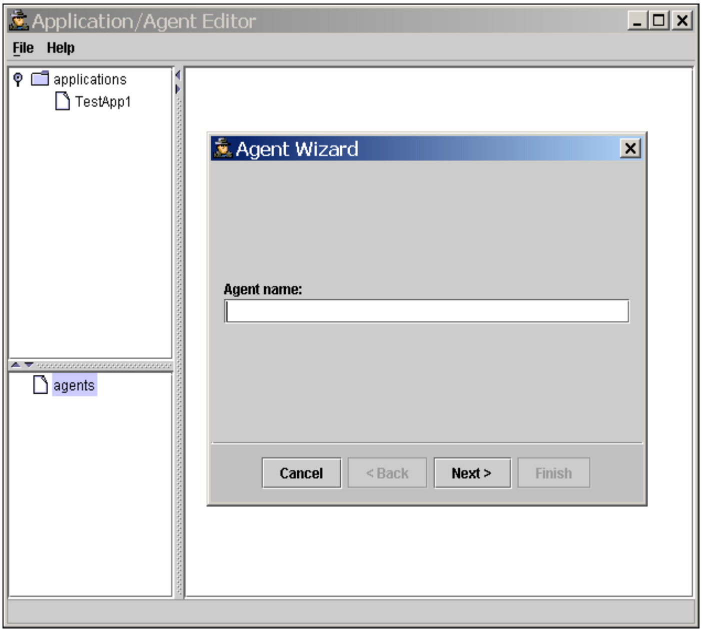
Figure 9-45. Application/Agent Editor - New Agent Listener - Adding



Creates a new agent listener.

2 Select New Agent Listener. The Agent Wizard displays.

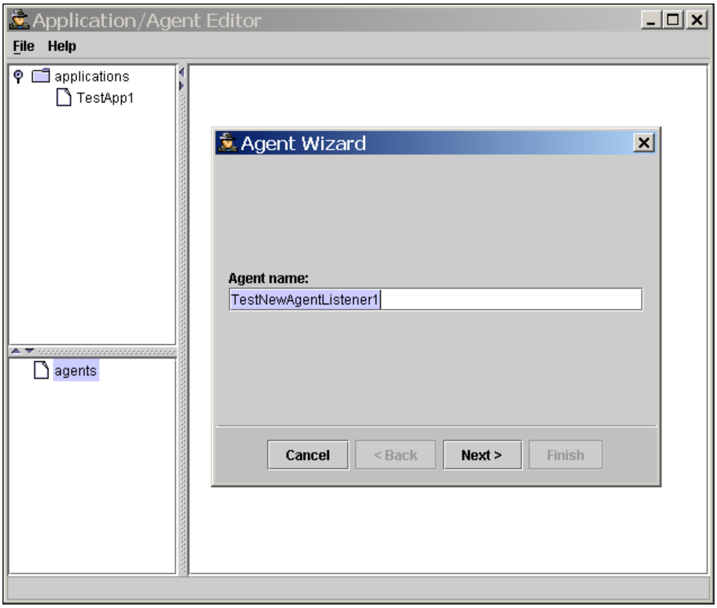
Figure 9-46. Application/Agent Editor - Agent Wizard Displays



Agent name:

3 Type a name for the New Agent Listener block and click the Next button to continue.

Figure 9-47. Application/Agent Editor - Agent Wizard Displays

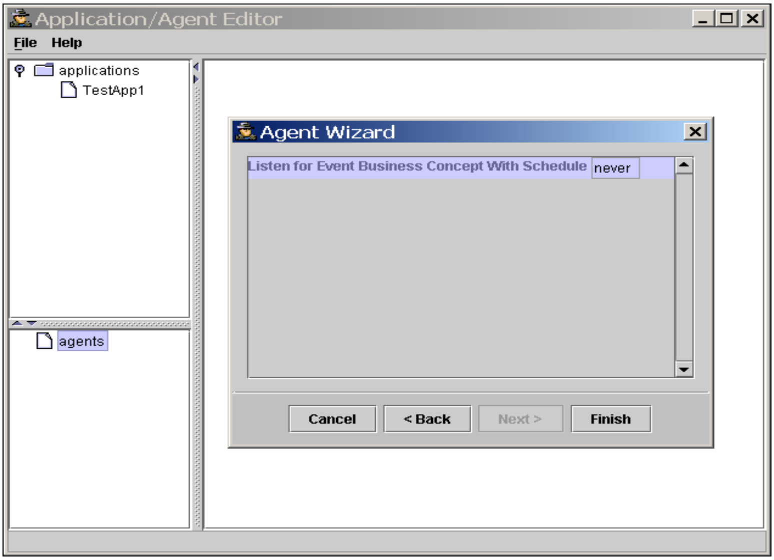


Agent name: TestNewAgentListener1

4 The Wizard now displays the first line of the New Agent Listener Block.

```
Listen for Event Business with schedule:schedule.
```

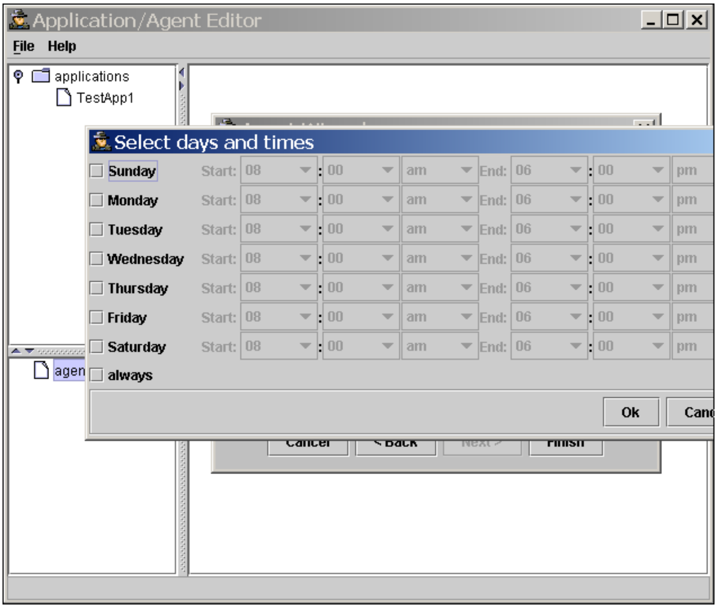
Figure 9-48. New Agent Listener Wizard - Initial Schedule Block Code



Listen for Event Business Concept With Schedule never

5 Select the schedule pop-up text. A daily Schedule displays.

Figure 9-49. New Agent Listener Wizard - Schedule Popup

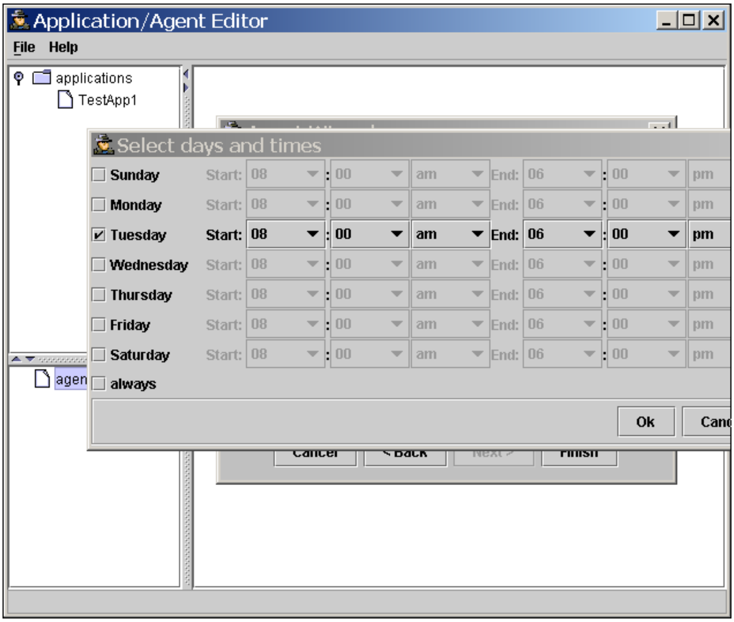


Select days and times

☐ Sunday	Start:	08	:	00	am	End:	06	:	00	pm
☐ Monday	Start:	08	:	00	am	End:	06	:	00	pm
☐ Tuesday	Start:	08	:	00	am	End:	06	:	00	pm
☐ Wednesday	Start:	08	:	00	am	End:	06	:	00	pm
☐ Thursday	Start:	08	:	00	am	End:	06	:	00	pm
☐ Friday	Start:	08	:	00	am	End:	06	:	00	pm
☐ Saturday	Start:	08	:	00	am	End:	06	:	00	pm
☐ always										

6 Use the cursor to place a checkmark next to a day of the week for which you want the New Agent Listener to be active. That row highlights.

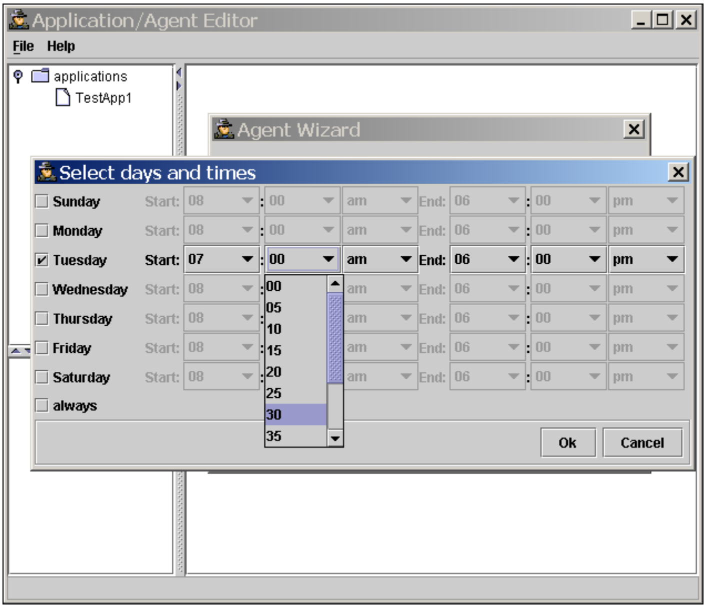
Figure 9-50. New Agent Listener Wizard - Day Selected



□ Sunday	Start:	08	: 00	am	End:	06	: 00	pm
□ Monday	Start:	08	: 00	am	End:	06	: 00	pm
☑ Tuesday	Start:	08	: 00	am	End:	06	: 00	pm
□ Wednesday	Start:	08	: 00	am	End:	06	: 00	pm
□ Thursday	Start:	08	: 00	am	End:	06	: 00	pm
□ Friday	Start:	08	: 00	am	End:	06	: 00	pm
□ Saturday	Start:	08	: 00	am	End:	06	: 00	pm
□ always								

7 Use the cursor to specify a Start time for your New Agent Listener.

Figure 9-51. New Agent Listener Wizard - Start Time Selected

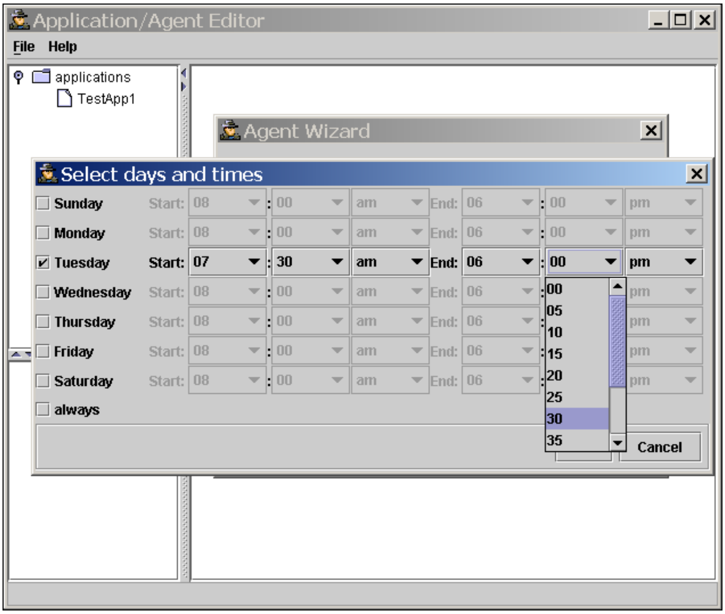


□ Sunday	Start:	08	: 00	am	End:	06	: 00	pm
□ Monday	Start:	08	: 00	am	End:	06	: 00	pm
☑ Tuesday	Start:	07	: 00	am	End:	06	: 00	pm
□ Wednesday	Start:	08	:	am	End:	06	: 00	pm
□ Thursday	Start:	08	:	am	End:	06	: 00	pm
□ Friday	Start:	08	:	am	End:	06	: 00	pm
□ Saturday	Start:	08	:	am	End:	06	: 00	pm

□ always

8 Use the cursor to specify an End time for your New Agent Listener.

Figure 9-52. New Agent Listener Wizard - End Time Selected



☐ Sunday	Start:	08	: 00	am	End:	06	: 00	pm
☐ Monday	Start:	08	: 00	am	End:	06	: 00	pm
☑ Tuesday	Start:	07	: 30	am	End:	06	: 00	pm
☐ Wednesday	Start:	08	: 00	am	End:	06	:	pm
☐ Thursday	Start:	08	: 00	am	End:	06	:	pm
☐ Friday	Start:	08	: 00	am	End:	06	:	pm
☐ Saturday	Start:	08	: 00	am	End:	06	:	pm
☐ always								

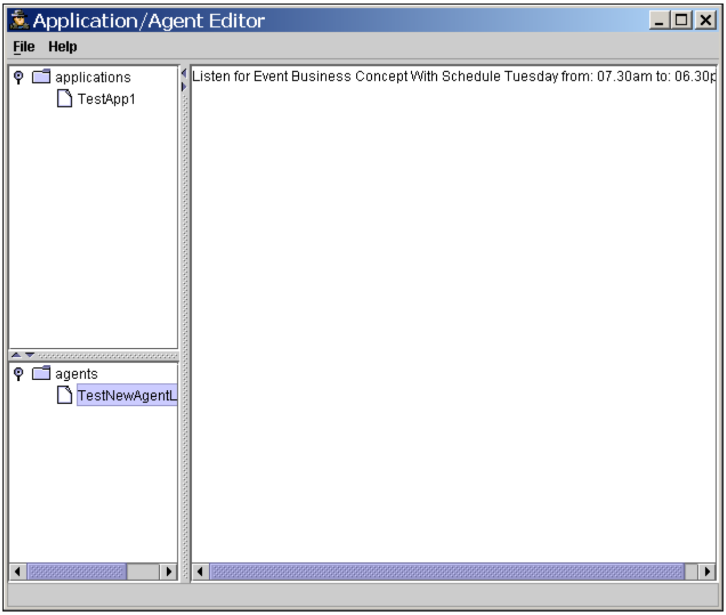
For the marked day, the New Agent Listener will activate only during the selected time period.

9 You can select more days and time periods using a similar procedure. When you have finished defining your Agent Listener Schedule, click the Ok button.

10 The New Agent Listener Wizard displays. Your New Agent Listener now has a schedule. Click the Finish button.

The Application/Agent Editor displays. The Agent Block you edited remains selected. The right-hand panel now displays the New Agent Listener Block code, including your defined schedule.

Figure 9-53. New Agent Listener Wizard- Schedule Displayed



Listen for Event Business Concept With Schedule Tuesday from: 07.30am to: 06.30p

You have successfully added a New Agent Listener.

Creating a New Agent for Application

The New Agent For Application command creates a named Agent that can be invoked by an Application or another Agent.

The procedure for creating a New Agent For Application is similar to that outlined in *Adding a Start Process on Demand Block on page 300.*

Modifying Agents

The Agent Wizards enable you to add or modify Agent Block code for specific Agents. There are 18 Agent Block code elements:

1. Select Sum
2. Extract
3. Listen
4. Reply
5. Create Assignment Expression
6. Infer Over
7. Java
8. Get By Reference
9. Join
10. Select business concept
11. Select Average
12. Apply Agent
13. Select Minimum
14. Publish
15. Select Maximum
16. Sort
17. Get Business Concept
18. Round

The procedure for adding Agent Block Code is similar to that outlined in *Adding Application Block Code on page 307*.

Using Agent Block Code

The Agent Block code elements display within the Agent Wizards with their active, editable components highlighted. Click within the highlighted region to activate the context-sensitive, pop-up edit functionality. Select the Finish button to exit the Agent Wizards. Your changes display on the right-hand panel of the Application/Agent Editor.

The commands execute sequentially, in a “flow”.

Identifying Command Flow Result Tokens

When a command executes, it returns a token to Knowledge Broker that can then be used by a subsequent command in the “flow”. The nature of the token varies according to the command. Some commands return a list of elements and associated values: these are Rowsets. Others return single values: these are Variables.

Understanding Select Sum

The Sum Of command enables you to create a named RowSet result that is the sum of a additive operation on selected instance data.

```
[RESULT] = Sum Of RowSet.attribute Group By no grouping Where no filter
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`RowSet`** widget to display the Select row sets dialog. Choose your desired RowSets. Select the Ok button to confirm your choice.

Click the **`attribute`** widget to display a list of Elements from the selected RowSet. Choose for which Element’s instance data you want a Sum.

Click the **`no grouping`** widget to select your grouping.

Click the **`no filter`** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Returns a Rowset if Group By is used.

Returns a Variable if Group By is unused.

Understanding Extract

The Extract command parses sets of instances taken from specified RowSets, looking for semi-structured news stories or analyst recommendations, and attempts to extract meaning from these data. It returns a vector of the results.

```
[RESULT] = Extract From RowSet.attribute
```

Click the **RESULT** text field to type a name for the ResultSet.

Click the **RowSet** widget to display the Select row sets dialog. Choose your desired RowSets. Select the Ok button to confirm your choice.

Click the **attribute** widget to display a list of Elements from the selected RowSet. Choose for which Element's instance data you want to extract meaning.

Returns a Rowset.

Understanding Listen

The Listen Command enables you to listen for an Event Business Concept that will trigger this Agent Block execution. The format is:

```
    Listen for concept
```

Click the **concept** widget to display the Select a Business Concept dialog. A list of available Business Concepts displays. Use the cursor to select your desired Business Concept. Select the Ok button to confirm your choice.

Event only. Displays a list of available Event Business Concepts.

Understanding Reply

The Reply command enables you to return a named RowSet. You can manipulate the format of the RowSet using a named Extensible Stylesheet Language Transformations (XSLT). The format is:

```
    Reply With RowSet Using no template
```

Click the **RowSet** widget to display the Select row sets dialog. Business Concepts are the "root" RowSets while other displayed RowSets depend on other Application and Agent activity. Select the Ok button to confirm your choice.

Click the **no template** text field to type the name of a valid Connection name (except for an XML file where the file name is used).

Returns a Rowset.

Understanding Create Assignment Expression

The Create Assignment Expression command enables you to type a freeform mathematical function to return a named RowSet.

```
[RESULT] = expression
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`expression`** text field to type your mathematical function.

Returns a Variable.

Understanding Infer Over

The Infer Over command enables you pass named RowSets to Knowledge Broker's inference engine for RETE processing. The format is:

```
[RESULT] = Infer Over RowSets Using Ruleset And returning Verb List
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`RowSet`** widget to display the Select row sets dialog. Choose your desired RowSets. Select the Ok button to confirm your choice.

Click the **`Ruleset`** widget to display the Select rule sets dialog. Choose your desired RulesetRulesets. Select the Ok button to confirm your choice.

Click the **`Verb List`** text field to type your desired list of verbs. The inference engine will only return recommendations using these verbs. Select the Ok button to confirm your choice.

Returns a Rowset

Understanding Java

The Java command enables you to enter raw Java code that will execute. The format is:

```
Java java...
```

Click the **`java...`** widget. The Java code dialog displays. You can type or paste your Java code here. Java code has complete access to all in-memory RowSets and Variables.

Understanding Get By Reference

[RESULT] = Get From Concept Variable **variable** , Using Filter Variable **no filter**

Click the **RESULT** text field to type a name for the ResultSet.

Click the **variable** text field to type a named RowSet or Business Concept.

Click the **no filter** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Returns a Rowset.

Understanding Join

The Join command enables you to create a named RowSet that is the result of a standard join operation on two other rowsets, using a selected Element as primary keys. The format is:

[RESULT] = Join **RowSet** , **RowSet** On **attribute list**

Click the **RESULT** text field to type a name for the ResultSet.

Click each **RowSet** widget to display the Select row sets dialog. Business Concepts are the "root" RowSets while other displayed RowSets depend on other Application and Agent activity. Select the Ok button to confirm your choice.

Click the **attribute list** widget to display a list of Elements from both selected RowSets. The procedure here is similar to the Join process in the Mapping Wizard of the Ontology Editor.

Returns a Rowset

Understanding Select business concept

The Select business concept command enables you to create a named RowSet that contains filtered instance data from a Business Concept. The format is:

```
[RESULT] = Select Attribute List From concept, all Instances, Where no
filter
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`Attribute List`** widget to display the Select Attributes dialog. A list of available Business Concepts and their attributes displays. Use the cursor and click the >> button to select your desired Attributes. Select the OK button to confirm your choice.

Click the **`concept`** widget to display the Select a Business Concept dialog. A list of available Business Concepts displays. Use the cursor to select your desired Business Concept. Select the Ok button to confirm your choice.

Click the **`all`** drop-down to display the Select number of instances dialog. Your choices here are

1. First *n* instances
2. All instances.

Click the **`no filter`** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

You can add, remove, or delete Element comparison rows using the toolbar icons.

Your expression displays in the bottom panel. When you are satisfied with your expression, select the Ok button to commit your filter.

Returns a Rowset for a multiple (>1) instances.

Returns a Variable for a single (=1) instance.

Understanding Select Average

The Sum Of command enables you to create a named RowSet result that is the statistical average of selected instance data.

```
[RESULT] = Average Of RowSet.attribute Group By no grouping Where no filter
```

Click the **RESULT** text field to type a name for the ResultSet.

Click the **RowSet** widget to display the Select row sets dialog. Choose your desired RowSets. Select the Ok button to confirm your choice.

Click the **attribute** widget to display a list of Elements from the selected RowSet. Choose for which Element's instance data you want an Average.

Click the **no grouping** widget to select your grouping.

Click the **no filter** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Returns a Rowset if Group By is used.

Returns a Variable if Group By is unused.

Understanding Apply Agent

The Apply Agent command enables you to call a named Agent. The format is:

```
Apply agent
```

Click the **agent** drop-down to display a list of available Agents. Click to select your desired Agent.

Specifies that any Rowset or Variable generated by the applied Agent can be used in the current flow.

Understanding Select Minimum

The Select Minimum command enables you to return the minimum value of a set of instances taken from specified RowSets.

```
[RESULT] = Minimum Of RowSet.attribute Group By no grouping Where no filter
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`RowSet`** widget to display the Select row sets dialog. Choose your desired RowSets. Select the Ok button to confirm your choice.

Click the **`attribute`** widget to display a list of Elements from the selected RowSet. Choose for which Element's instance data you want a minimum.

Click the **`no grouping`** widget to select your grouping.

Click the **`no filter`** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Returns a Rowset if Group By is used.

Returns a Variable if Group By is unused.

Understanding Publish

The Publish command enables you to output RowSet result data from the Agent to external datasources, message feeds, or third-party applications. The format is:

```
    Publish RowSets To media Using no template
```

Click the **`RowSet`** widget to display the Select row sets dialog. Business Concepts are the “root” RowSets while other displayed RowSets depend on other Application and Agent activity. Select the Ok button to confirm your choice.

Click the **`media`** drop-down to display a list of your publishing media options:

1. xmlFile - Publish to an XML file
2. csvFile - Publish to CSV file
3. jmsMessage - Publish to a JMS message
4. tibrvMessage - Publish to a TIB/rv message
5. rdbmsTable - Publish to a relational database table

Click the **`no template`** text field to type the name of a valid Connection name (except for an XML file where the file name is used).

Returns a Rowset.

Understanding Select Maximum

The Select Maximum command enables you to create a named RowSet result that is the statistical average of selected instance data.

```
[RESULT] = Maximum Of RowSet.attribute Group By no grouping Where no filter
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`RowSet`** widget to display the Select row sets dialog. Choose your desired RowSets. Select the Ok button to confirm your choice.

Click the **`attribute`** widget to display a list of Elements from the selected RowSet. Choose for which Element's instance data you want a Maximum.

Click the **`no grouping`** widget to select your grouping.

Click the **`no filter`** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Returns a Rowset if Group By is used.

Returns a Variable if Group By is unused.

Understanding Sort

The Sort command enables you to sort a Business Concept instance data as desired. The format is:

```
    Sort concept On attribute, Ascending
```

Click the **`concept`** widget to display the Select a Business Concept dialog. A list of available Business Concepts displays. Use the cursor to select your desired Business Concept. Select the Ok button to confirm your choice.

Click the **`attribute`** widget to display a list of Elements from the selected RowSet. Choose for which Element's instance data you want to perform a Sort.

Click the **Ascending** widget to select between *ascending* or *descending* sort order.

Returns a Rowset.

Understanding Get Business Concept

The Get Business Concept command enables you to selectively convert the data from a Business Concept into a RowSet, that is, a selection of instance data. Knowledge Broker's components can further manipulate this RowSet. The format is:

```
Get all Instances Of concept Where no filter, Including Attribute List
```

Click the **`all`** drop-down to display the Select number of instances dialog. Your choices here are

1 First *n* instances

2 All instances.

Click the **`concept`** widget to display the Select a Business Concept dialog. A list of available Business Concepts displays. Use the cursor to select your desired Business Concept. Select the Ok button to confirm your choice.

Click the **`no filter`** widget to display the Select Attributes dialog. Within this dialog you can apply arithmetical and string comparison operation to instance data from Business Concept Elements. Only the instance data satisfying your logical operations will be returned.

Click the **`Including`** drop-down widget to display a list of options. Click to select either Including or Excluding.

Click the **`Attribute List`** widget to display the Select Attributes dialog. A list of available Business Concepts and their attributes displays. Use the cursor and click the >> button to select your desired Attributes. Select the OK button to confirm your choice.

You can add, remove, or delete Element comparison rows using the toolbar icons.

Your expression displays in the bottom panel. When you are satisfied with your expression, select the Ok button to commit your filter.

Returns a Rowset.

Understanding Round

The Round command enables you to create a named RowSet that contains truncated data. It is useful in financial calculations. You can also use it to convert decimals to integer values.

```
[RESULT] = Round variable To 10 Places
```

Click the **`RESULT`** text field to type a name for the ResultSet.

Click the **`variable`** text field to type a named RowSet or Business Concept.

Click the **`10`** text field to type your number of desired decimal digits.

Returns a Variable.

Chapter

10

Creating Rules

This chapter provides step-by-step instructions for defining Rules. *Rules* are logical statements that describe how an object (person, place, thing, or event) will behave in a particular situation. They are the means for “teaching” the computer which datasources to query, patterns to analyze, implications to recognize, and real-time recommendations to make.

Knowledge Broker contains a Rule Editor that enables you to define rules that describe two things: how your business typically responds to individual transactions and what situations will require different responses to seemingly similar transactions.

Introducing Rules

There are three basic types of rules:

- *Legal rules,* which are mandated by a federal, state, city, or other warrantable agency (including SEC, OSHA, FCC, EPA, and unions), cannot be broken unless the issuing agency amends the rule. For example, an SEC rule mandates a “quiet period” before a company’s initial public offering.
- *Domain-specific rules,* which are defined by a specific business or field of knowledge (including physics, chemistry, and law), can also not be broken unless amended by the issuing domain. For example, a brokerage house may stipulate that stockbrokers employed with the firm for less than six months may not offer speculative stock advice.
- *Common-sense rules,* which include opinions, hunches, and best practices, can be broken if circumstances warrant. For example, a brokerage house may institute a “best practices” policy whereby speculative stocks will not be recommended to a person who is 70 years old and living on a fixed income.

Knowledge Broker currently supports a single category of rules:

- *Constraint Satisfaction*, which enables you to identify the rules, concepts, and data that should be processed by Knowledge Broker.[1]

Each rule consists of three parts in the basic form:

```
if <SUBJECT> then <VERB > <OBJECT>
```

or, simply, "**if X is true, then perform action Y on Z**".

To say that "X is true" actually means to find within the ontology a greater-than-zero number of items that can be described by X. For example, if X is "customer's age is less than 45", then there must be at least one customer satisfying that condition for the rule to activate. Thus, a *subject* that is "true" can be also be described as being "non empty".

[1] Future releases of Knowledge Broker will support these additional subcategories of rules: *Sequencing*, which enables you to specify that some rules must be activated before another rule may be processed. *Duration*, which enables you to specify that a rule is valid only for a specified time period. *Control*, which enables you to include both global and local rules. Global rules are defined for the entire organization, while local rules are defined by an individual office, department, or person.

More complex rules are of the form:

```
IF
          <concept>
            such that {<property_value>
                        <operator>
                        <comparison_value>}
          AND
                       {<concept.property_value>
                        <operator>
                        <concept.property_value>}
THEN
     <action>
          <concept>
            such that {<property_value>
                        <operator>
                        <comparison_value>}
          AND
                       {<concept.property_value>
                        <operator>
                        <concept.property_value>}
```

where `<concept>` is a concept in the ontology

`<property_value>` is an element of the concept, retrieved from the ontology

`<operator>` is one of: `=`, `!=`, `<`, `>`, `<=`, `>=`

`<comparison_value>` is any arbitrary value

`<action>` is a verb, such as `buy` or `predict` or `recommend`

For example:

```
if customer's age is less than 45 then recommend buy adventure sports
```

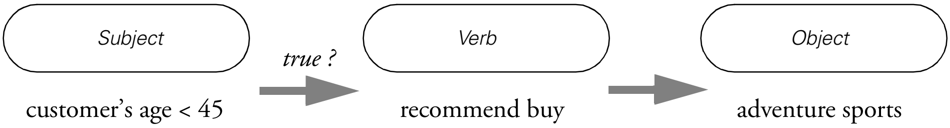
In this example, the *subject* (sometimes known as the antecedent) is a Business Concept (customer), the *property* is a property of the customer Business Concept (age), and the *property value* is a filter parameter (less than 45).

The *object* (sometimes known as the consequent) is a Business Concept (adventure sports).

The *verb* (recommend buy) acts on the direct object if the contents of the *subject* are true, or non-empty.

Another way to think about this is that the *object* (adventure sports) receives the action of the *verb* (recommend buy). The *verb* acts only if an assertion or operation in the preceding *subject* has been tested and found to be true, or non-empty.

Figure 10-1. Verb Acting on Direct Object



If the customer's age is greater than or equal to 45 then this rule will not activate.

Using Comparison Operators

Fundamentally, Knowledge Broker analyzes Rules by comparing the value (or instance) of Concept Element against the value of another Concept Element or a fixed value. One or more of these comparison assertions when combined logically constitutes the Subject Expression. The Object Expression is constructed similarly.

A Rule asserts that a comparison in the Subject is true, according to the comparison operator selected. If the assertion is true, Knowledge Broker moves on to calculate the truth or non-truth of other Subject assertions.

If they exist, multiple Subject assertions are themselves logically combined. Finally, if the overall Subject Expression calculates as true, then Knowledge Broker performs a similar test-and-aggregation on the Object Expression.

You can compare Concept Elements in the *Subject* or the *Object* parts of a rule. The format is:

```
Concept_A.Property OPERATOR Concept_B.Property
```

Comparisons are a powerful mechanism with two main functions:

1. In the *Subject* part of a rule, you use them to compare the Elements of Concepts derived from one datasource against the Elements of Concepts derived from a different datasource.
2. In the *Object* part of a rule, you use them to perform the same comparison and then to apply the Verb to those instances selected by the left Concept of the comparison (if in fact instances exist that satisfy the conditions of the comparison).

For example, to `recommend buy` specifically `Volatile Equities` of companies operating in a `Sector` preferred by your `High Risk Customers`, your *Object* Comparison expression form is:

```
    Volatile.Sector = High Risk.Sector Preference
```

The Comparison operators available are described in Table 10-1:

Table 10-1. Concept Comparison Operators

Operator	Description
=	Tests if Concept_A.Property *is equal to* Concept_B.Property.
>	Tests if Concept_A.Property *is greater than* Concept_B.Property.
>=	Tests if Concept_A.Property *is greater than or equal to* Concept_B.Property.
<	Tests if Concept_A.Property *is less than* Concept_B.Property.
<=	Tests if Concept_A.Property *is less than or equal to* Concept_B.Property.
!=	Tests if Concept_A.Property *is not equal to* Concept_B.Property.

Concept Comparison Numeric Operators

If the Properties being compared are numeric then all operators function (using sample Property values of `43.4` and `54.0`) as described in Table 10-2:

Table 10-2. Example of Concept Comparison of Numerical Properties

ConceptA. Property	Operator	ConceptB. Property	Result
43.4	=	54.0	False
43.4	>	54.0	False
43.4	>=	54.0	False
43.4	<	54.0	True
43.4	<=	54.0	True
43.4	!=	54.0	True

Concept Comparison Alphabetic Operators

If the Properties being compared are alphabetic, only the `==` and `!=` operators function. Applying any other operator will result in an undefined result and the Concept

Comparison will not perform any operation. This is illustrated in Table 10-3 (using sample Property values of `High Risk` and `Low Risk`):

Table 10-3. Example of Concept Comparison of Alphabetic Properties

ConceptA. Property	Operator	ConceptB. Property	Result
High Risk	=	Low Risk	False
High Risk	>	Low Risk	Undefined
High Risk	>=	Low Risk	Undefined
High Risk	<	Low Risk	Undefined
High Risk	<=	Low Risk	Undefined
High Risk	!=	Low Risk	True

A Comparison using an undefined operation does not produce either a `true` or `false` value. However, other Comparisons within the same Rule can still operate and the Rule can still activate (or not activate).

Using Rulesets

The Rule Editor enables you to define rules and rulesets. A rule is a logical statement that describes two things: how your business typically responds to individual transactions and what situation will require different responses in seemingly similar situations. Rules are the means for “teaching” the computer how to respond to situations relevant to your business requirements, and are the basis for generating real-time recommendations.

A Ruleset is a collection of rules. The number of rules can grow very large and in this case becomes easier and more convenient to manage them using rulesets. Rulesets each contain specific, user-defined Verbs, and the Rules within each Ruleset can use one or more of these Verbs.

Every Ruleset shares three characteristics:

Table 10-4. Ruleset Characteristics

Name	Description	Required
Ruleset name	User-entered logical name that identifies a Ruleset.	Yes
Rule Explanation Level	Sets the level of inference explanation output produced by Knowledge Broker.	Yes
Description	User-entered details explaining or categorizing a Ruleset.	No

Understanding Rule Explanations

Knowledge Broker can output explanations of its inferencing processes, enabling users to perform a “sanity check” on the validity and nature of the recommendations. The level of detail of these explanations is configurable during Ruleset creation. Initially, the explanation level is set to Low.

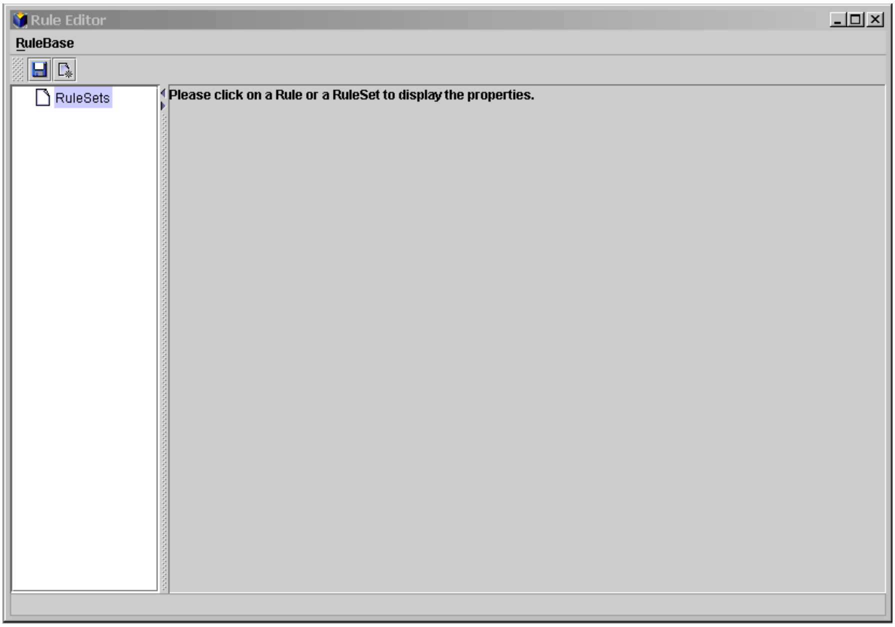
There are five levels of rule explanation:

Table 10-5. Rule Explanation Level

Name	Description
Low	This is the default.
Medium	
High	
Natural Language	
None	

The Rule Editor displays when you click the Rule Editor icon on the Launcher Toolbar.

Figure 10-2. Rule Editor - Initial Display



Please click on a Rule or a RuleSet to display the properties.

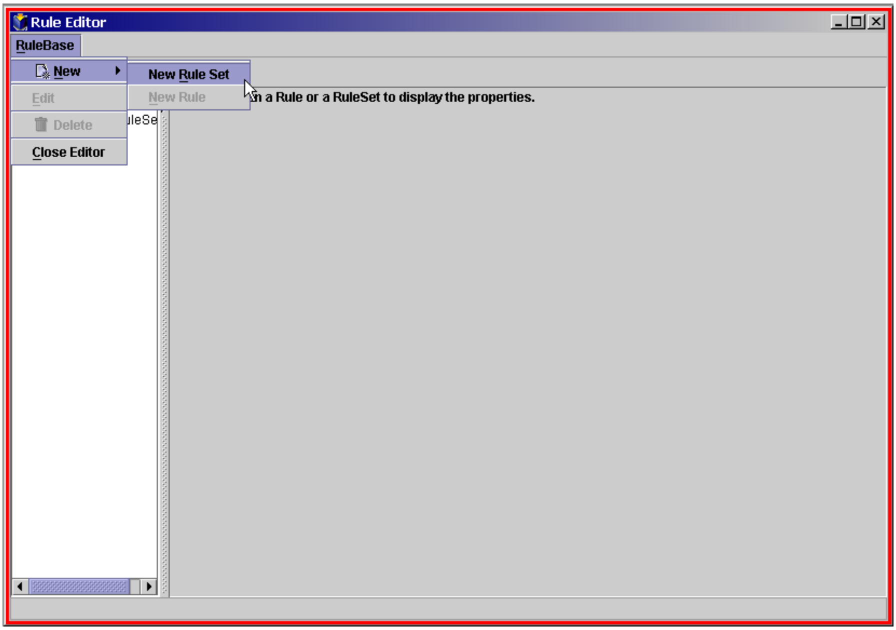
Creating a Ruleset

1 Open the Rule Editor, if it is not currently open.

2 Select the Rulesets root element within the left-hand pane.

3 Select the menu item **RuleBase > New > New Rule Set**.

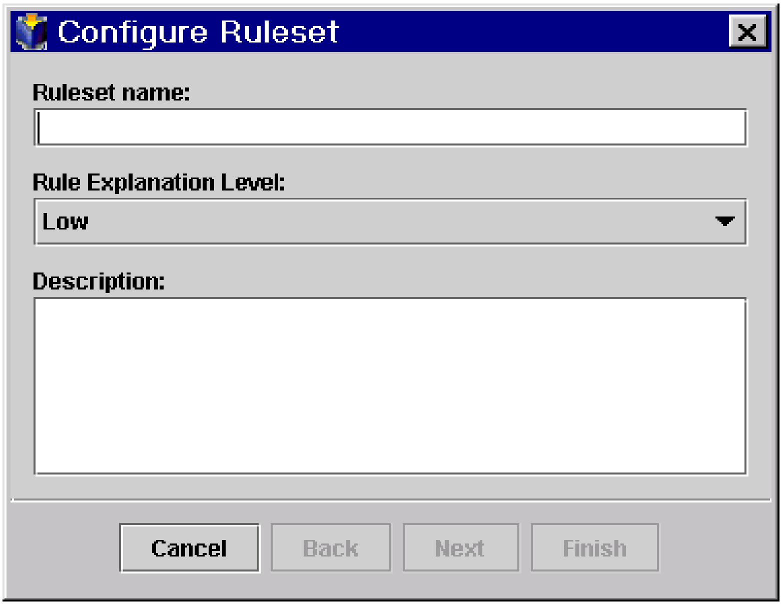
Figure 10-3. Rule Editor - Creating Ruleset - Menu Item Selected



...n a Rule or a RuleSet to display the properties.

4 The Configure Ruleset Wizard displays.

Figure 10-4. Rule Editor - Creating Ruleset - Initial Display



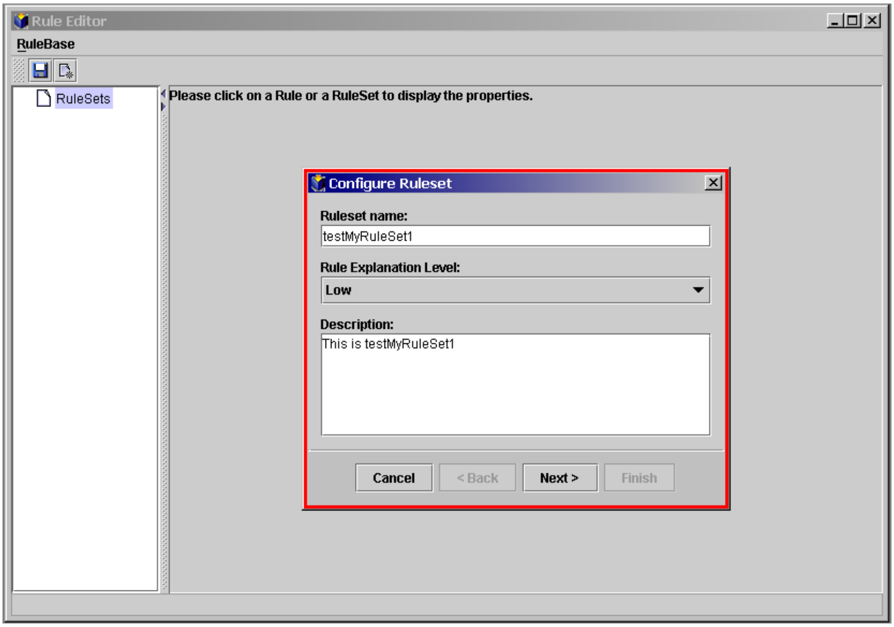
Rule Explanation Level: Low

5 Type a name for the ruleset in the Ruleset Name field.

6 Select a Rule Explanation Level from the pull-down menu.

7 Type a description for the ruleset in the Description field.

Figure 10-5. Rule Editor - Creating Ruleset - Initial Details Entered



Please click on a Rule or a RuleSet to display the properties.

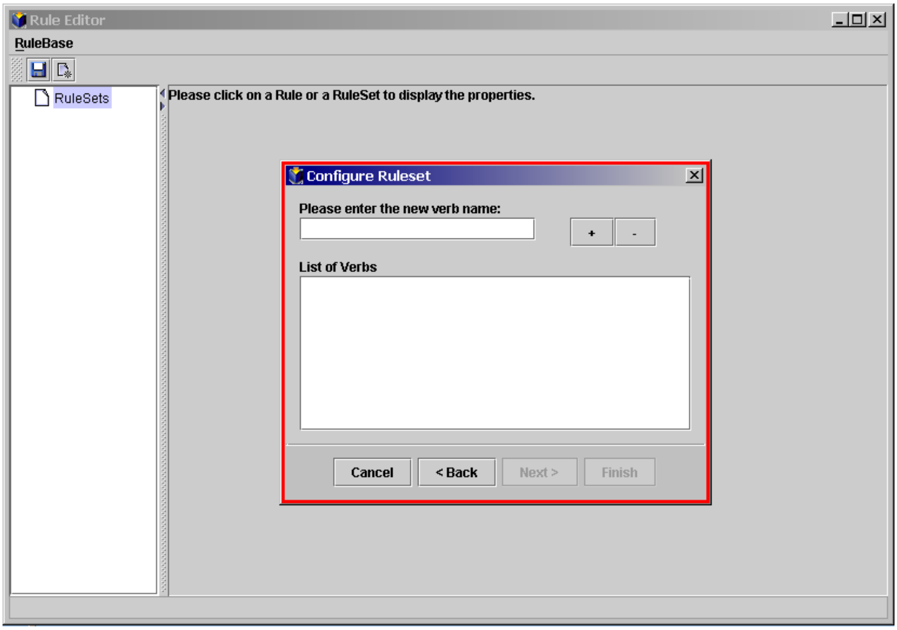
Ruleset name: testMyRuleSet1

Rule Explanation Level: Low

Description: This is testMyRuleSet1

8 Click Next. The Configure Ruleset Wizard updates, now requesting entry of the active Verbs for the Ruleset.

Figure 10-6. Rule Editor - Creating Ruleset - Entering Verb Names



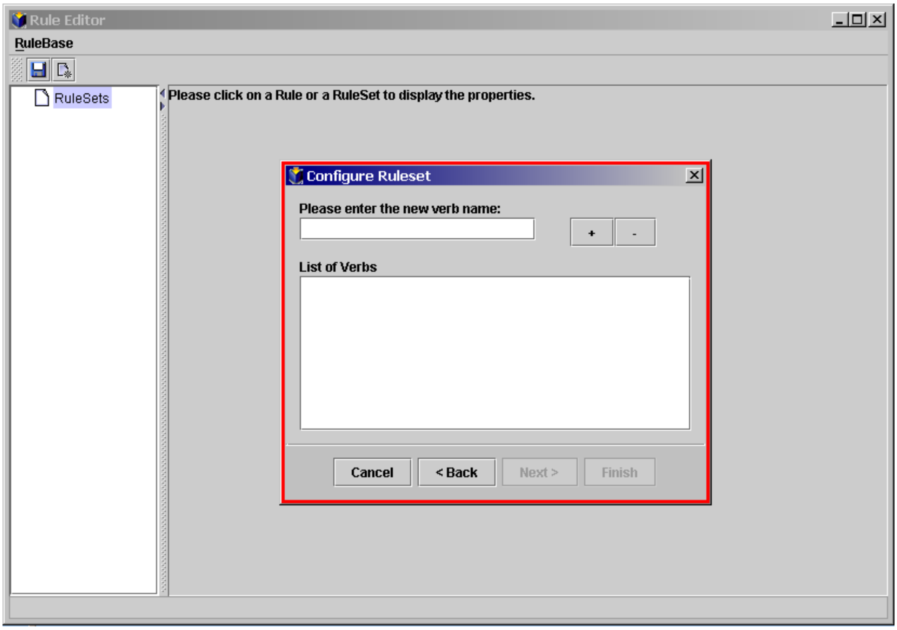
Please click on a Rule or a RuleSet to display the properties.

Please enter the new verb name:

List of Verbs

9 Type a new verb name in the New Verb Name field. Add the Verb to the List of Verbs by clicking on the "+" icon.

Figure 10-7. Rule Editor - Creating Ruleset - Adding Verbs to the List of Verbs

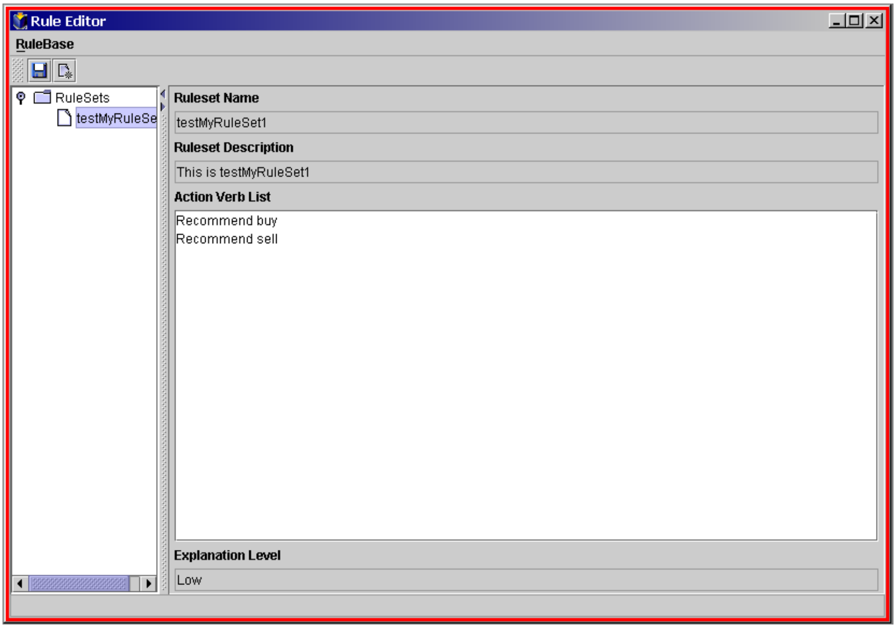


Please click on a Rule or a RuleSet to display the properties.

You can remove verb names by highlighting them in the List of Verbs and then clicking on the "-" icon.

10 Click Finish. The newly created ruleset displays in the Rule Editor.

Figure 10-8. Rule Editor



Ruleset Name

testMyRuleSet1

Ruleset Description

This is testMyRuleSet1

Action Verb List

Recommend buy
Recommend sell

Explanation Level

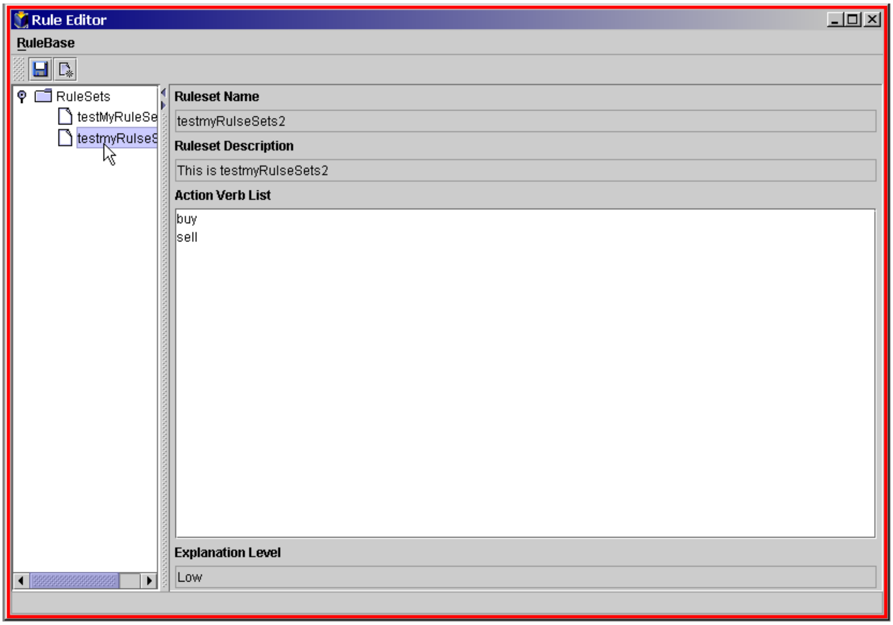
Low

Deleting a Ruleset

1 Open the Rule Editor, if it is not currently open.

2 Within the left-hand pane, select the Ruleset you wish to delete.

Figure 10-9. Rule Editor - Deleting Ruleset - Initial Display



Ruleset Name
testmyRulseSets2

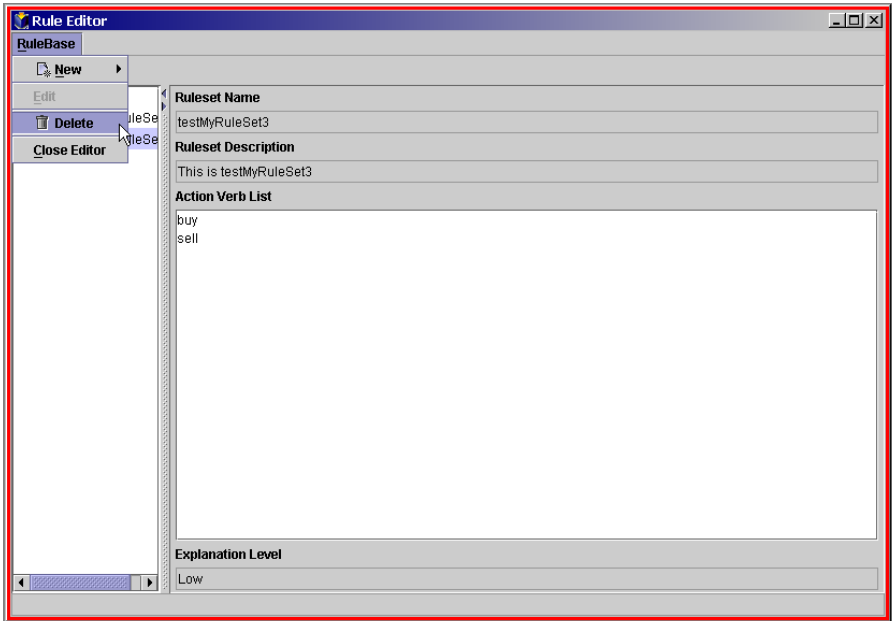
Ruleset Description
This is testmyRulseSets2

Action Verb List
buy
sell

Explanation Level
Low

3 Select the menu item **RuleBase > Delete.**

Figure 10-10. Rule Editor - Deleting Ruleset - Menu Selected



Ruleset Name: testMyRuleSet3

Ruleset Description: This is testMyRuleSet3

Action Verb List:
buy
sell

Explanation Level: Low

4 The Rule Editor redisplays with the selected Ruleset deleted.

Using Rules

You modify the contents of Rulesets by creating deleting and editing their component Rules, using the Rule Editor.

Every Rule shares four characteristics:

Table 10-6. Ruleset Characteristics

Name	Description	Required
Ruleset	Identifies the “container” Ruleset for a Rule. This value is auto-generated by the Configure Rule Wizard.	Yes
Rule Id	Identifies the Rule using a numeric token. This value is auto-generated by the Configure Rule Wizard.	Yes
Priority	Identifies with which level of priority a Rule will execute within Knowledge Broker’s inference engine. This value is user-selected from one of `Low`, `Medium`, or `High`.	Yes
Description	User-entered details explaining or categorizing a Rule.	No

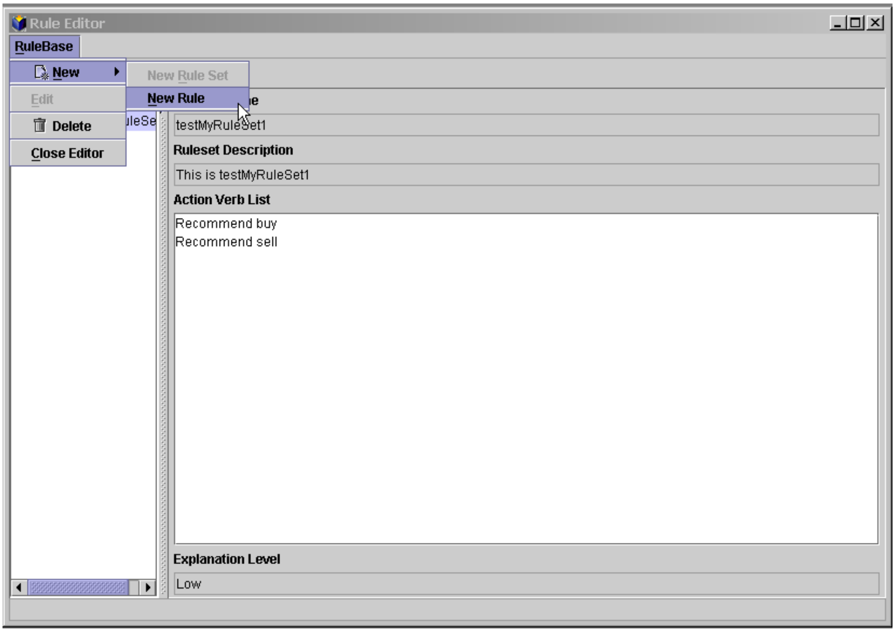
Creating a Rule

1 Open the Rule Editor, if it is not currently open.

2 Select a Ruleset element from within the left-hand panel.

3 Select the menu item **RuleBase > New > New Rule**.

Figure 10-11. Creating a Rule - Menu Item Selected



testMyRuleSet1

Ruleset Description

This is testMyRuleSet1

Action Verb List

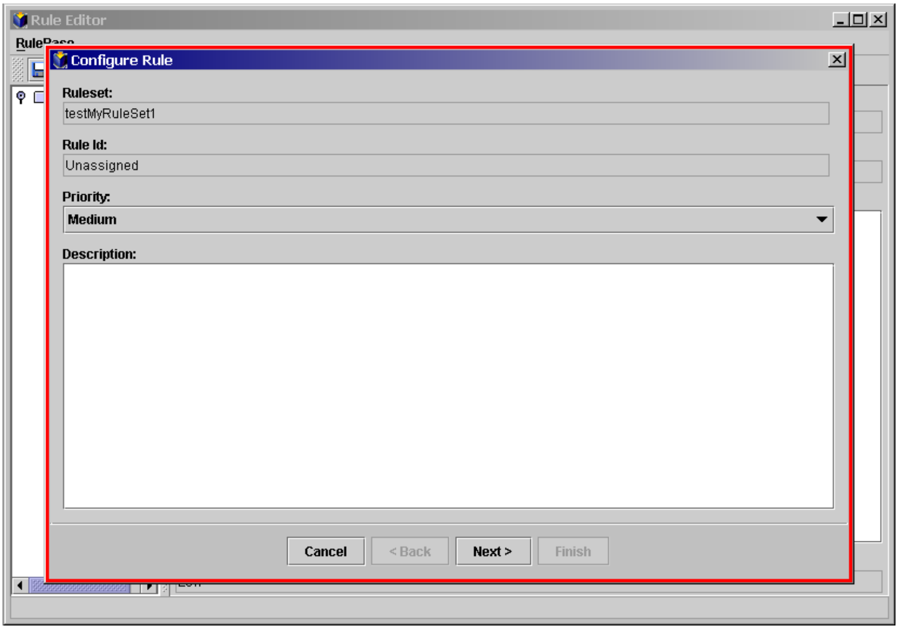
Recommend buy
Recommend sell

Explanation Level

Low

4 The Configure Rule Wizard displays.

Figure 10-12. Creating a Rule - Configure Rule Wizard - Initial Display



Ruleset: testMyRuleSet1

Rule Id: Unassigned

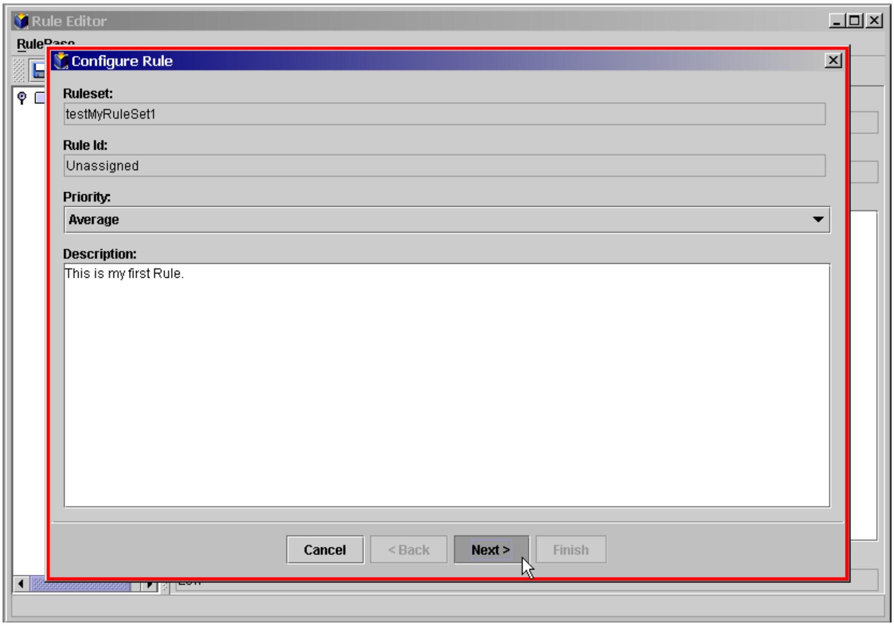
Priority: Medium

Description:

5 Select a Priority setting using the pull-down menu.

6 Type a description for the rule in the Description field. Click Next.

Figure 10-13. Creating a Rule - Configure Rule Wizard - Rule Characteristics Entered



Ruleset: testMyRuleSet1

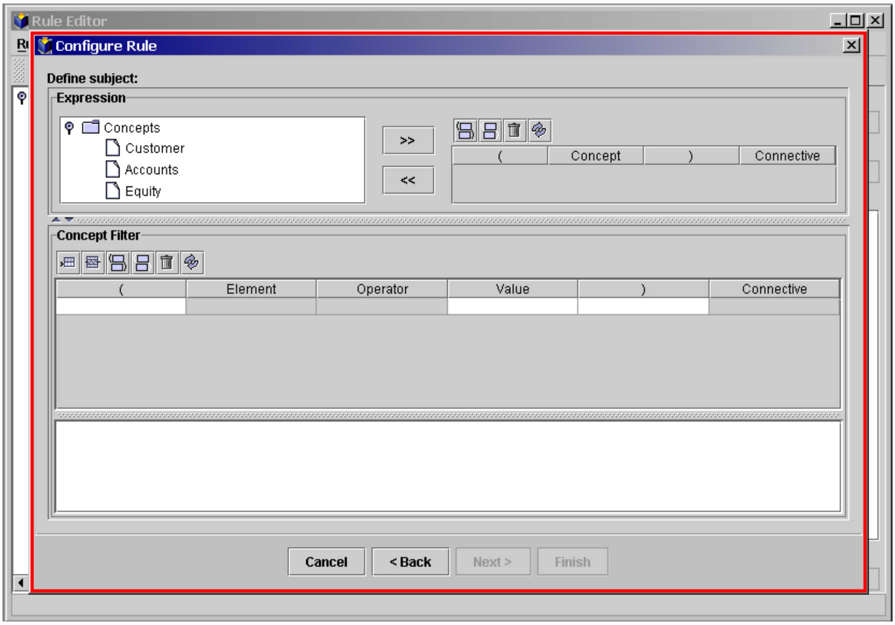
Rule Id: Unassigned

Priority: Average

Description: This is my first Rule.

7 The Configure Rule Wizard (Define subject:) displays.

Figure 10-14. Creating a Rule - Configure Rule Wizard - Define Subject - Initial



Within this editor, you create and modify the Concept components of the Subject, or antecedent, of the Rule. That is, the rule will be true and the Verb applied to the Object only if there are some instance data that satisfy the constraints described by the Subject Concept Filters.

The upper-left panel displays the available Subject Concepts. You select which Concepts contain Elements that you want within your Subject Expression.

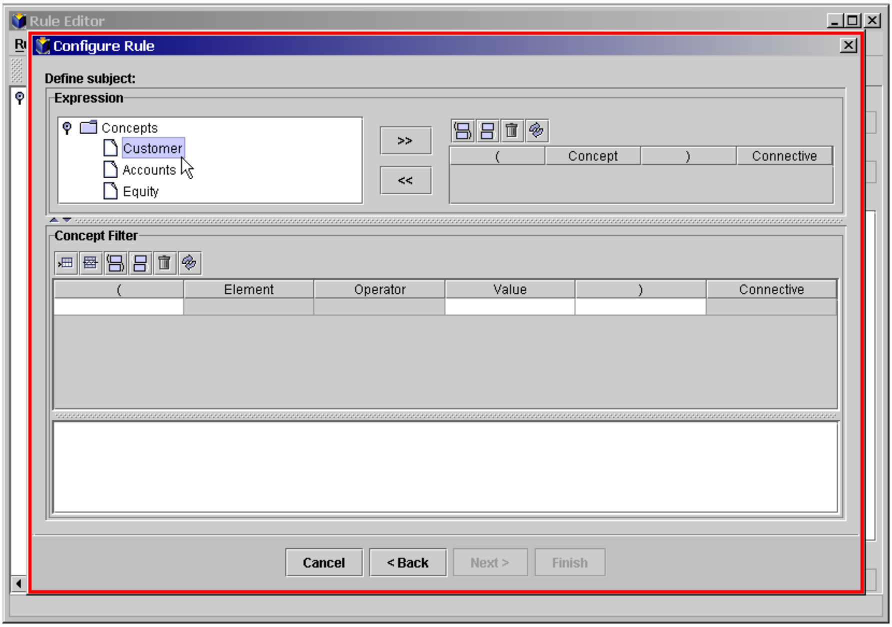
You build your Subject Expression using the expression-level Concept rows within the upper-right panel. You can form complex Subject Expressions by joining multiple Concepts using logical connectives.

Each expression-level Concept row contains at least one Element comparison. These display within the center panel. Element are compared using a variety of logical operators, and can themselves be joined using logical connectives to create complex Concept filters.

As you add and modify Concepts and Elements, your Subject Expression appears in the bottom panel.

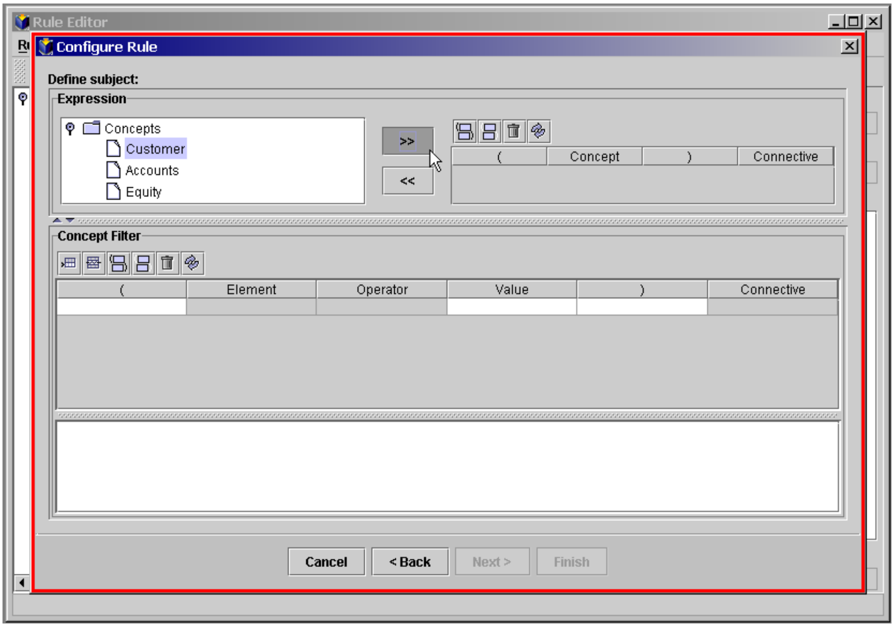
8 To proceed, highlight your selected Concept.

Figure 10-15. Creating a Rule - Configure Rule Wizard - Define Subject - Concept Selected



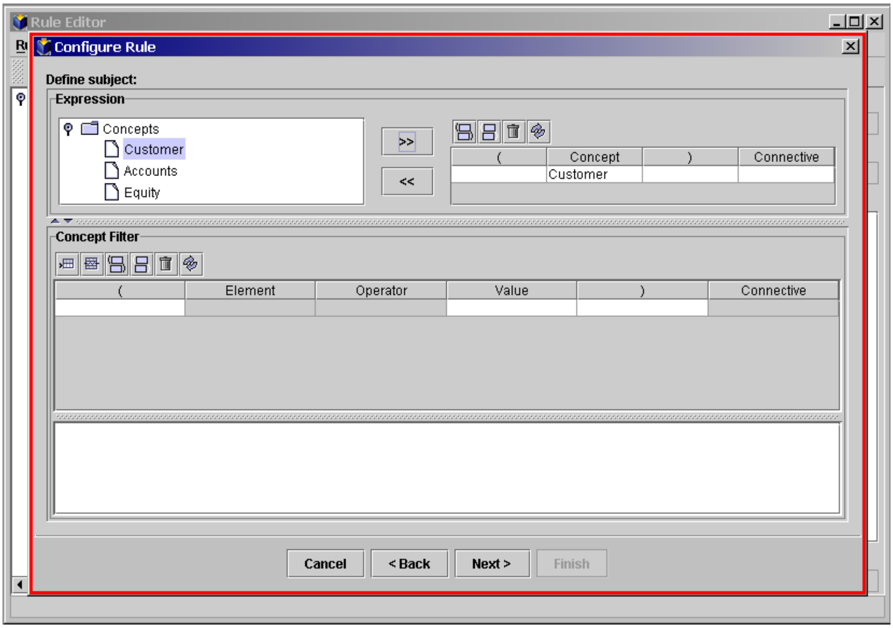
9 Click the >> on-screen button to move your selected Concept within the Subject Concept Row (the upper-right panel).

Figure 10-16. Creating a Rule - Configure Rule Wizard - Define Subject - Concept Moving



10 The selected Concept now displays within the Subject Concept Row (the upper-right panel).

Figure 10-17. Creating a Rule - Configure Rule Wizard - Define Subject - Concept Row Created



Define subject:

Expression

- Concepts
  - Customer
  - Accounts
  - Equity

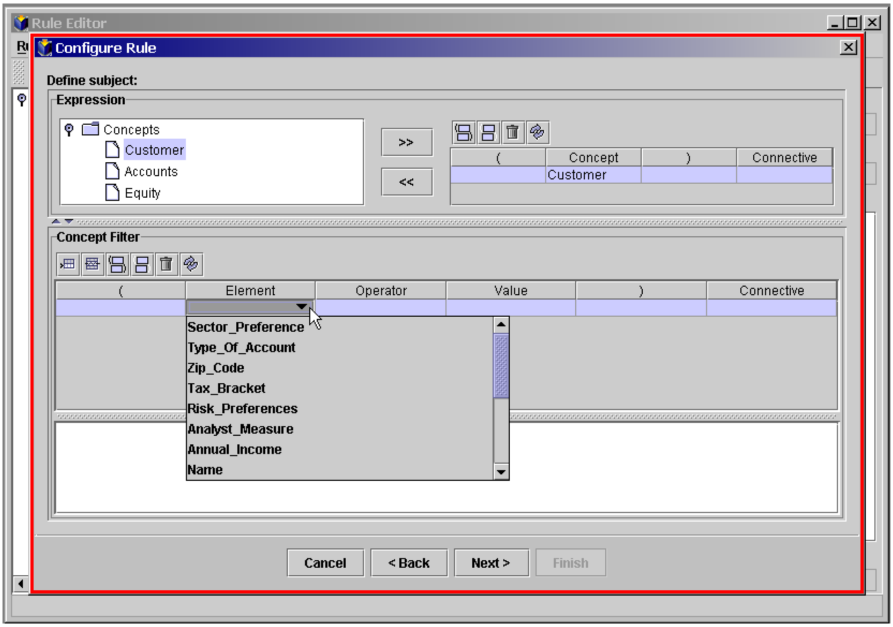
(	Concept	)	Connective
	Customer		

Concept Filter

(	Element	Operator	Value	)	Connective

11 Highlight the Concept Row. Within the Concept Filter central panel, click the field directly beneath the Element on-screen field button. A drop-down menu displays the Elements for the selected Concept.

Figure 10-18. Creating a Rule - Configure Rule Wizard - Define Subject - Concept Filter Element



Define subject:

Expression

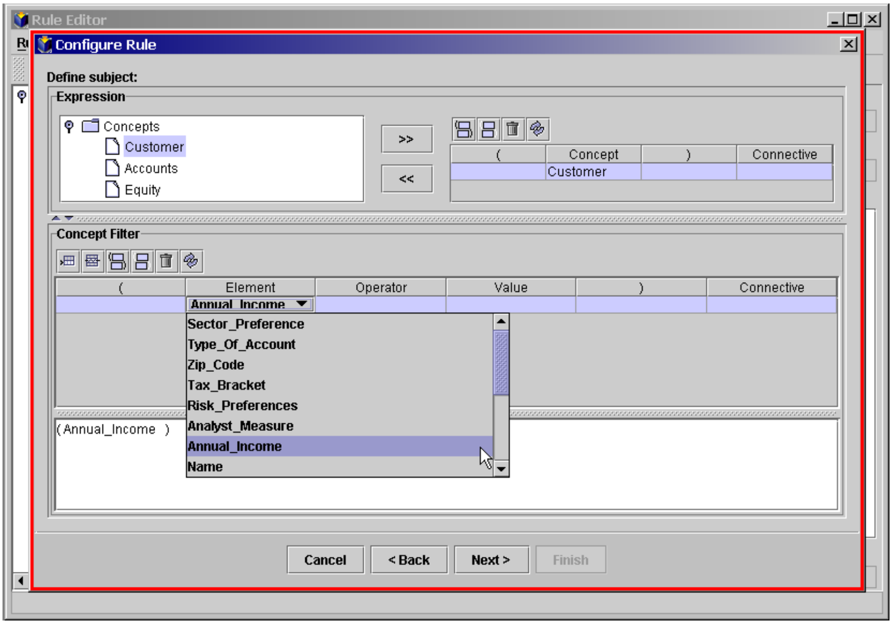
(	Concept	)	Connective
	Customer		

Concept Filter

(	Element	Operator	Value	)	Connective

12 Navigate the drop-down list until you locate the desired Element.

Figure 10-19. Creating a Rule - Configure Rule Wizard - Define Subject - Selecting Element



(	Concept	)	Connective
	Customer		

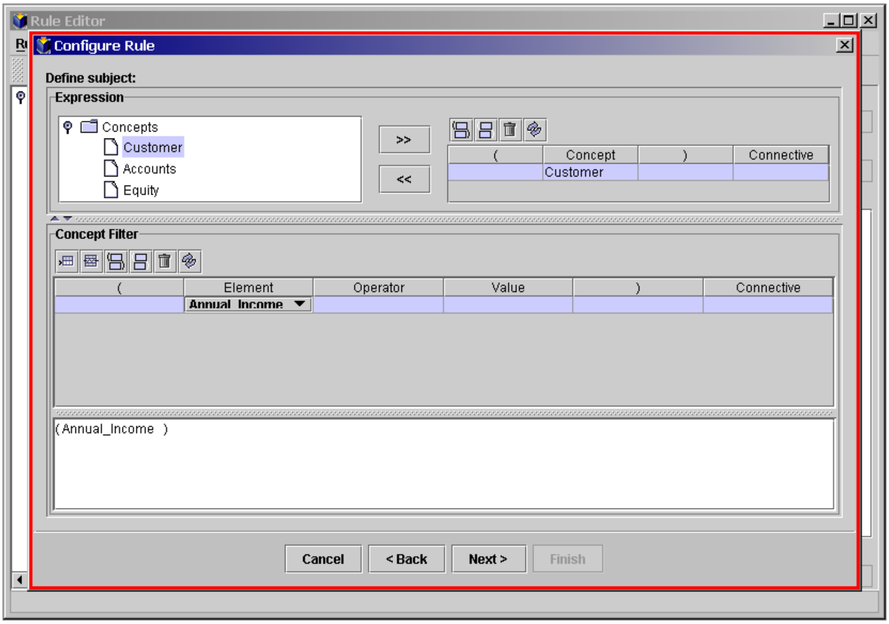
(	Element	Operator	Value	)	Connective
	Annual_Income				

- Sector_Preference
- Type_Of_Account
- Zip_Code
- Tax_Bracket
- Risk_Preferences
- Analyst_Measure
- Annual_Income
- Name

(Annual_Income)

13 Click to select this Element. The Element displays in the lower Subject Expression panel.

Figure 10-20. Creating a Rule - Configure Rule Wizard - Define Subject - Element Selected



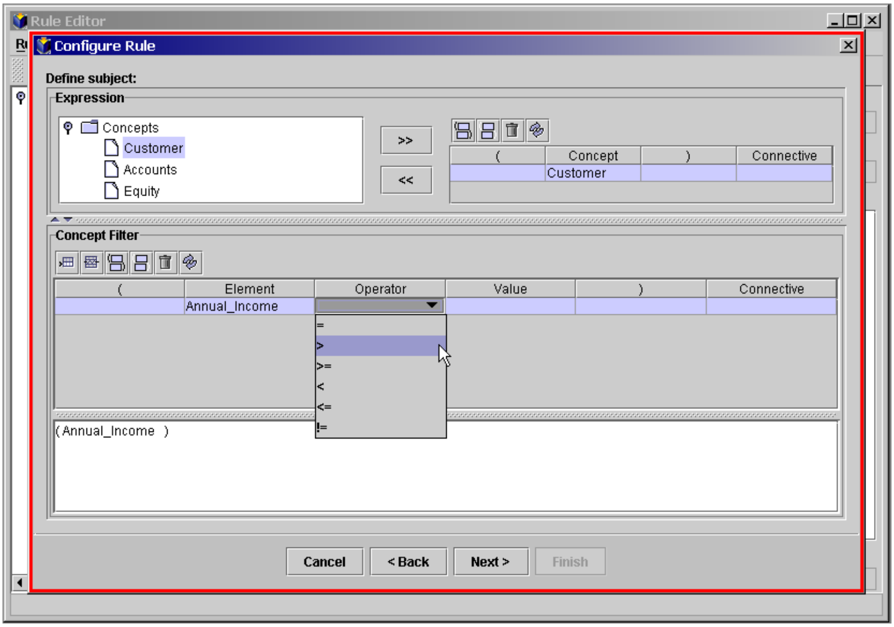
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income ▼				

```
(Annual_Income )
```

14 Within the Concept Filter central panel, click the field directly beneath the Operator on-screen field button. A drop-down menu displays the available logical comparison operators for the selected Element.

Figure 10-21. Creating a Rule - Configure Rule Wizard - Define Subject - Selecting Operator



(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income				

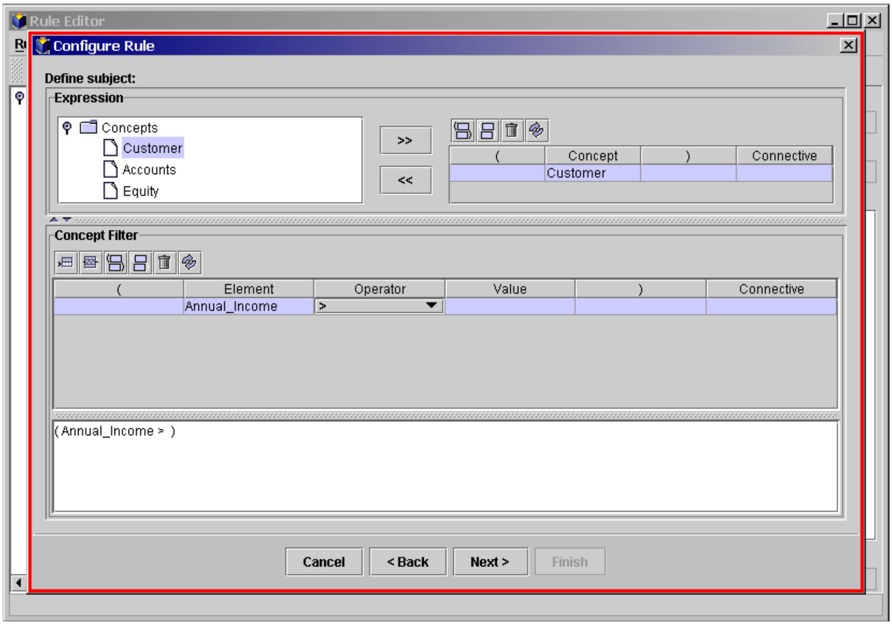
(Annual_Income)

Because of XSD restrictions, comparing a literal value against a string element requires you to quote the literal value using the “`value`” convention.

For information about using Filters, see *Using Filters on page 123*.

15 Within the drop-down operator list, click the desired comparison operator. The Subject Expression displays in the lower panel updates to reflect your choice.

Figure 10-22. Creating a Rule - Configure Rule Wizard - Define Subject - Operator Selected



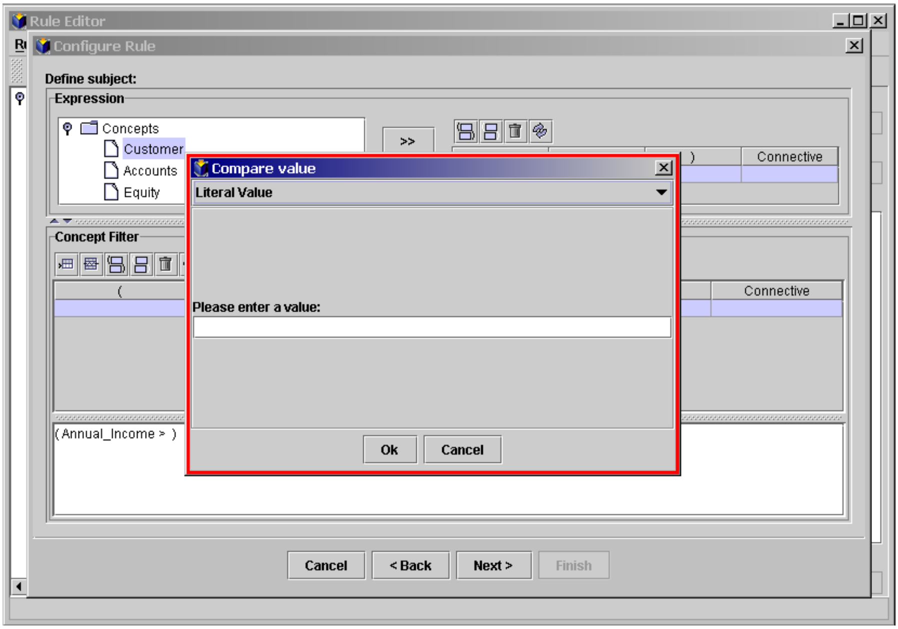
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income	>			

(Annual_Income >)

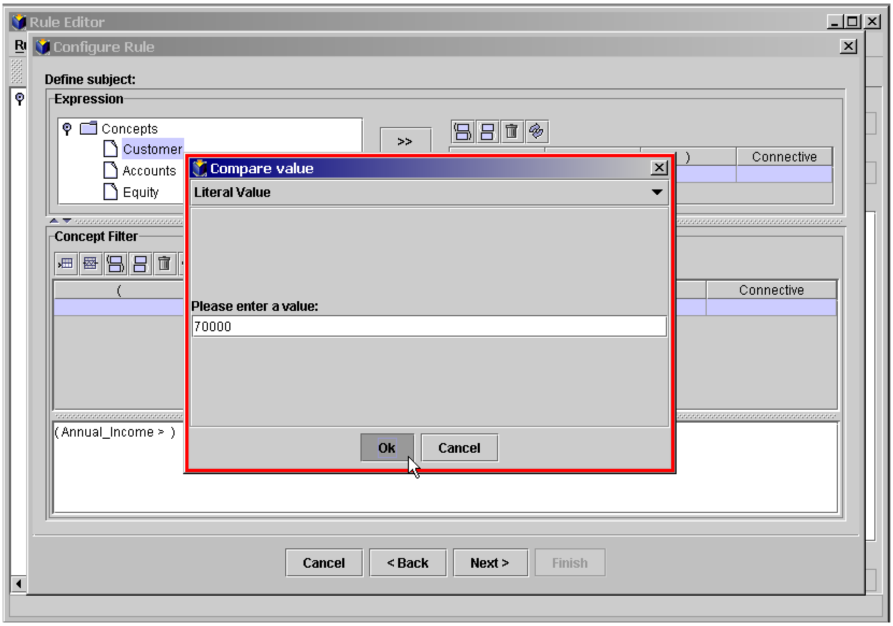
16 Within the Concept Filter central panel, click the field directly beneath the Value on-screen field button. The Compare Value dialog displays. From within this dialog, you can select a Literal Value Comparison or an Inter-Concept Comparison.

Figure 10-23. Creating a Rule - Configure Rule Wizard - Define Subject - Compare Value



17 A Literal Value Comparison is a simple, scalar comparison of the instance data of an Element against a single user-entered value. To create a Literal Value comparison, type or paste a value into the value field. Click the Ok button to continue.

Figure 10-24. Creating a Rule - Configure Rule Wizard - Define Subject - Compare Literal Value

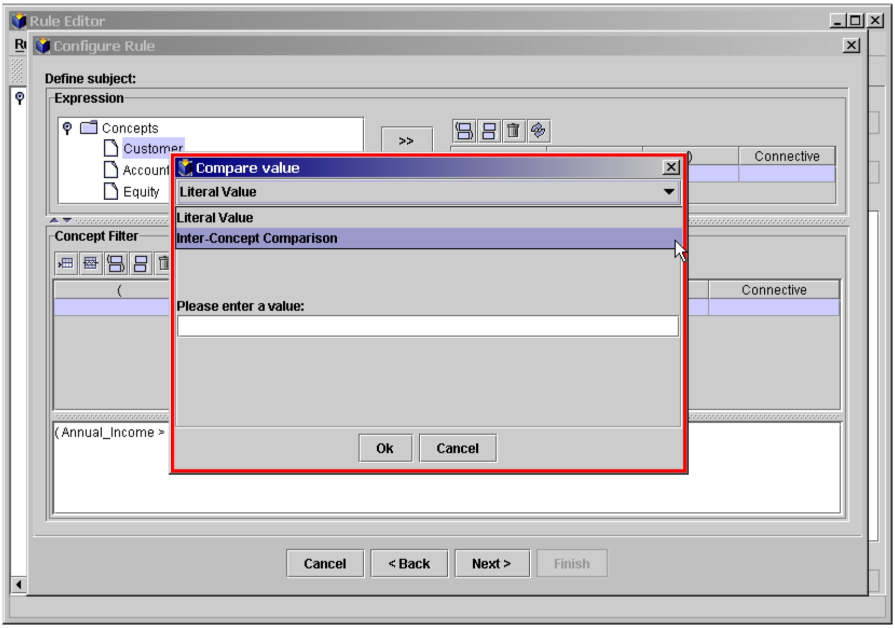


Please enter a value: 70000

(Annual_Income >)

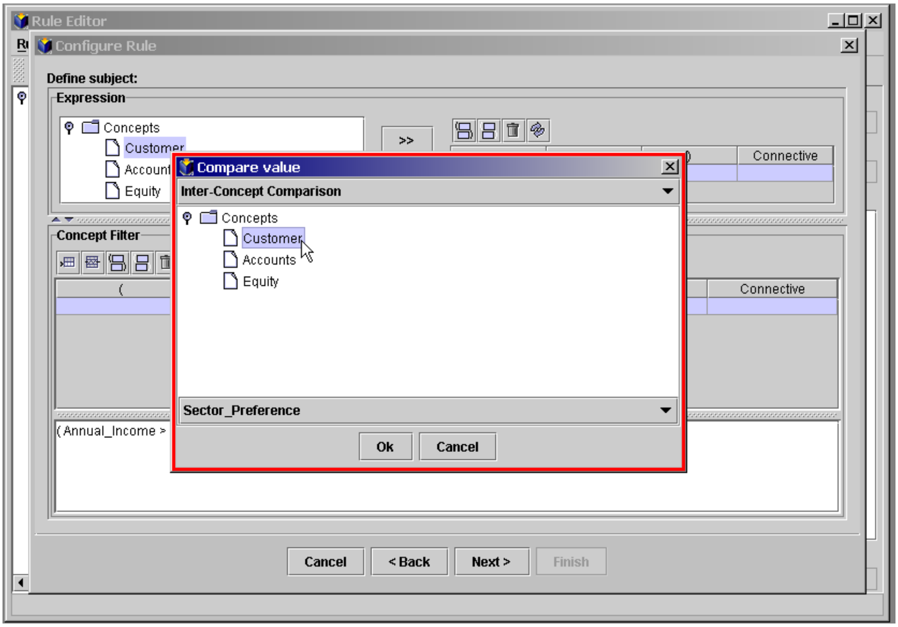
18 Alternatively, you can create an Inter-Concept Comparison. This is a complex, vector comparison of the instance data of one Element against the instance data of another Element. To create an Inter-Concept Comparison, click the Literal Value on-screen menu. A Drop-down menu displays. Click within this to select the Inter-Concept Comparison option.

Figure 10-25. Creating a Rule - Configure Rule Wizard - Define Subject - InterConcept Comparison



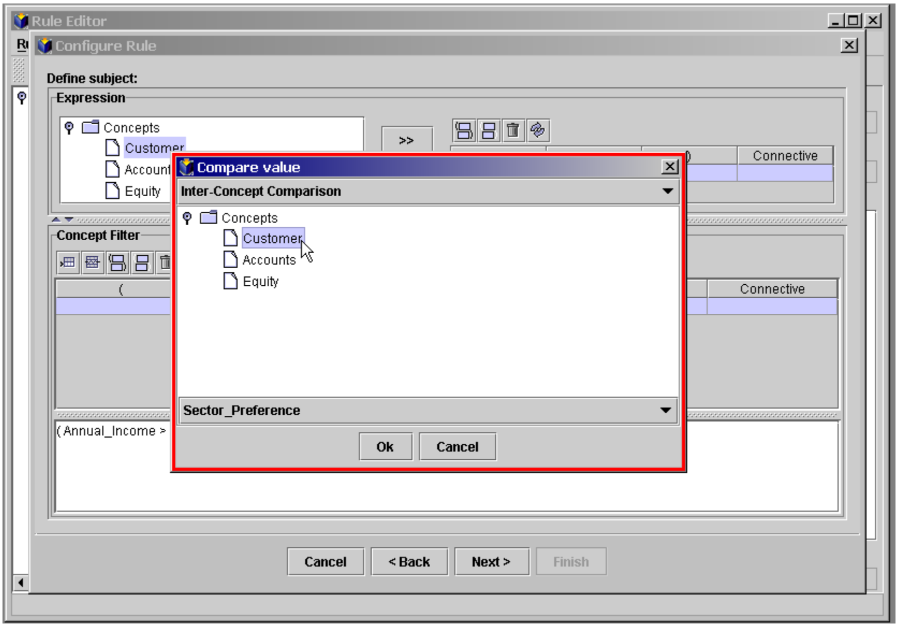
19 A list of available Concepts displays. Highlighting each Concept changes the lowest line of the Inter-Concept Comparison display.

Figure 10-26. Creating a Rule - Configure Rule Wizard - Define Subject - Comparison Concept



20 Click within this line to display a drop-down list of the available Elements for the selected Comparison Concept.

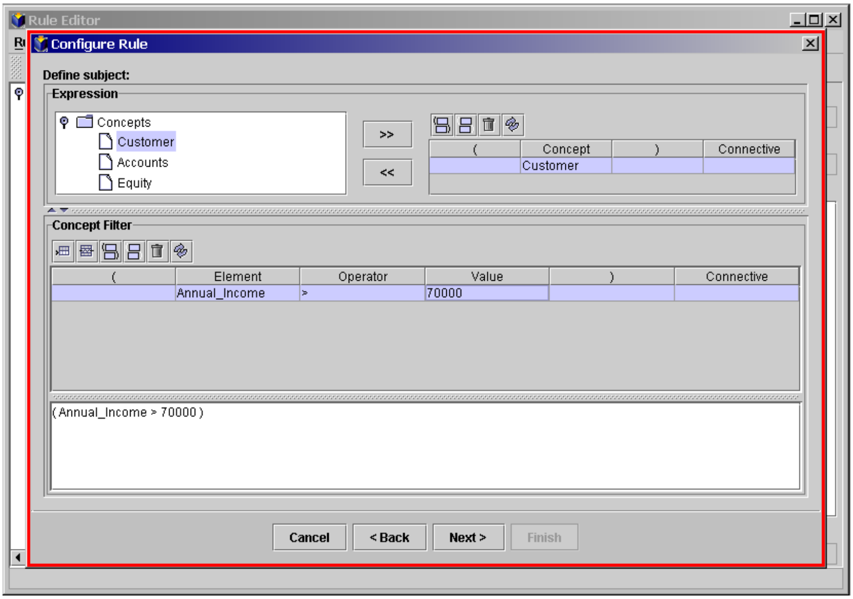
Figure 10-27. Creating a Rule - Configure Rule Wizard - Define Subject - Comparison Concept



Instance data from your previously selected Concept Element will be compared against instance data from this Element. Click Ok to continue.

21 Your selected Comparison Value displays in the lower Subject Expression panel.

Figure 10-28. Creating a Rule - Configure Rule Wizard - Define Subject - Literal Value

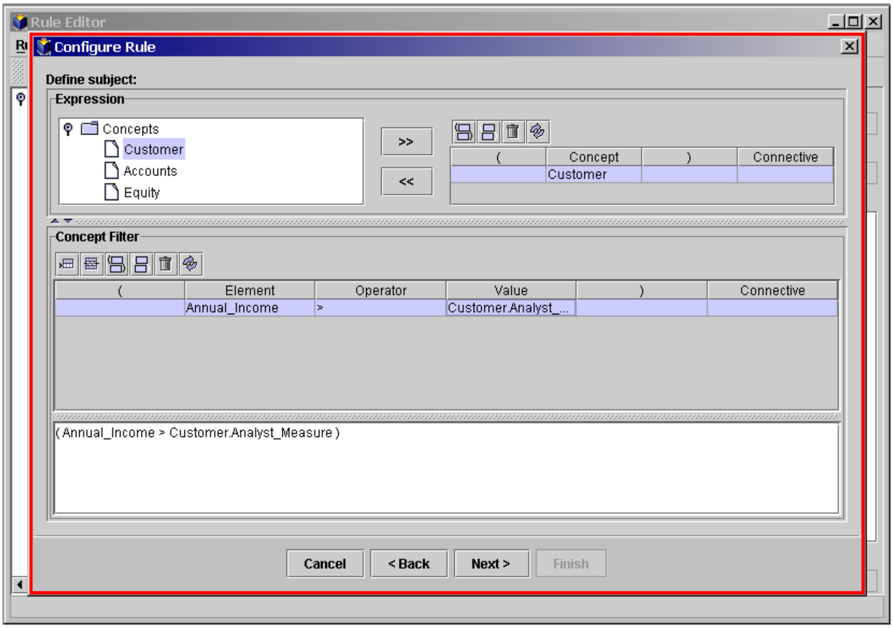


(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		

(Annual_Income > 70000)

Figure 10-29. Creating a Rule - Configure Rule Wizard - Define Subject - InterConcept Comparison



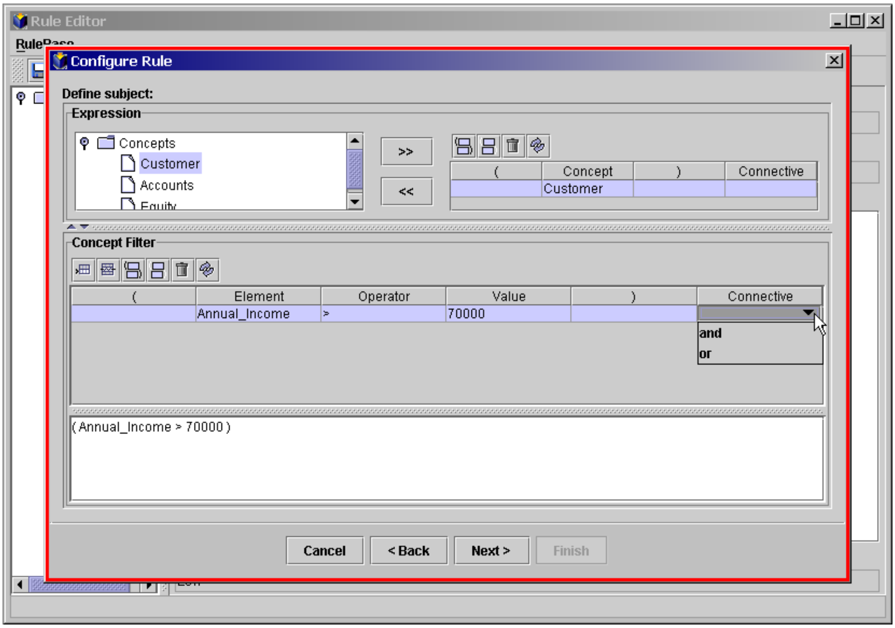
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	Customer.Analyst_...		

(Annual_Income > Customer.Analyst_Measure)

22 To add another Concept Filter Comparison, click the field directly beneath the Connective button within the central Concept Filter panel.

Figure 10-30. Creating a Rule - Configure Rule Wizard - Define Subject - Add Filter Row



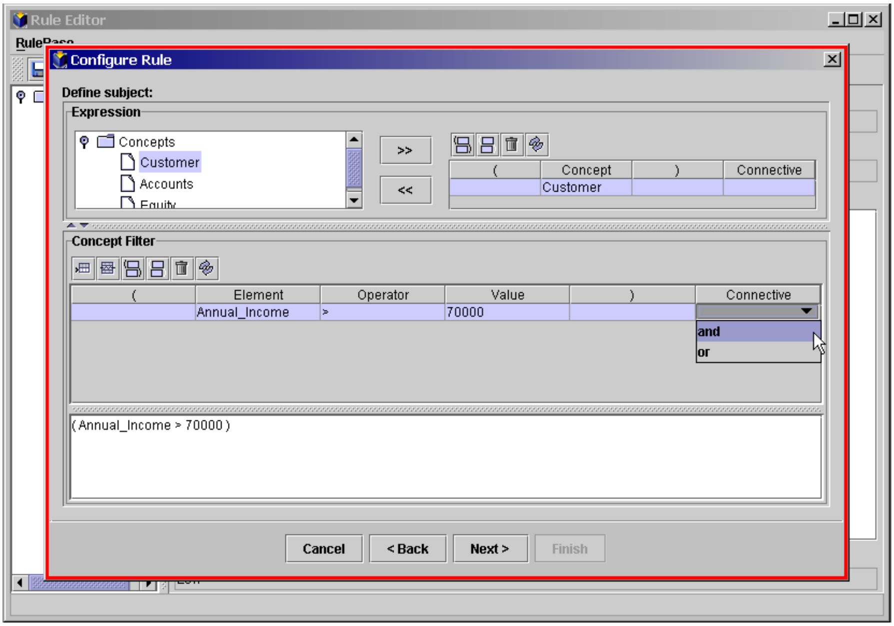
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		

(Annual_Income > 70000)

23 Select the desired logical Connective from within the drop-down list of Connectives.

Figure 10-31. Creating a Rule - Configure Rule Wizard - Define Subject - Connective Selected



Define subject:

Expression

(	Concept	)	Connective
	Customer		

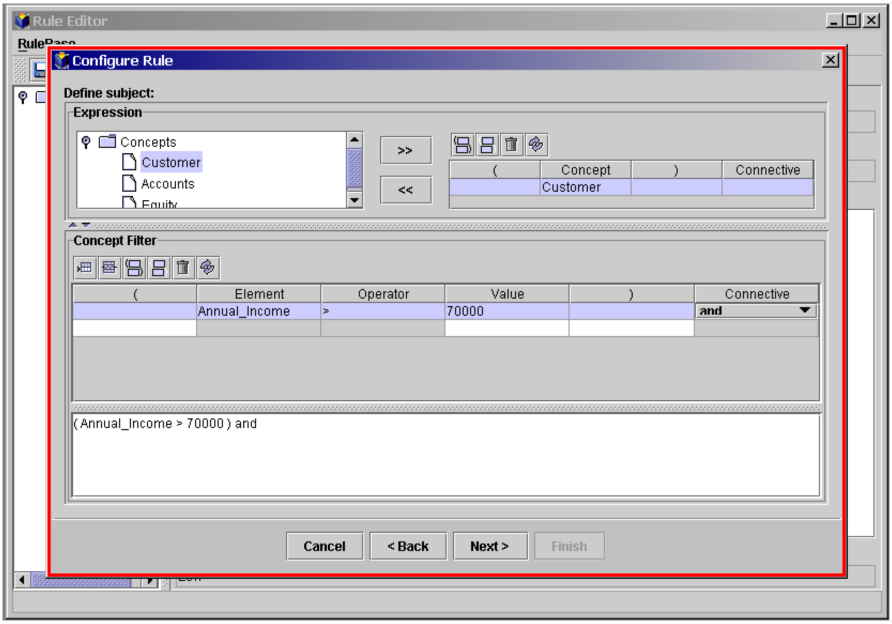
Concept Filter

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		and or

(Annual_Income > 70000)

24 The logical Connective now displays in the lower Subject Expression panel and a new Concept Filter row displays within the central panel.

Figure 10-32. Creating a Rule - Configure Rule Wizard - Define Subject - New Concept Filter Row



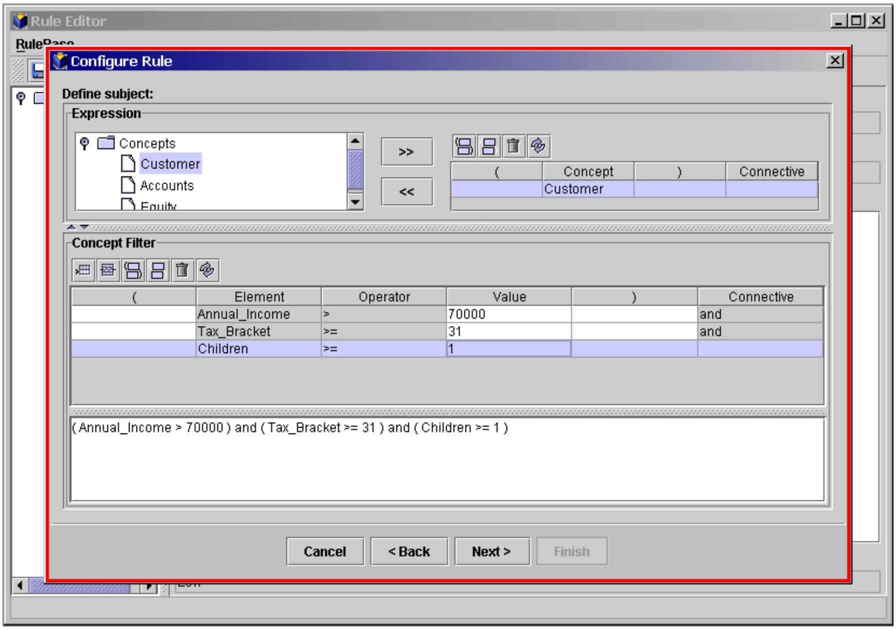
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		and

(Annual_Income > 70000) and

25 Repeat Steps 11 through 21 as required to define new Concept Filter Comparison Rows. You can add as many Concept Filter Comparison Rows as desired with the limitation that the choice of Elements is limited to the Concept selected in the upper-right Subject Concept panel. All Concept Filter Comparisons display in the lower Subject Expression panel.

Figure 10-33. Creating a Rule - Configure Rule Wizard - Define Subject - All Concept Filter Rows



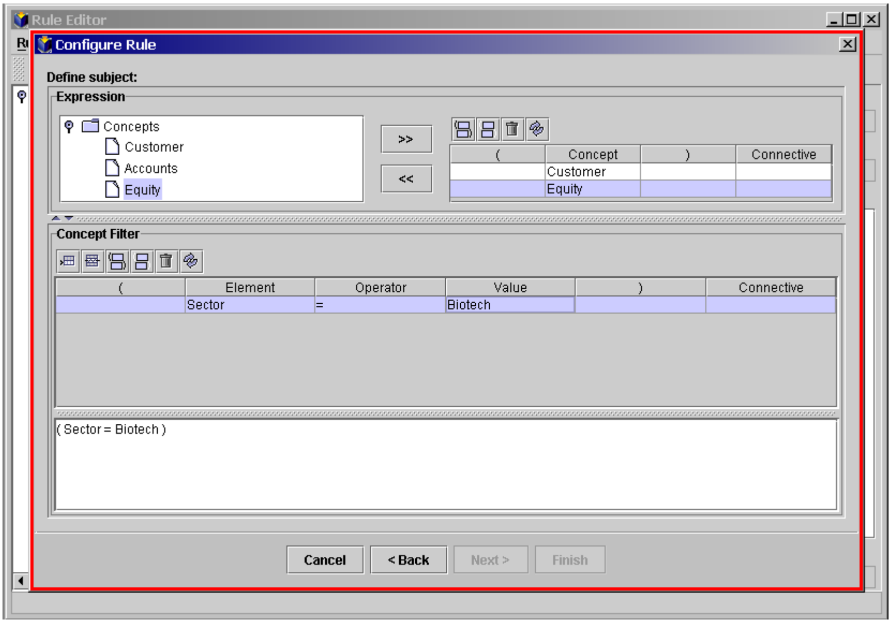
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		and
	Tax_Bracket	>=	31		and
	Children	>=	1		

(Annual_Income > 70000) and (Tax_Bracket >= 31) and (Children >= 1)

26 To add another Concept to the Subject Concept Row, repeat Steps 8 through 10 as required. The selected Concepts display within the upper-right panel.

Figure 10-34. Creating a Rule - Configure Rule Wizard - Define Subject - Multiple Concept Rows



Define subject:

Expression

(	Concept	)	Connective
	Customer		
	Equity		

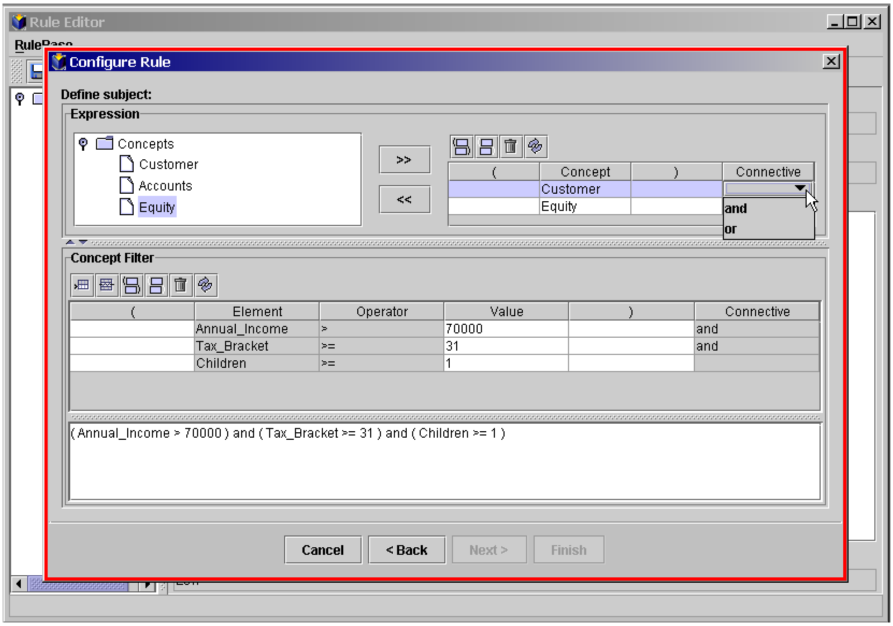
Concept Filter

(	Element	Operator	Value	)	Connective
	Sector	=	Biotech		

(Sector = Biotech)

27 To join Subject Row Concepts using logical connectives you must select within the upper-right Subject Concept panel. Highlight the Connective field button between your two selected Concepts.

Figure 10-35. Creating a Rule - Configure Rule Wizard - Define Subject - Concept Connective



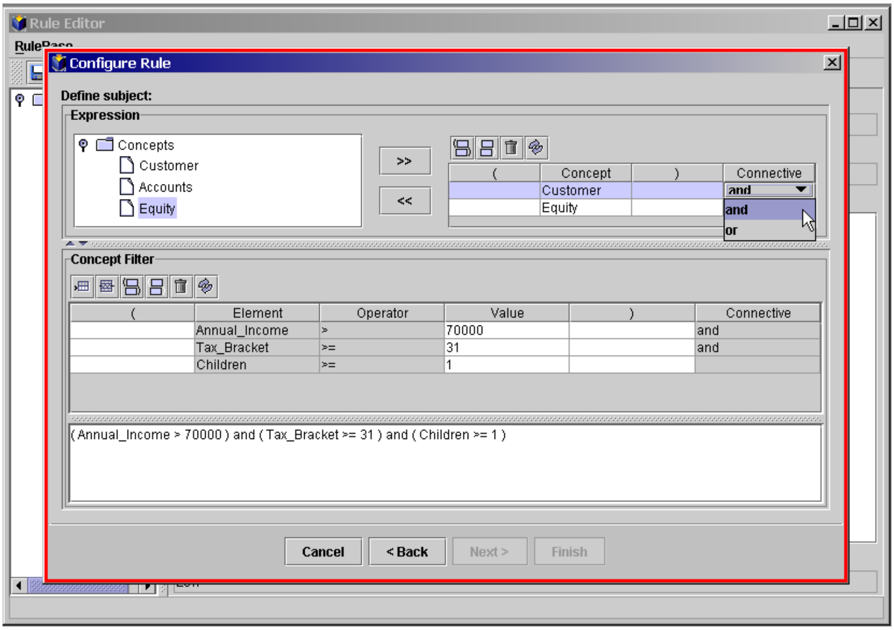
(	Concept	)	Connective
	Customer		
	Equity		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		and
	Tax_Bracket	>=	31		and
	Children	>=	1		

(Annual_Income > 70000) and (Tax_Bracket >= 31) and (Children >= 1)

28 Click to select your desired connective from within the drop-down list of logical Connectives.

Figure 10-36. Creating a Rule - Configure Rule Wizard - Define Subject - And Concept Connective



Define subject:

Expression

- Concepts
  - Customer
  - Accounts
  - Equity

(	Concept	)	Connective
	Customer		and
	Equity		

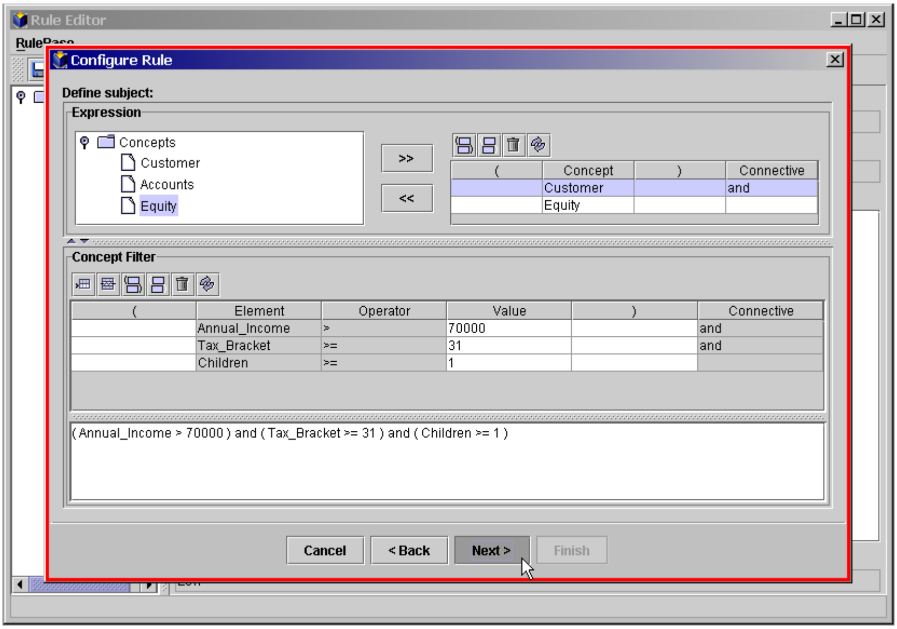
Concept Filter

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		and
	Tax_Bracket	>=	31		and
	Children	>=	1		

```
( Annual_Income > 70000 ) and ( Tax_Bracket >= 31 ) and ( Children >= 1 )
```

29 When you have finished adding Concept Rows and Concept Filters to the Subject Expression, it is time to move on to indicate which Verbs to use. Click the Next> button to continue.

Figure 10-37. Creating a Rule - Configure Rule Wizard - Define Subject - Completed



(	Concept	)	Connective
	Customer		and
	Equity		

(	Element	Operator	Value	)	Connective
	Annual_Income	>	70000		and
	Tax_Bracket	>=	31		and
	Children	>=	1		

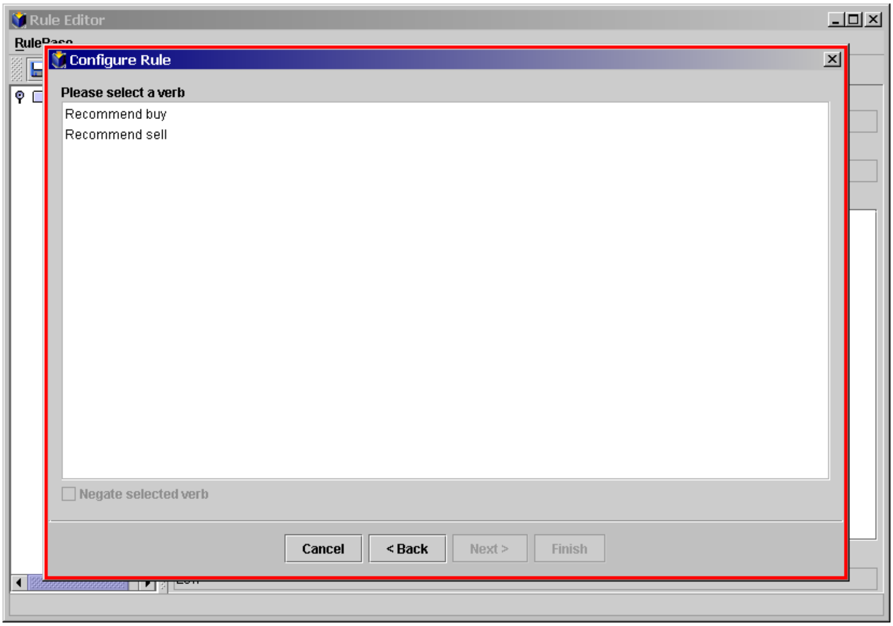
(Annual_Income > 70000) and (Tax_Bracket >= 31) and (Children >= 1)

The Concept Filter-level and connective requires that all filter conditions so joined be satisfied for the component to evaluate true during the rule processing phase. Selecting multiple filtered Properties within a Concept will introduce more constraints into that component of the rule.

On the *Subject* side of the rule, this will make the rule less likely to activate. On the *Object* side of the rule, this will reduce the number of instances onto which the *Verb* will be applied. So even if a rule activates, with strict filters on *Objects*, the number of items satisfying the filter conditions may decline to zero, the rule therefore acts on zero instances, and so has no real effect.

30 The Configure Rule Wizard now displays all Verbs available to Rules defined within this Ruleset.

Figure 10-38. Creating a Rule - Configure Rule Wizard - Verb - Initial



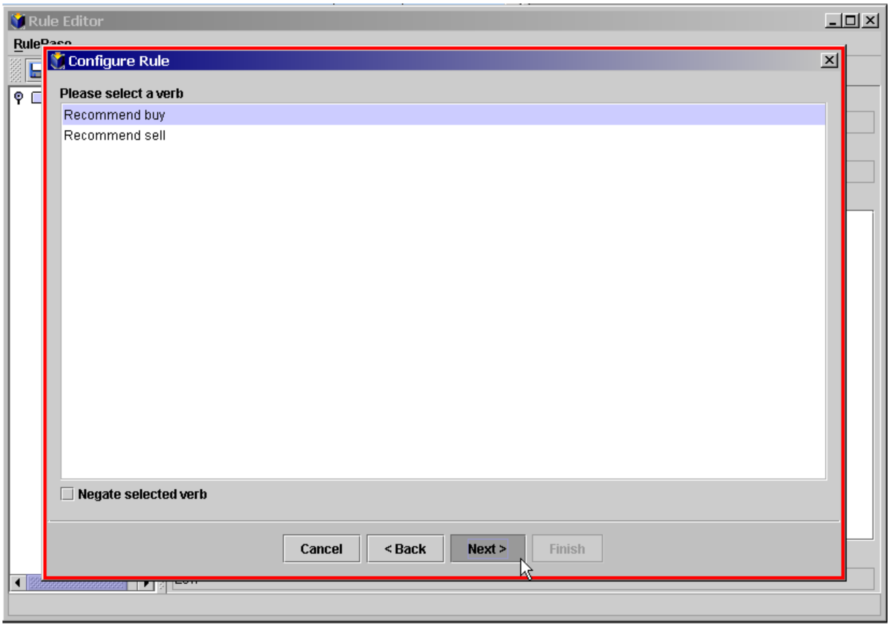
Please select a verb

Recommend buy
Recommend sell

□ Negate selected verb

31 Click on a Verb to activate it for this Rule. The selected Verb highlights. Click the Next> button to continue.

Figure 10-39. Creating a Rule - Configure Rule Wizard - Verb - Selected



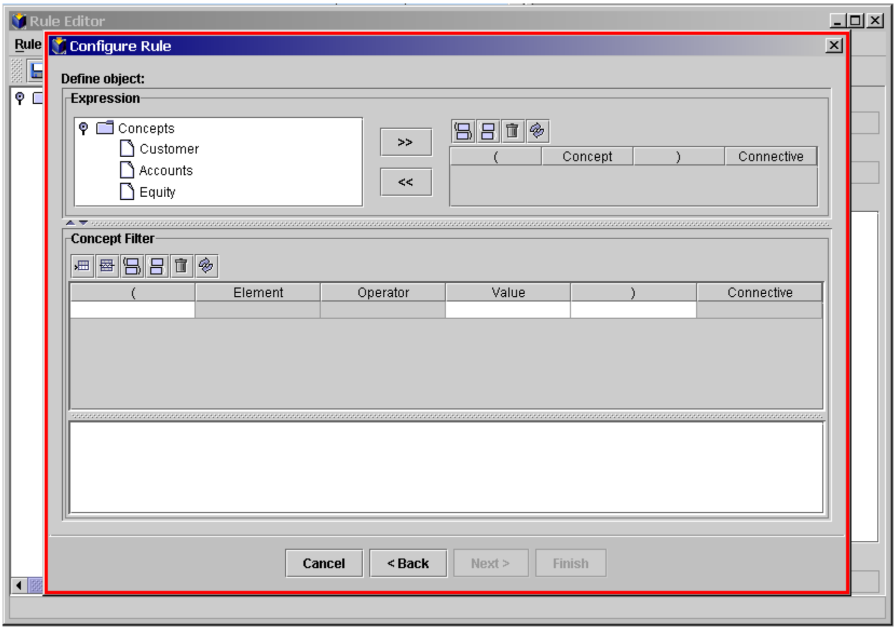
Please select a verb

Recommend buy
Recommend sell

□ Negate selected verb

32 The Configure Rule Wizard (Define object:) displays.

Figure 10-40. Creating a Rule - Configure Rule Wizard - Define Object - Initial



Within this editor, you create and modify the Concept components of the Object, or consequent, of the Rule. That is, the Verb will be applied to whatever instance data satisfy the constraints described by the Object Concept Filters.

The upper-left panel displays the available Object Concepts. You select which Concepts contain Elements that you want within your Object Expression.

You build your Object Expression using the expression-level Concept rows within the upper-right panel. You can form complex Object Expressions by joining multiple Concepts using logical connectives.

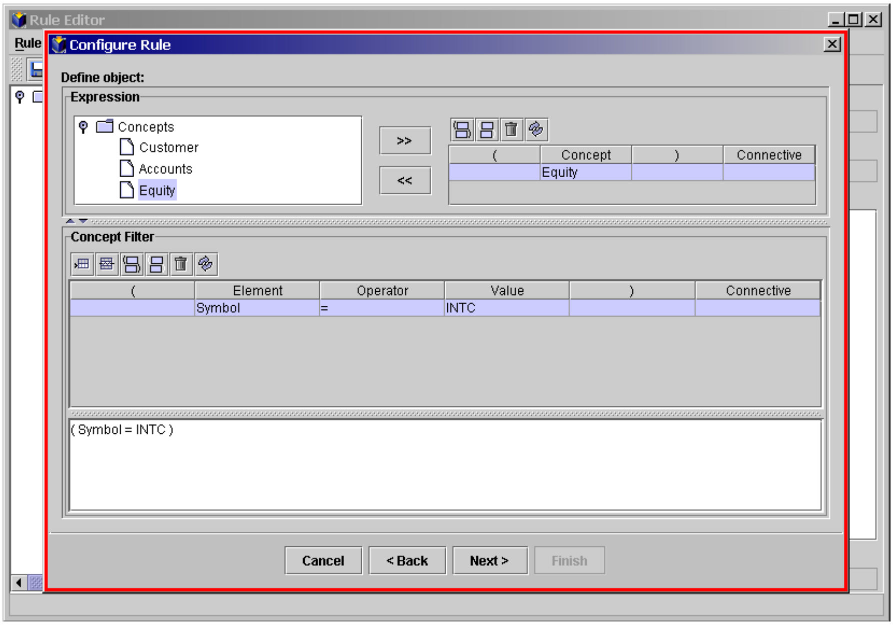
Each expression-level Concept row contains at least one Element comparison. These display within the center panel. Element are compared using a variety of logical operators, and can themselves be joined using logical connectives to create complex Concept filters.

As you add and modify Concepts and Elements, your Object Expression appears in the bottom panel.

33 Creating Object Expressions use the same syntax and construction methods as creating Subject Expressions. Repeat Steps 7-29 as necessary to add more Concept Rows and Concept Filters. Object Expressions display in the lower panel similarly to Subject Expressions.

Inter-Concept Comparisons are not currently supported for Object Expression Concept Filters.

Figure 10-41. Creating a Rule - Configure Rule Wizard - Define Object - Complete

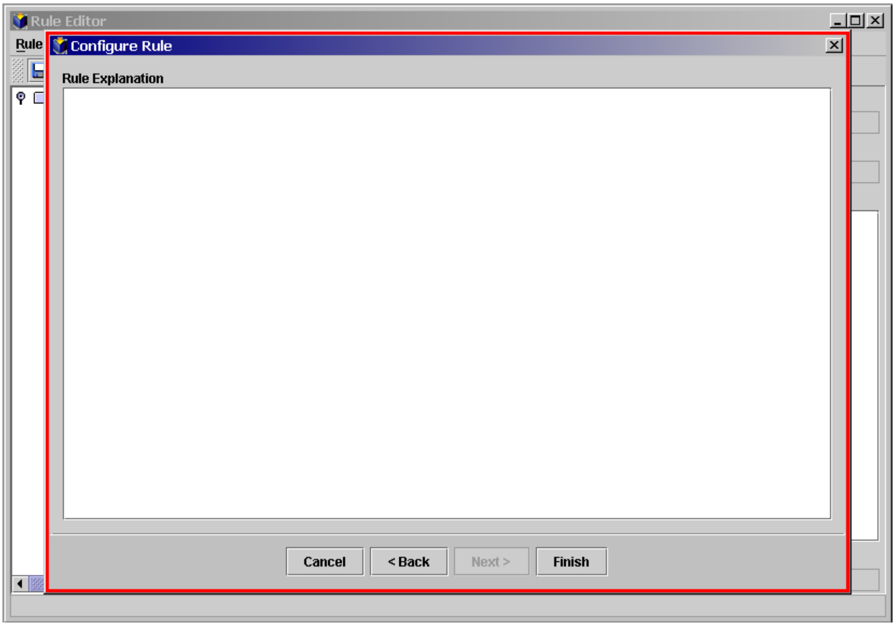


(Symbol = INTC)

The Verb applies the action to all ontology objects described by all the instances of the Object. An Object with too many conditions, therefore, might not be true for any instances in the Knowledge Base. In this case, the Verb would not apply to any objects and the rule, although activated, would not return a recommendation. You should exercise caution when constructing complex Objects..

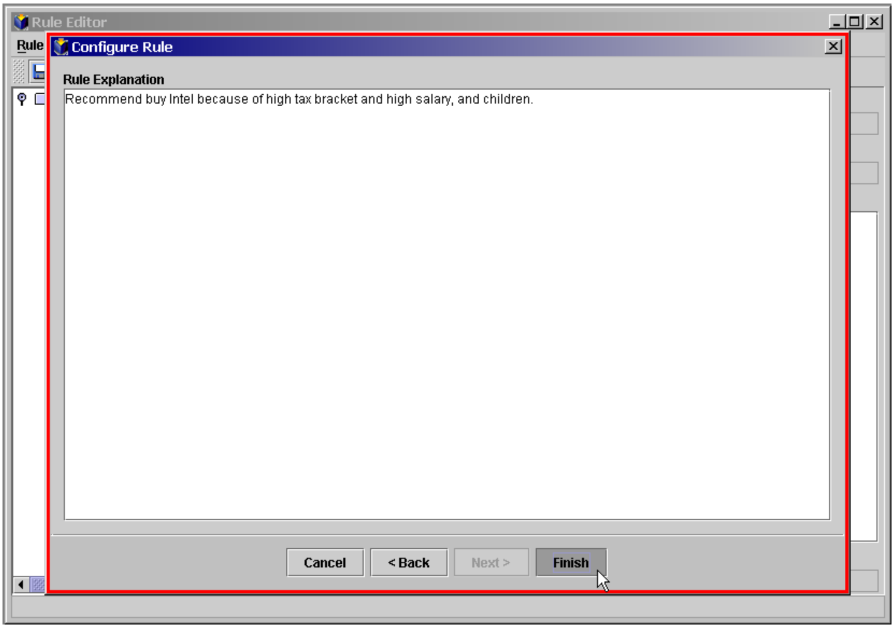
34 When you are finished, click the Next> button to continue. The Rule Explanation Dialog displays.

Figure 10-42. Creating a Rule - Configure Rule Wizard - Rule Explanation Displays



35 Type or paste text into the Rule Explanation text box. To complete creating the Rule, click the Finish button.

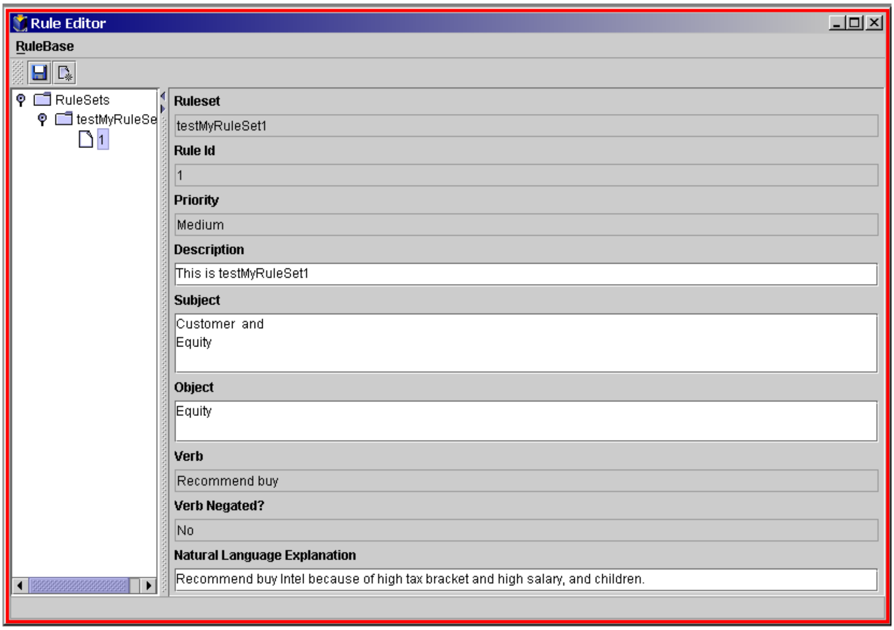
Figure 10-43. Creating a Rule - Configure Rule Wizard - Rule Explanation Complete



Recommend buy Intel because of high tax bracket and high salary, and children.

36 The Configure Rule Wizard closes and the Rule Editor display updates, showing the newly created Rule. You have now finished creating a new Rule.

Figure 10-44. Creating a Rule - Rule Editor - Rule Complete



Ruleset
testMyRuleSet1

Rule Id
1

Priority
Medium

Description
This is testMyRuleSet1

Subject
Customer and
Equity

Object
Equity

Verb
Recommend buy

Verb Negated?
No

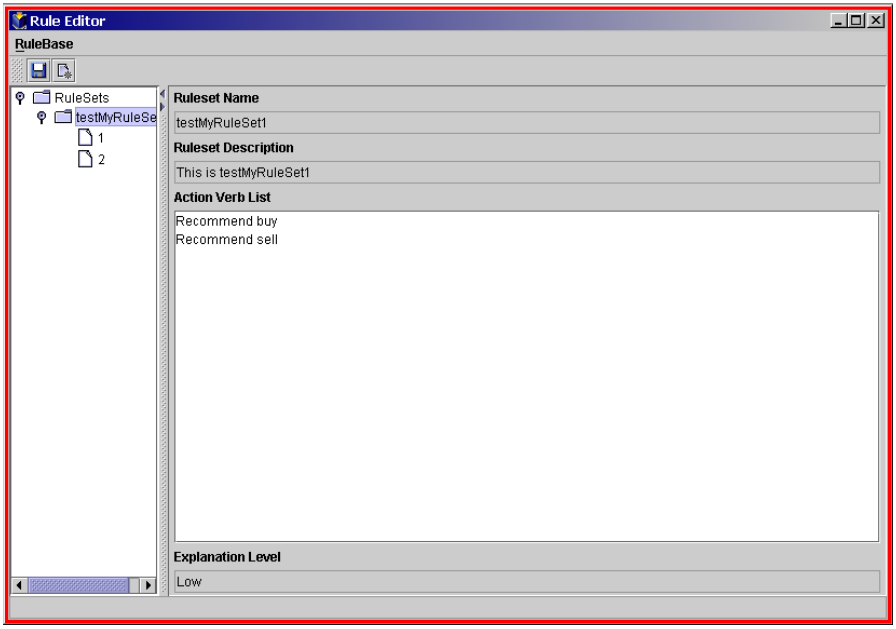
Natural Language Explanation
Recommend buy Intel because of high tax bracket and high salary, and children.

This does not save the rule to the knowledge base. You must select **Knowledge Base > Save** in the Knowledge Broker Launcher to save the rule to the Knowledge Base.

Editing a Rule

1 To edit an existing Rule, open the Rule Editor, if it is not currently open.

Figure 10-45. Editing a Rule - Rule Editor Display - No Rule Selected



Ruleset Name
testMyRuleSet1

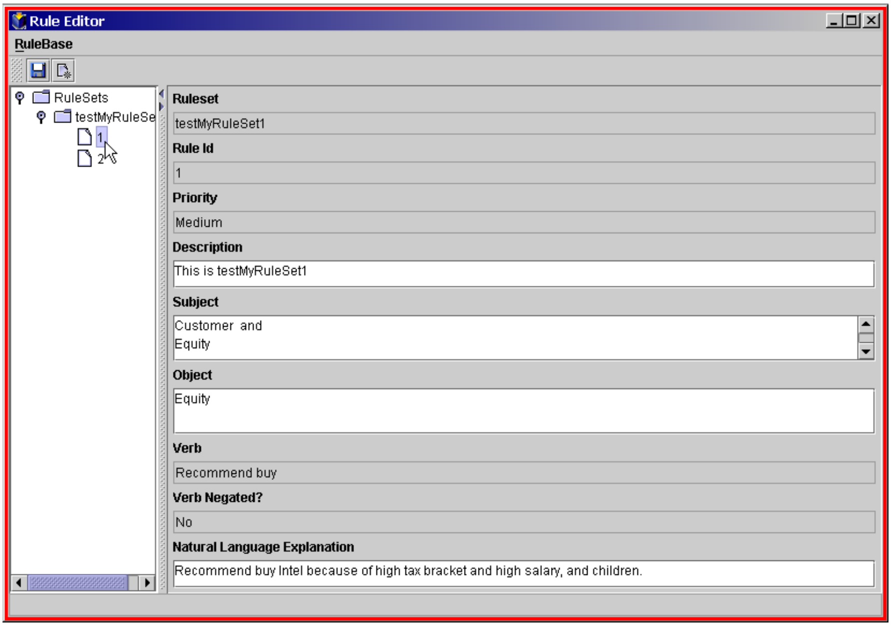
Ruleset Description
This is testMyRuleSet1

Action Verb List
Recommend buy
Recommend sell

Explanation Level
Low

2 Select the Rule from within the left-hand Ruleset panel.

Figure 10-46. Editing a Rule - Rule Editor Display - Rule Selected for Editing



Ruleset
testMyRuleSet1

Rule Id
1

Priority
Medium

Description
This is testMyRuleSet1

Subject
Customer and
Equity

Object
Equity

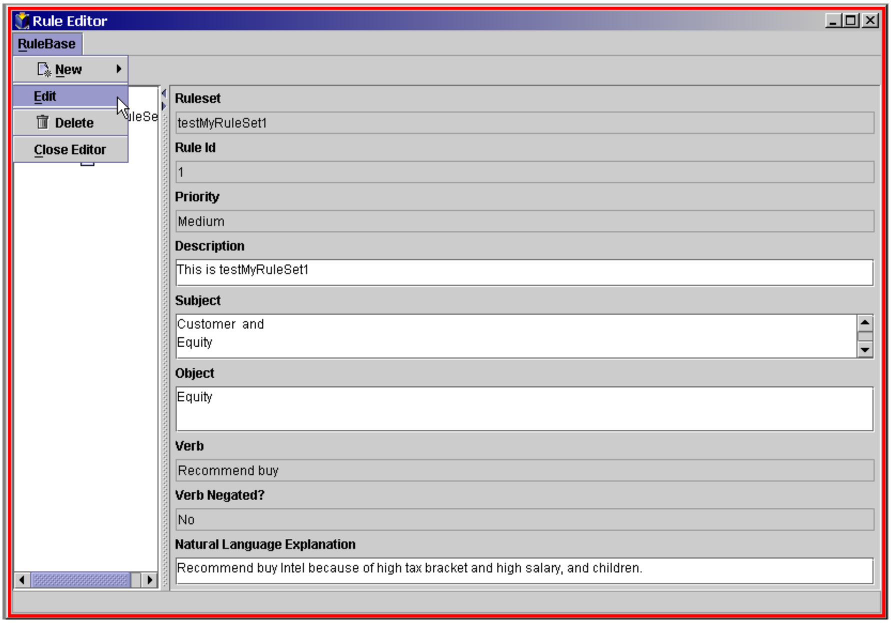
Verb
Recommend buy

Verb Negated?
No

Natural Language Explanation
Recommend buy Intel because of high tax bracket and high salary, and children.

3 Select the menu item **RuleBase > Edit.**

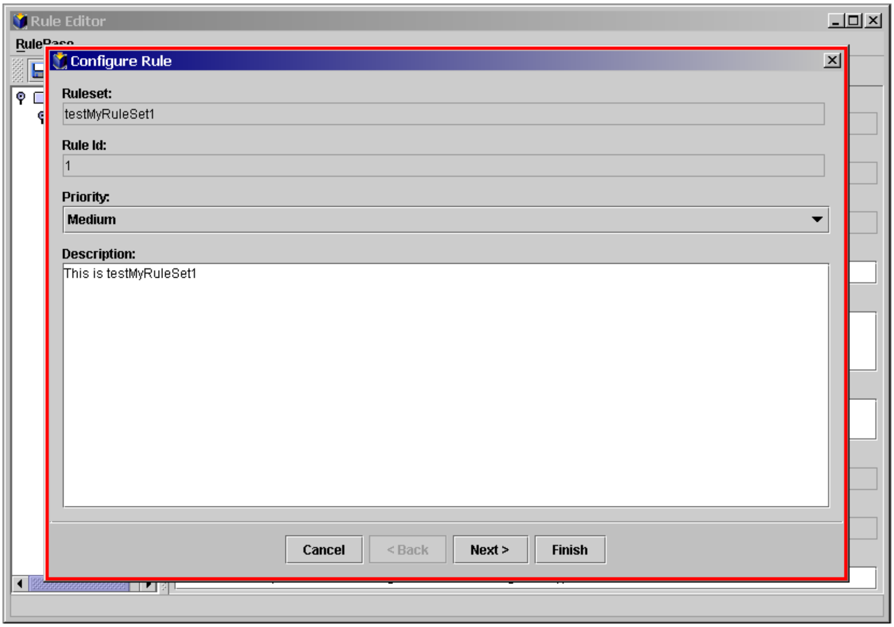
Figure 10-47. Editing a Rule - Rule Editor Display - Menu Item "Edit" Selected



Ruleset
testMyRuleSet1
Rule Id
1
Priority
Medium
Description
This is testMyRuleSet1
Subject
Customer and
Equity
Object
Equity
Verb
Recommend buy
Verb Negated?
No
Natural Language Explanation
Recommend buy Intel because of high tax bracket and high salary, and children.

4 The Configure Rule Wizard displays. Proceed using the same techniques described within the section *Creating a Rule on page 360.*

Figure 10-48. Editing a Rule - Rule Editor Display - Configure Rule Wizard Displays



Ruleset: testMyRuleSet1

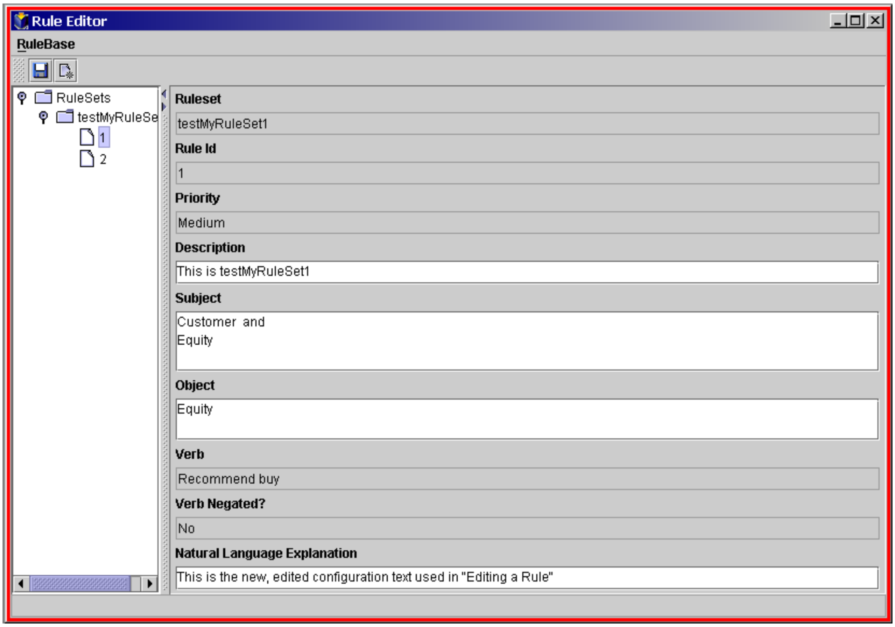
Rule Id: 1

Priority: Medium

Description: This is testMyRuleSet1

5 Click the Finish button when you have completed editing the Rule. The Rule Editor displays the modified Rule.

Figure 10-49. Editing a Rule - Rule Editor Displays Modified Rule



Ruleset
testMyRuleSet1

Rule Id
1

Priority
Medium

Description
This is testMyRuleSet1

Subject
Customer and
Equity

Object
Equity

Verb
Recommend buy

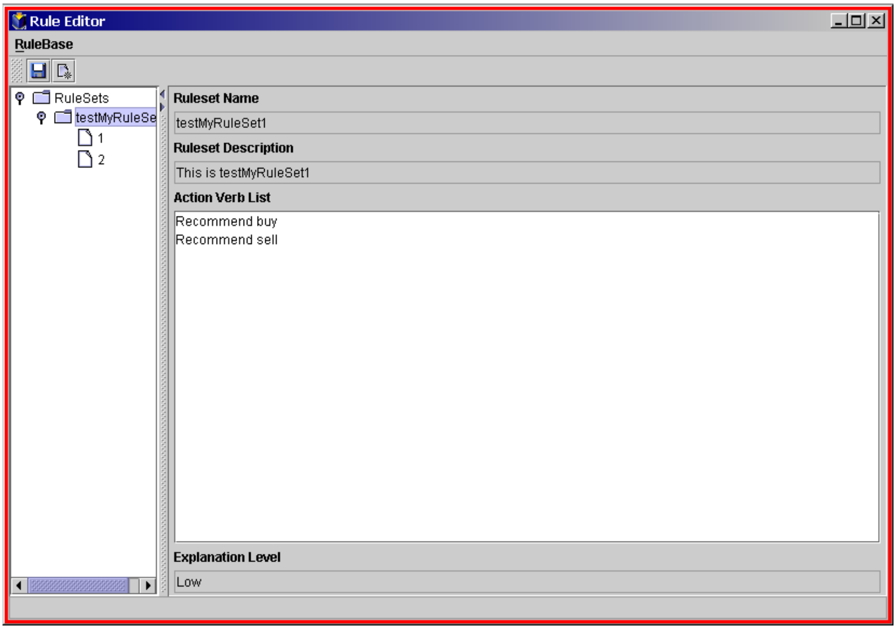
Verb Negated?
No

Natural Language Explanation
This is the new, edited configuration text used in "Editing a Rule"

Deleting a Rule

1 To delete an existing Rule, open the Rule Editor, if it is not currently open.

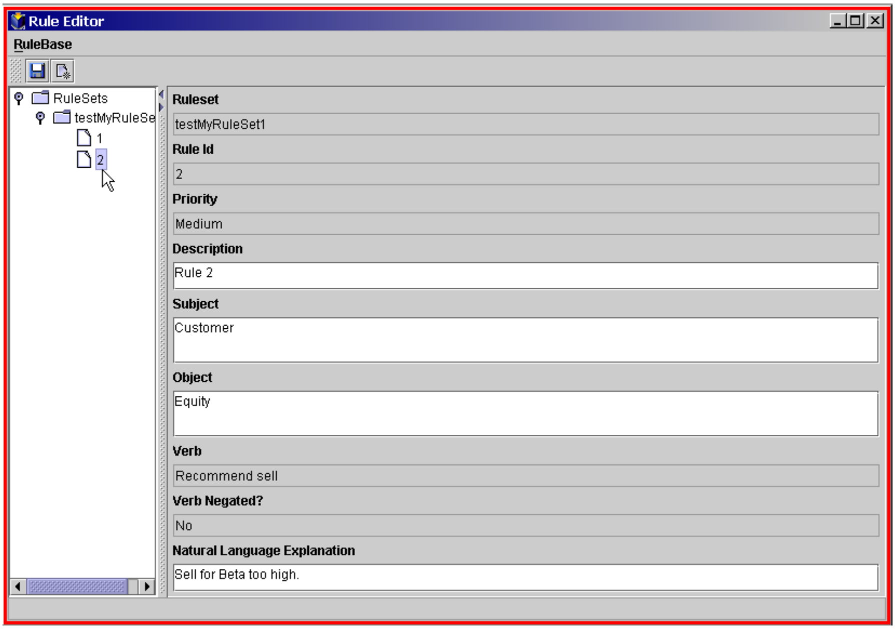
Figure 10-50. Deleting a Rule - Rule Editor Display - No Rule Selected



Ruleset Name
testMyRuleSet1
Ruleset Description
This is testMyRuleSet1
Action Verb List
Recommend buy
Recommend sell
Explanation Level
Low

2 Select the Rule from within the left-hand Ruleset panel.

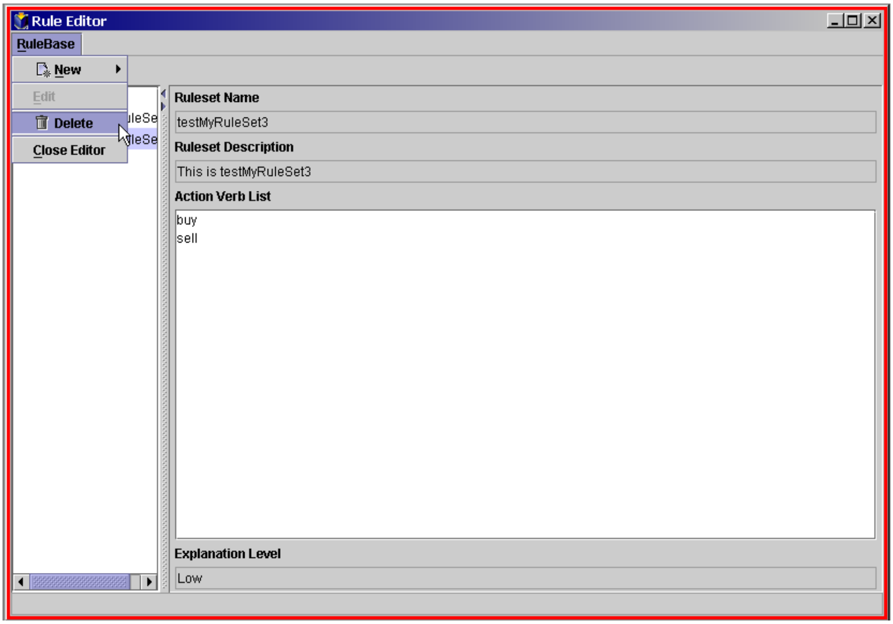
Figure 10-51. Deleting a Rule - Rule Editor Display - Rule Selected for Deletion



Ruleset
testMyRuleSet1
Rule Id
2
Priority
Medium
Description
Rule 2
Subject
Customer
Object
Equity
Verb
Recommend sell
Verb Negated?
No
Natural Language Explanation
Sell for Beta too high.

3 Select the menu item **RuleBase > Delete.**

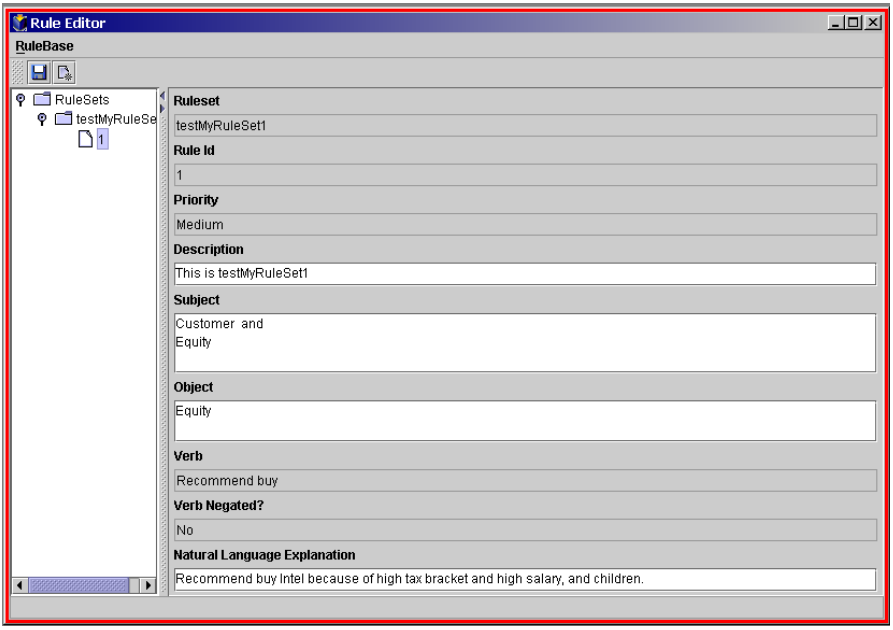
Figure 10-52. Deleting a Rule - Rule Editor Display - Menu Item “Delete” Selected



Ruleset Name
testMyRuleSet3
Ruleset Description
This is testMyRuleSet3
Action Verb List
buy
sell
Explanation Level
Low

4 The Rule Editor display updates showing the Rule is no longer present. The Rule is deleted.

Figure 10-53. Deleting a Rule - Rule Editor Display - Rule Deleted



Ruleset
testMyRuleSet1
Rule Id
1
Priority
Medium
Description
This is testMyRuleSet1
Subject
Customer and
Equity
Object
Equity
Verb
Recommend buy
Verb Negated?
No
Natural Language Explanation
Recommend buy Intel because of high tax bracket and high salary, and children.

Negating Rules

Suppose you have a Rule such as:

```
If Customer.Age > 55 then Recommend Buy Equity.Type = Bonds
```

But you may have extremely adventuresome, wealthy, and speculative seniors for whom this “safe” recommendation is not acceptable. In this case and only for those Customers, you want to negate the Bond Recommendation. Your Rule might look something like this:

```
If Customer.Investment_Goals = Growth and
   Customer.Risk_Preference = High and
   Customer.Like_Investor_Pool = Venturesome and
   Customer.Annual_Income>200000 then
   Do Not Recommend Buy
    Equity.Type = Bonds
```

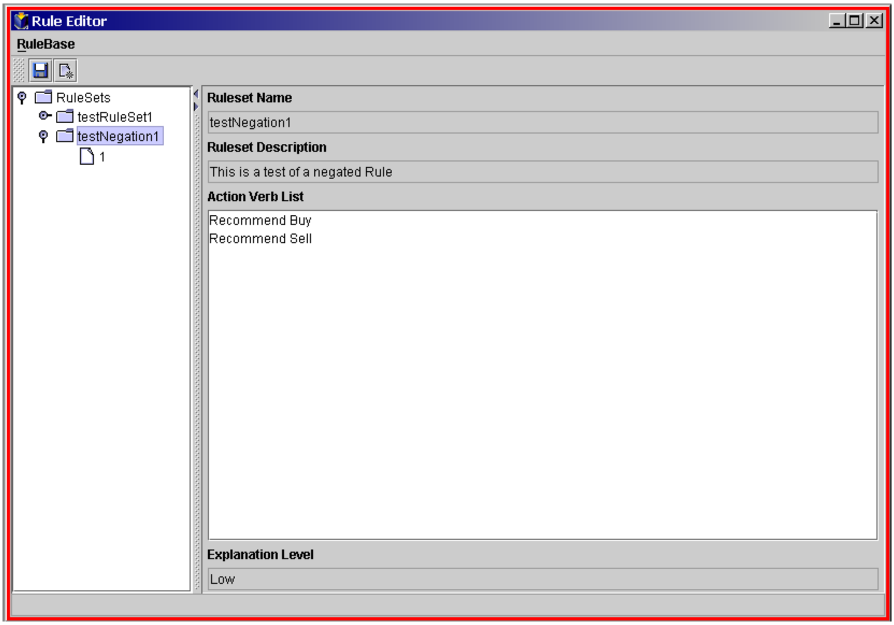
To accomplish this you do not have to add a new Verb to a Ruleset. Instead, you create a Rule specifying Verb Negation on “Recommend Buy”.

When two or more Rules within a Ruleset activate and all share the same Verb (in this case, “Recommend Buy”) and priority, a Negated Verb “trumps” all other equal-priority active Rules for all instances satisfying the negated Rule’s Subject Expression.

Creating a Negated Rule

1 To negate a Rule, open the Rule Editor, if it is not currently open. Within the left-hand Rulesets panel, select the Ruleset Element that contains the Rule or Rules you want to negate.

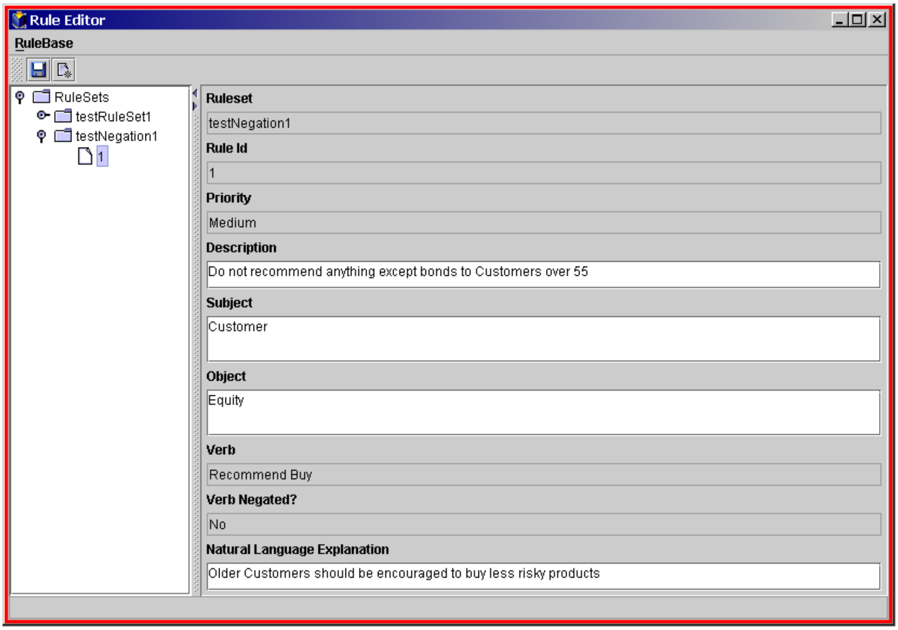
Figure 10-54. Negating a Rule - Rule Editor Display - Ruleset Selected



Ruleset Name
testNegation1
Ruleset Description
This is a test of a negated Rule
Action Verb List
Recommend Buy
Recommend Sell
Explanation Level
Low

2 Select the Rule from within the left-hand Ruleset panel. Note the name of the Verb you want to negate within the right-hand Rule Status display.

Figure 10-55. Negating a Rule - Rule Editor Display - Rule Selected



Ruleset
testNegation1

Rule Id
1

Priority
Medium

Description
Do not recommend anything except bonds to Customers over 55

Subject
Customer

Object
Equity

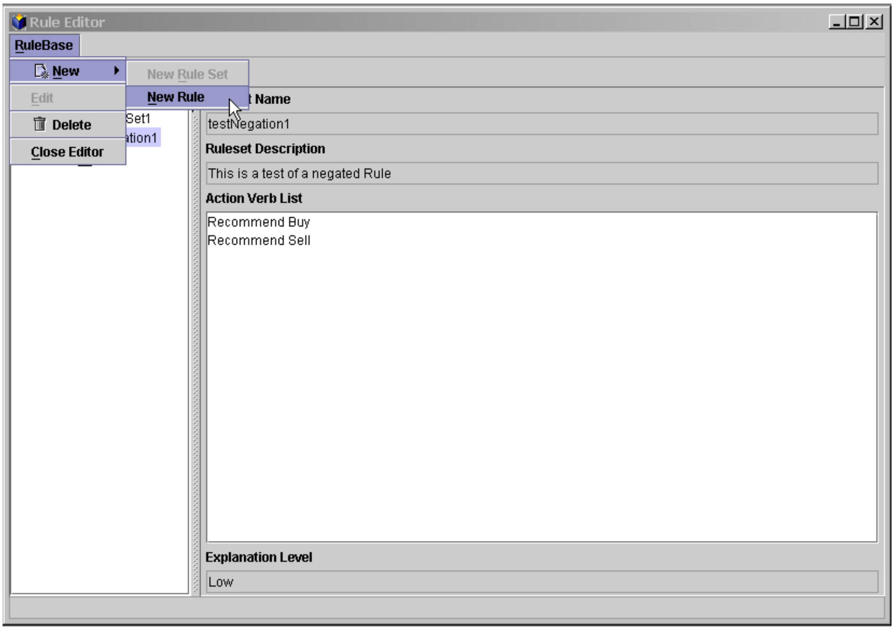
Verb
Recommend Buy

Verb Negated?
No

Natural Language Explanation
Older Customers should be encouraged to buy less risky products

3 Highlight the Ruleset Element and select the menu item **RuleBase > New > New Rule**.

Figure 10-56. Negating a Rule - Rule Editor Display - Menu Item "New Rule" Selected



t Name
testNegation1

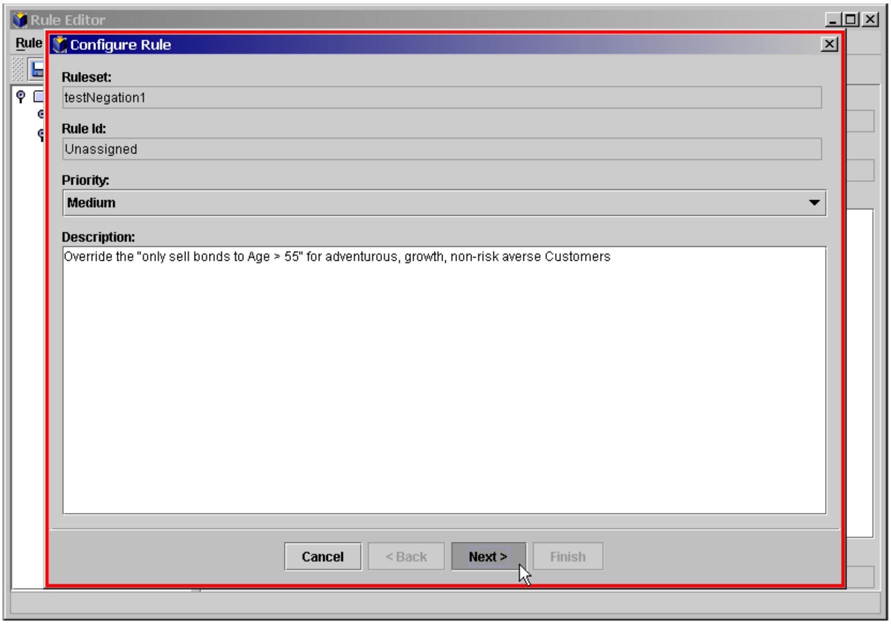
Ruleset Description
This is a test of a negated Rule

Action Verb List
Recommend Buy
Recommend Sell

Explanation Level
Low

4 The Configure Rule Wizard displays. Type or paste the Negation Rule description here. Click the Next> button to continue.

Figure 10-57. Negating a Rule - Configure Rule - Entering Description



Ruleset: testNegation1

Rule Id: Unassigned

Priority: Medium

Description: Override the "only sell bonds to Age > 55" for adventurous, growth, non-risk averse Customers

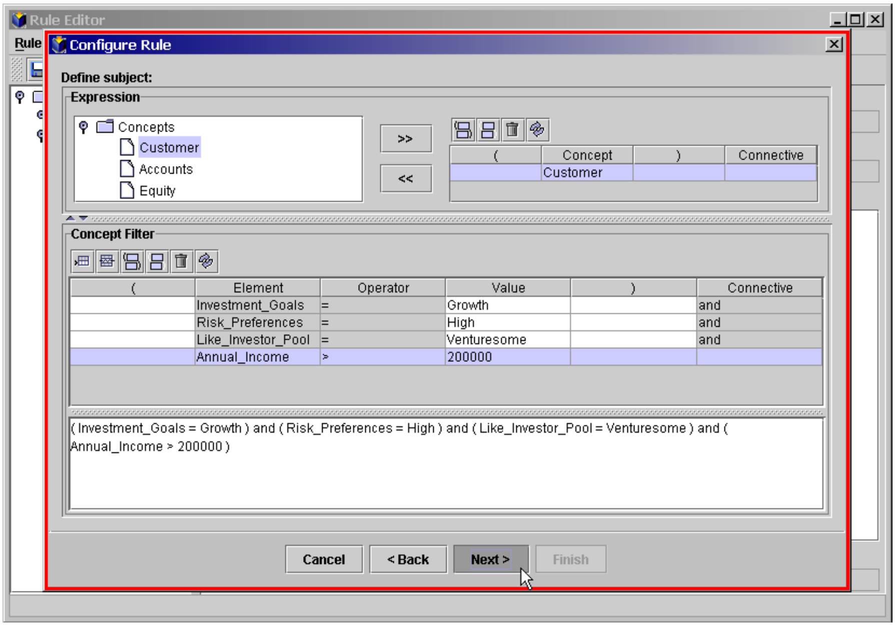
5 The Configure Rule Wizard now displays the Subject Expression screen. Using the same techniques outlined in the section *Creating a Rule on page 360*, create a Subject Expression that contains these Concept Filters:

```
Customer.Investment_Goals = Growth
Customer.Risk_Preference = High
Customer.Like_Investor_Pool = Venturesome
Customer.Annual_Income > 200000
```

and link them all using the logical AND connective.

Click the Next> button to continue.

Figure 10-58. Negating a Rule - Configure Rule - Subject Expression



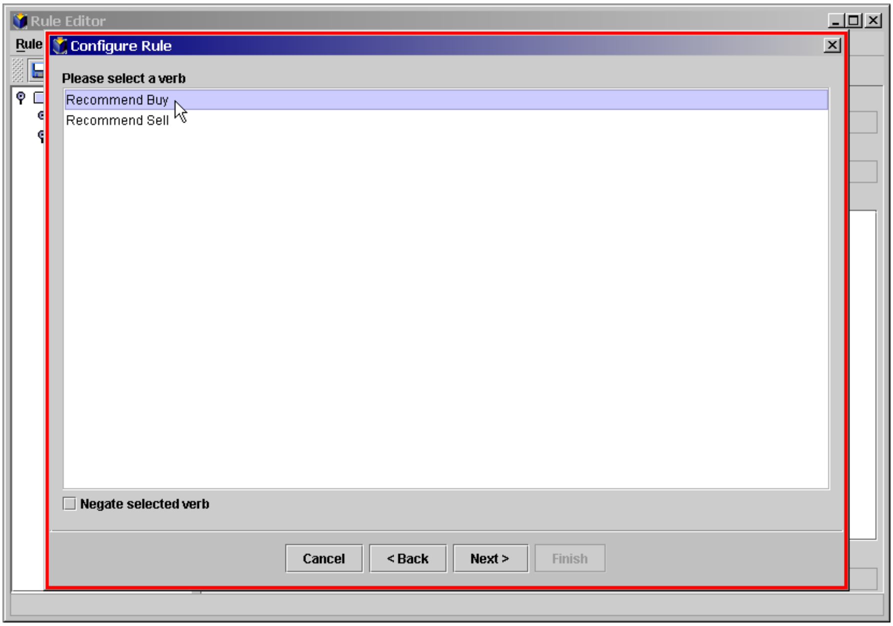
(	Concept	)	Connective
	Customer		

(	Element	Operator	Value	)	Connective
	Investment_Goals	=	Growth		and
	Risk_Preferences	=	High		and
	Like_Investor_Pool	=	Venturesome		and
	Annual_Income	>	200000		

(Investment_Goals = Growth) and (Risk_Preferences = High) and (Like_Investor_Pool = Venturesome) and (Annual_Income > 200000)

6 The Configure Rule Wizard displays the list of available Verbs. Click to select the Verb you want to negate.

Figure 10-59. Negating a Rule - Configure Rule - Verb Selected



Please select a verb

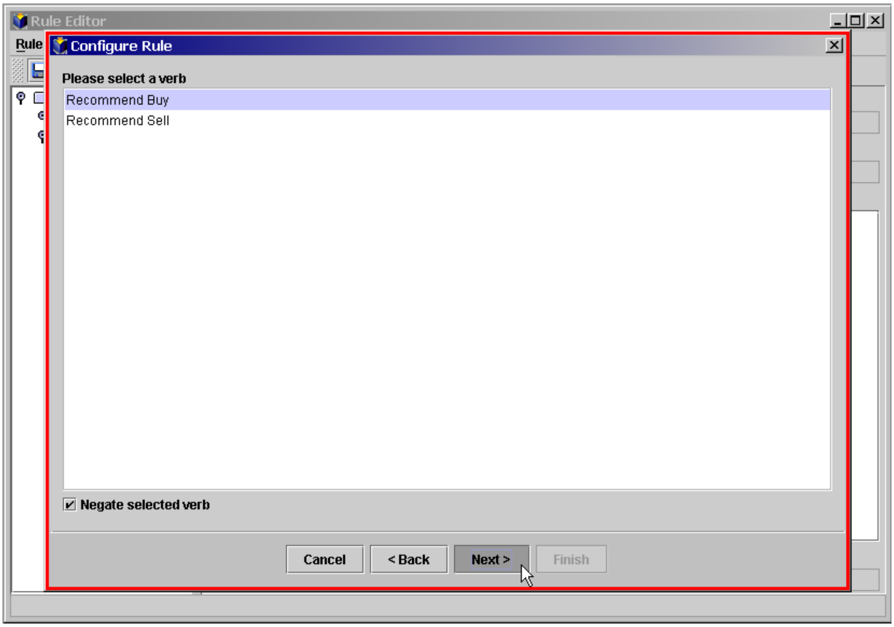
Recommend Buy

Recommend Sell

□ Negate selected verb

7 Click the Negate selected verb checkbox. The Verb is now negated. Click the Next> button to continue.

Figure 10-60. Negating a Rule - Configure Rule - Selected Verb Negated



Please select a verb

Recommend Buy

Recommend Sell

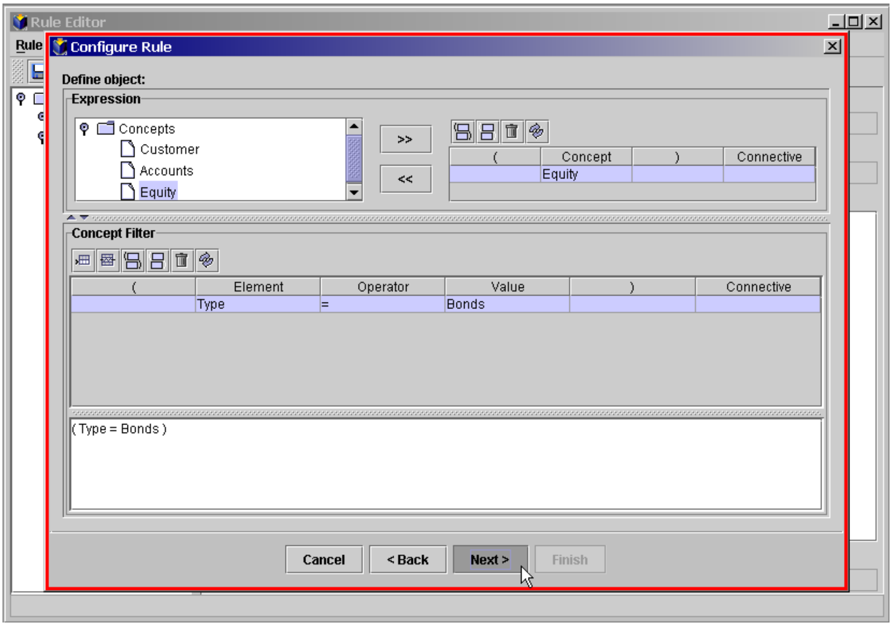
☑ Negate selected verb

8 The Configure Rule Wizard now displays the Object Expression screen. Using the same techniques outlined in the section *Creating a Rule on page 360*, create an Object Expression that contains this Concept Filters:

```
Equity.Type = Bonds
```

Click the Next> button to continue.

Figure 10-61. Negating a Rule - Configure Rule - Object Expression



Define object:

Expression

(	Concept	)	Connective
	Equity		

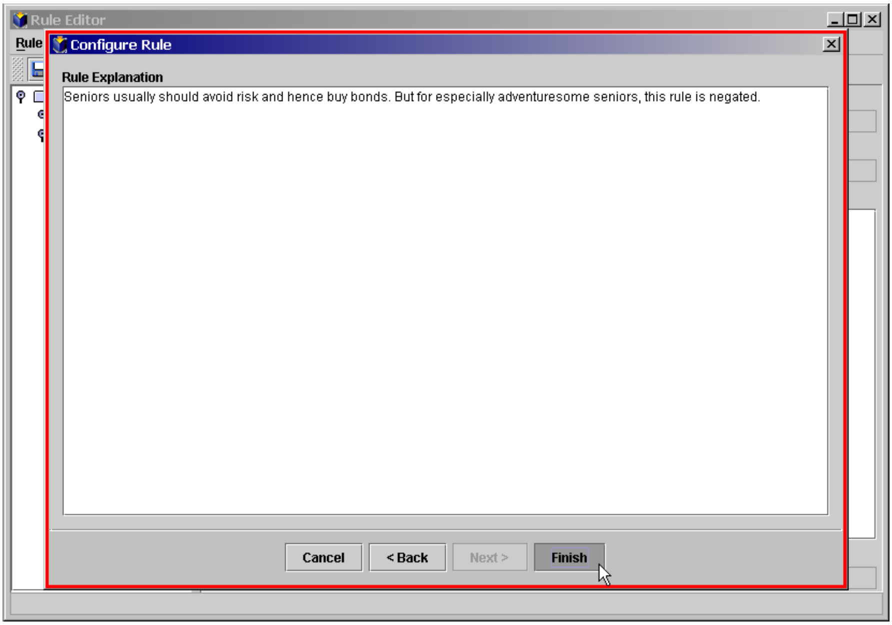
Concept Filter

(	Element	Operator	Value	)	Connective
	Type	=	Bonds		

(Type = Bonds)

9 Type or paste text into the Rule Explanation text box. To complete creating the Negation Rule, click the Finish button.

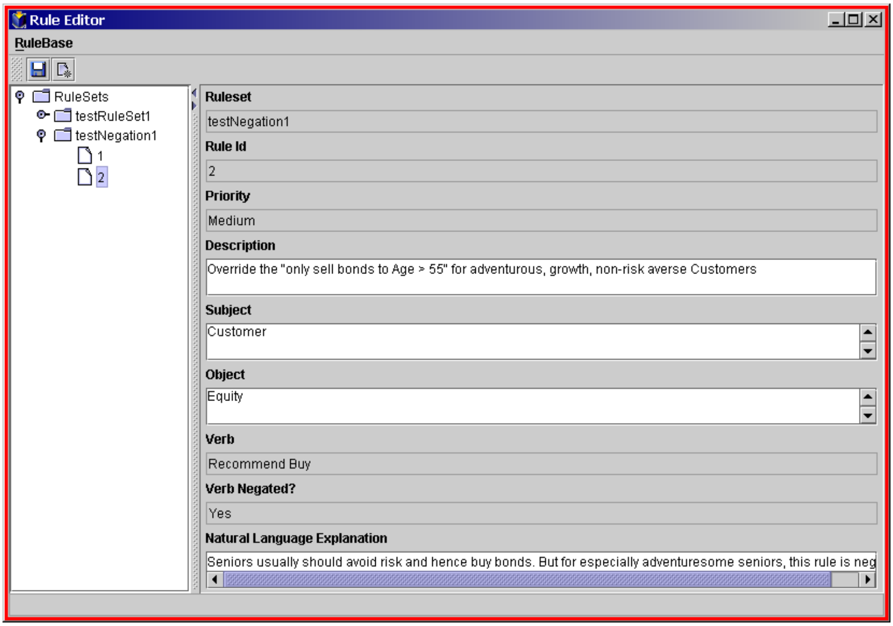
Figure 10-62. Negating a Rule - Configure Rule - Rule Explanation



Seniors usually should avoid risk and hence buy bonds. But for especially adventuresome seniors, this rule is negated.

10 The Rule Editor displays the new Rule. You have finished negating a Rule.

Figure 10-63. Negating a Rule - Rule Editor - New Negation Rule Displays



Ruleset
testNegation1

Rule Id
2

Priority
Medium

Description
Override the "only sell bonds to Age > 55" for adventurous, growth, non-risk averse Customers

Subject
Customer

Object
Equity

Verb
Recommend Buy

Verb Negated?
Yes

Natural Language Explanation
Seniors usually should avoid risk and hence buy bonds. But for especially adventuresome seniors, this rule is neg

Using Complex Rule Features

Rulesets and Rules can become very complex. Prioritization, Grouping, and Precedence are advanced features to help you manage this complexity.

Prioritizing Rules

Knowledge Broker enables you to prioritize Rules, determining which Rule will take precedence in the event of two contradictory Rules saved within a Ruleset. A Rule with a higher priority will execute in preference to a rule with a lower priority. The priority order from highest to lowest is described in Table 10-7.

Table 10-7. Rule Priorities

Priority	Numeric Value
High	3
Medium	2
Low	1

For example, suppose the following two rules fire:

1. `Recommend buy technology stocks`
2. `Do not recommend Microsoft stocks`

In this example, the negated rule (Rule 2) outweighs the positive rule (Rule 1) because a negated rule will always trump a positive rule of the same priority. In this case all the technology stocks except for Microsoft will be recommended. If you were to assign a higher priority to Rule 1, all Technology stocks including Microsoft would be recommended.

Using Grouping and Precedence

Knowledge Broker's Rule Editor enables you to join Expressions using both the logical `AND` and the logical `OR` operators. The AND operator is easily used and understood whereas the `OR` operator introduces ambiguity that is minimized using the Grouping parenthesis operators `(` and `)` to constrain Precedence.

Without using the OR operator, to encode a hunch such as "If Customers are rich *or* adventurous then recommend risky stocks" would require two separate rules:

- `If Customers are rich then recommend risky stocks.`
- `If Customers are adventurous then recommend risky stocks.`

The resulting set of rules could become large and repetitive, making it difficult to read and maintain.

The `AND` operator is true (returns a recommendation) if all the Comparisons joined by `AND` are themselves true (contain one or more instances).

Knowledge Broker supports the use of the logical `OR` operator to combine expressions in the Subject or Object. This allows you to create more complex Rules that more naturally represent real-world concepts. In addition, combining many expressions using `ORs` potentially reduces the number of Rules required to model a problem. This is because a single Rule that combines expressions using `ANDs` and `ORs` can replace many near-identical rules formed using only `ANDs`.

For the earlier example, therefore, a single rule using the new `OR` functionality suffices:

- "If Customers are rich *or* Customers are adventurous then recommend risky stocks".

The `OR` operator is true (returns a recommendation) if at least one of the Comparisons joined by `OR` are themselves true (contain one or more instances).

However, this new functionality introduces extra complexity into the Rule processing process. Using the rules of logic, any number of expressions joined by an `AND` operator can be calculated in any order. By contrast, the result of combining expressions using `ORs`, or a combination of `ORs` and `ANDs`, is not as straightforward.

Precedence

On a basic level, Knowledge Broker enforces **left-to-right Precedence**. That is, when processing a Rule, it reads from left to right as with Standard English, processing the logical connectives one-by-one. If all connectives were logical ANDs, this would present no issues.

To illustrate, consider this expression combination:

```
Equity1 AND Equity2 AND Equity3 AND Equity4
```

Knowledge Broker would process this as

1 Calculate the result of `Equity1 AND Equity2`

2 Make this result equal `Equity1-2`.

3 Calculate the result of `Equity1-2 AND Equity3`

4 Make this result equal `Equity1-2-3`.

5 Calculate the result of `Equity1-2-3 AND Equity4`.

6 Make this result equal `Equity1-2-3`-4.

Due to the rules of logic (AND is commutative, a + b always equals b + a), this expression could be rewritten as the following expression combination and the result, `Equity2-1-4-3`, would always equal `Equity1-2-3`-4:

```
Equity2 AND Equity1 AND Equity4 AND Equity3
```

By contrast, logical OR is not commutative, and so the processing sequence will affect the value of the final result.

Examine the following expression combination:

```
Equity1 OR Equity2 AND Equity3 OR Equity4
```

Knowledge Broker would process this as

1 Calculate the result of `Equity1 OR Equity2`

2 Make this result equal `Equity1-2`.

3 Calculate the result of `Equity1or2 AND Equity3`

4 Make this result equal `Equity1or2and3`.

5 Calculate the result of `Equity1or2and3 AND Equity4`.

6 Make this result equal `Equity1or2and3`or4.

If the expression was sequenced differently, then the result, `Equity2or3and4or1`, would not necessarily equal the previous result, `Equity1or2and3`or4:

```
Equity2 OR Equity3 AND Equity4 OR Equity1
```

So specifying the logical sequence when using Precedence and ORs is critical. You can, however, guide Precedence by using Knowledge Broker's Parentheses operators (and) to specify Grouping.

Grouping or Nesting

Parentheses are a standard way to define which logical subunits of an expression should be calculated first. When Knowledge Broker reads a Rule from left-to-right, if Parentheses operators are present then it first evaluates the deepest level of Parentheses, also using left-to-right Precedence. It goes on evaluating (or "expanding") each nested level of Parentheses until there are no more remaining. Then it processes the fully expanded Rule from left-to-right.

For an example of nested Parentheses expansion, consider this expression that lacks operators but displays some complex nesting:

```
( A ( ( ( B ) C ( D ) ( E ) F ) G ) )
  1       4   3   4     4   3   2
```

The numbers underneath expression components indicate their nesting "level". Using Knowledge Broker's left-to-right Precedence and Grouping, this expression expands as follows:

```
1  ( A ( ( ( B ) C ( D ) ( E ) F ) G ) )
     1       4   3   4     4   3   2

2  ( A ( ( B C D E F ) G ) )
     1         3       2

3  ( A ( B C D E F G ) )
     1        2

4  ( A B C D E F G )
            1
```

So expansion really means progressively removing the Parentheses from the "inside-out".

Using Parentheses, the earlier expression combination:

```
Equity1 OR Equity2 AND Equity3 OR Equity4
```

using left-to-right Precedence and Parentheses becomes:

```
((Equity1 OR Equity2) AND Equity3) OR Equity4
```

This is the default calculation sequence. But you could also define your preferred calculation sequence using parentheses. Such sequences could be represented as:

```
(Equity1 OR Equity2) AND (Equity3 OR Equity4)

(Equity1 OR (Equity2 AND Equity3)) OR Equity4
```

The earlier expression combination can be re-written using Parentheses as both of these expressions:

```
1  (Equity1 OR Equity2) AND (Equity3 OR Equity4)

2  (Equity1 OR (Equity2 AND Equity3)) OR Equity4
```

Expressions contained within parentheses are calculated first, and this result is then used in the calculation. For example, in expression combination 1 above, Knowledge Broker's calculation reasoning process follows this sequence:

1. Calculate the result of `Equity1 OR Equity2`.
2. Make this result equal `Equity1or2`.
3. Calculate the result of `Equity3 OR Equity4`.
4. Make this result equal `Equity3or4`.
5. Calculate the result of `Equity1or2 AND Equity3or4`.
6. Make this result equal `Equity1or2andEquity3or4`.

The second expression combination above uses nested levels of Parentheses, and therefore Knowledge Broker's processing follows a different sequence to arrive at a different result:

1. Calculate the result of `Equity2 AND Equity3`.
2. Make this result equal `Equity2and3`.
3. Calculate the result of `Equity1 OR Equity2and3`.
4. Make this result equal `Equity1orEquity2and3`.
5. Calculate the result of `Equity1orEquity2and3 OR Equity4`
6. Make this result equal `Equity1orEquity2and3orEquity4`.

Judicious use of the parentheses operators allows you to create complex rules. For example, suppose you want to recommend Volatile technology stocks to specific customers. Specifically, if your customers combine a high liquidity with a preference for high risk, then recommend these stocks to them. Additionally, recommend these stocks to customers with a "high value" using certain company-standard metrics.

Use a Filter in the Ontology Editor to specify High, Moderate, and Low Liquidity Customer DBCs according to the value of their Liquidity Preference Property Value. Further specify High, Moderate, and Low Risk Customer DBCs according to the value of their Risk Preference Property Value. Additionally, specify a High Value Business Concept DBC as a Customer with `Like Investor Pool = Active AND Investment Goals = Growth AND Annual Income >= $100,000`.

Finally, categorize Equities as Volatile, Moderate, or Low DBCs according to their Beta values.

Using `ORs` and Parentheses, you can express your goal as follows:

```
If ((High Liquidity AND High Risk) OR High Value) then recommend buy
Volatile Technology
```

Note that the above rule will produce a different recommendation not in accordance with your wishes if composed as follows:

```
If (High Liquidity AND (High Risk OR High Value)) then
recommend buy Volatile Technology
```

Chapter

11

Discovering Knowledge

This chapter provides step-by-step instructions for using the Knowledge Discovery Center to create Data Sets and train Models. Models can be embedded within the Ontology as Calculated Elements.

Introducing Knowledge Discovery

Data Sets “sample” the instance data from a Business Concept to create a subset of data that you use to train or test Models.

During training, Knowledge Broker optimizes a particular Model to “fit” the submitted Data Set. When trained, Models can analyze new data based on its similarity to patterns discovered during training. Models can also predict new data based on probabilistic reasoning.

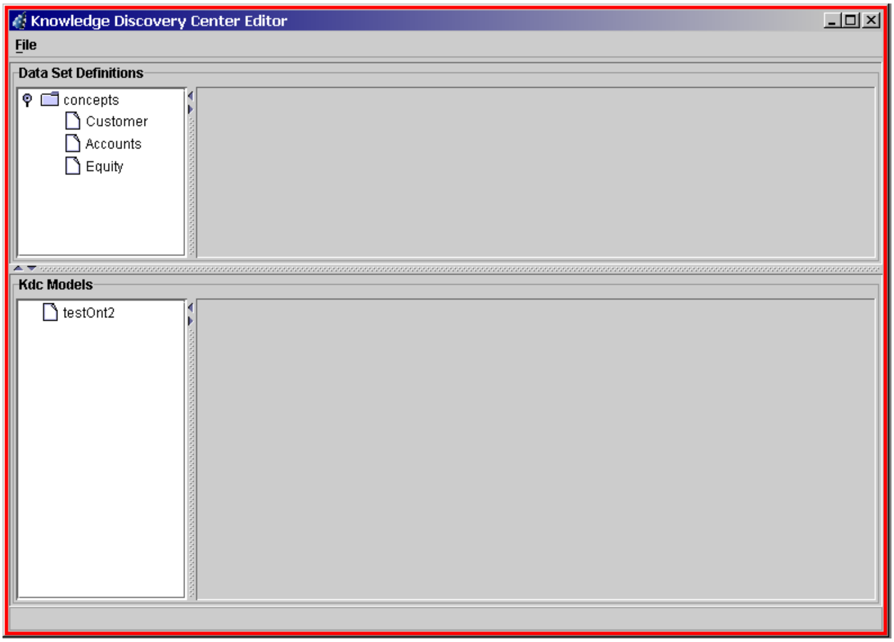
Using Data Sets

It is unrealistic and unproductive to train a Model using all available data. Instead, you sample a subset of this data to create a representative Data Set.

Creating Data Sets

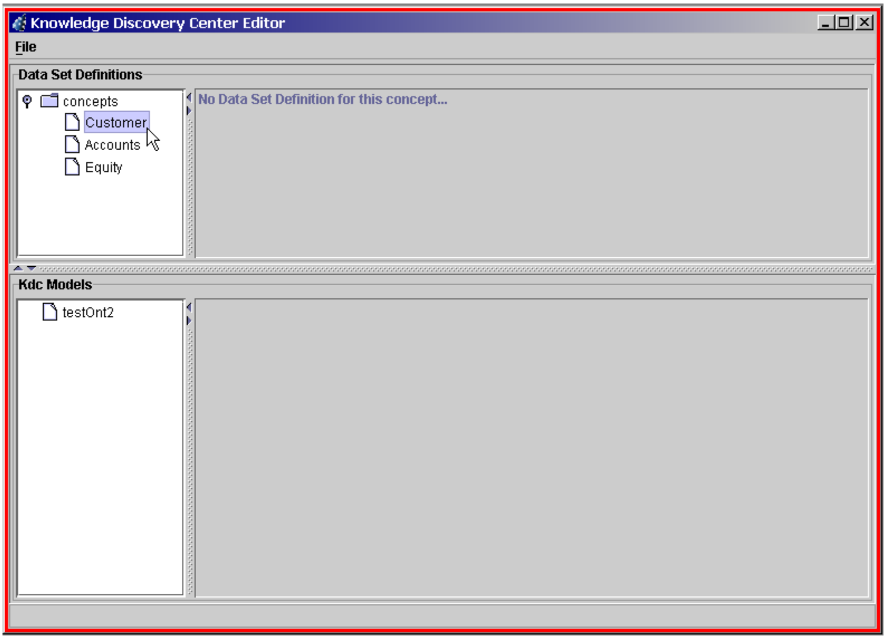
1 Open the Knowledge Discovery Center, if it is not currently open.

Figure 11-1. Knowledge Discovery Center - Initial Display



2 The available Concepts display in the Data Set Definitions upper left-hand panel. Click to select the Concept from which you will create a Data Set.

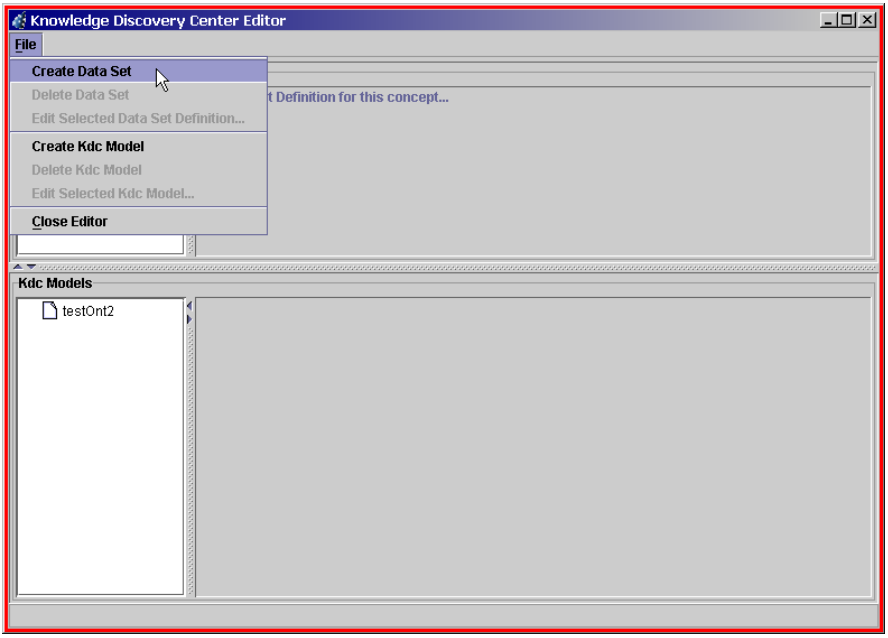
Figure 11-2. Creating a Data Set - Source Concept Selected



No Data Set Definition for this concept...

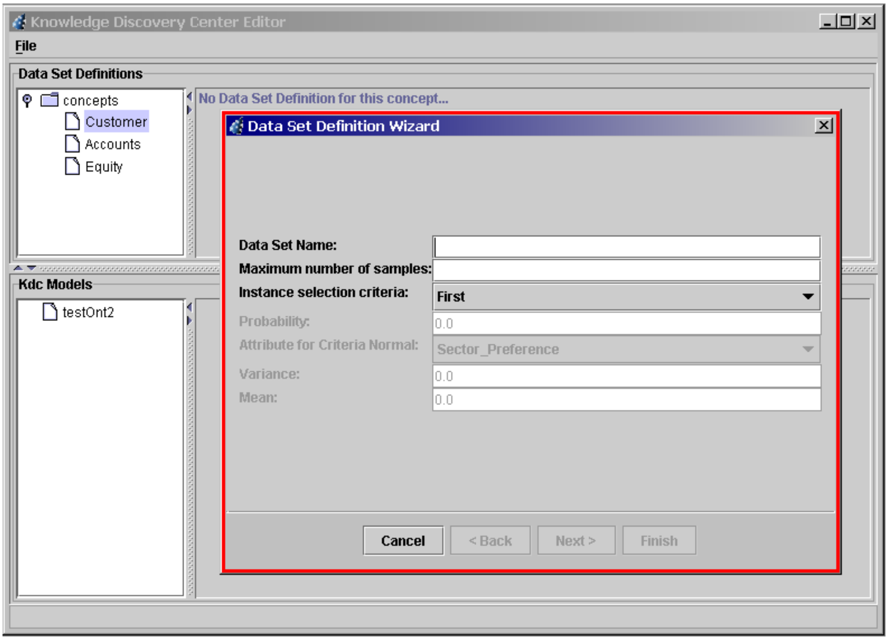
3 Select the menu item **File > Create Data Set.**

Figure 11-3. Creating a Data Set - Menu Item "Create" Selected


Knowledge Discovery Center Editor
File
Create Data Set
Delete Data Set
Edit Selected Data Set Definition...
Create Kdc Model
Delete Kdc Model
Edit Selected Kdc Model...
Close Editor
t Definition for this concept...
Kdc Models
testOnt2

4 The Data Set Definition Wizard displays.

Figure 11-4. Creating a Data Set - Data Set Definition Wizard Displays



No Data Set Definition for this concept...

Data Set Name:

Maximum number of samples:

Instance selection criteria: First

Probability: 0.0

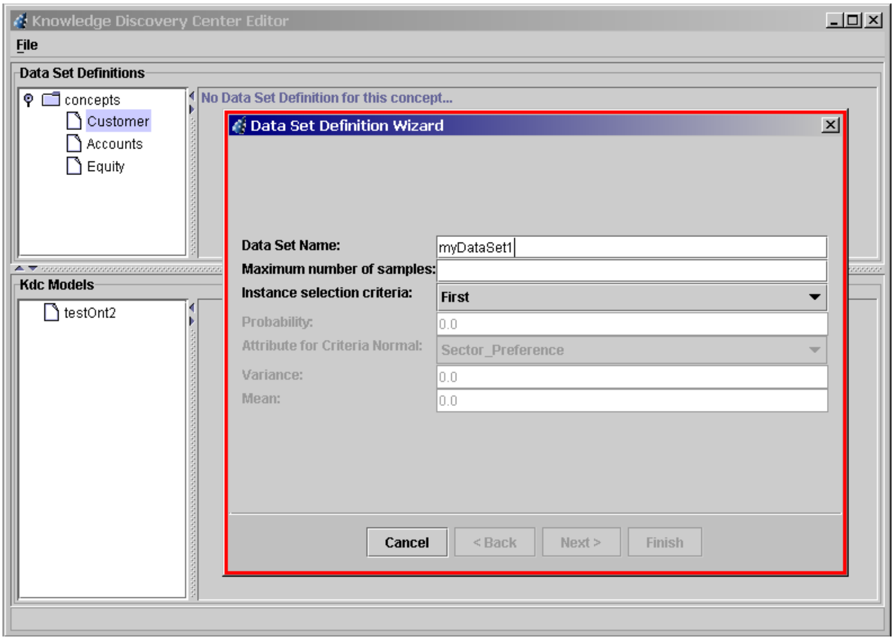
Attribute for Criteria Normal: Sector_Preference

Variance: 0.0

Mean: 0.0

5 You must name a Data Set by typing or pasting a name into the Data Set Name field.

Figure 11-5. Creating a Data Set - Naming the Data Set



No Data Set Definition for this concept...

Data Set Name: myDataSet1

Maximum number of samples:

Instance selection criteria: First

Probability: 0.0

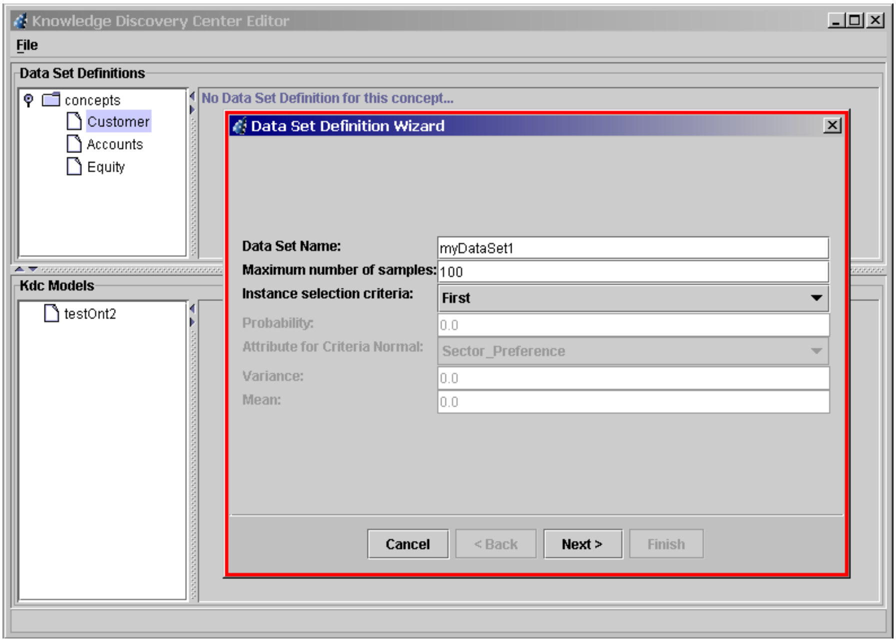
Attribute for Criteria Normal: Sector_Preference

Variance: 0.0

Mean: 0.0

6 You must set a limit to the size of the Data Set by typing or pasting a number into the Maximum number of samples field.

Figure 11-6. Creating a Data Set - Selecting the Sample Size



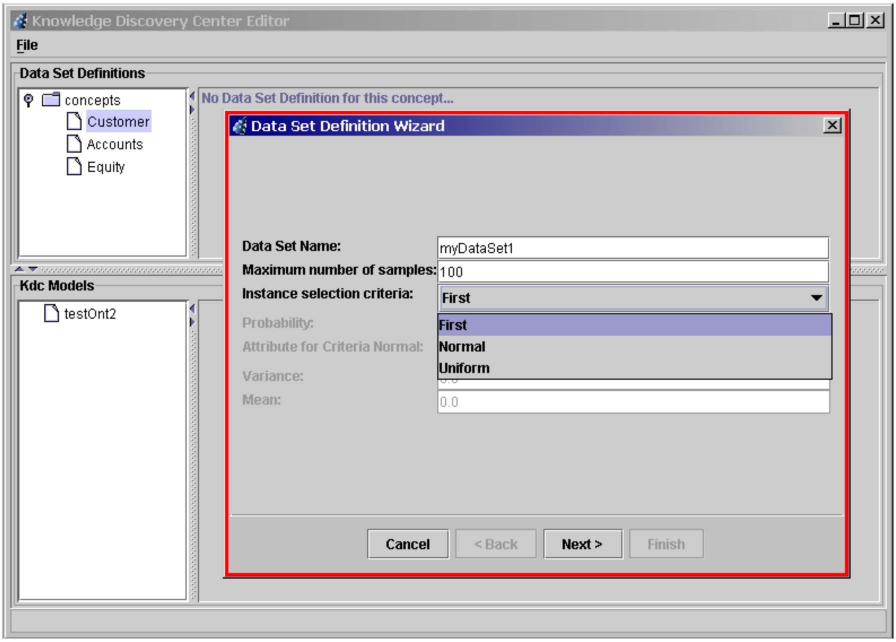
Data Set Name: myDataSet1
Maximum number of samples: 100
Instance selection criteria: First
Probability: 0.0
Attribute for Criteria Normal: Sector_Preference
Variance: 0.0
Mean: 0.0

Knowledge Discovery Center Editor
File
Data Set Definitions
concepts
Customer
Accounts
Equity
Kdc Models
testOnt2
No Data Set Definition for this concept...
Data Set Definition Wizard
Cancel
< Back
Next >
Finish

To create a Data Set that includes all of the instance data of its parent Concept, type or paste a number larger than the expected number of instances in the Concept.

7 You must now select a sampling method appropriate for your Data Set purposes. Click the Instance selection criteria drop-down menu. A list of the available sampling methods displays.

Figure 11-7. Creating a Data Set - Displaying the Sampling Methods



No Data Set Definition for this concept...

Data Set Name: myDataSet1

Maximum number of samples: 100

Instance selection criteria: First

Probability:

Attribute for Criteria Normal:

Variance: 0.0

Mean: 0.0

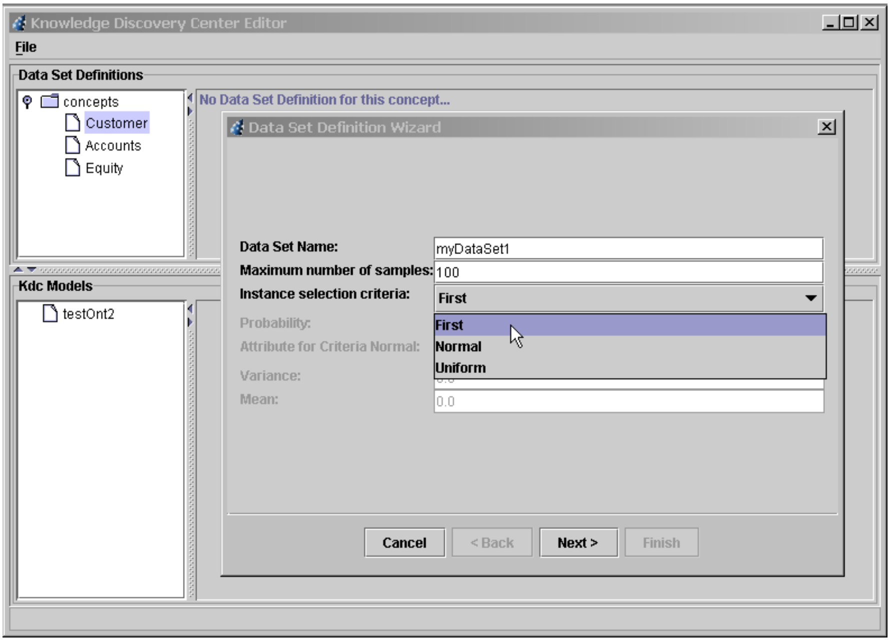
The three available sampling methods are:

Table 11-1. Data Set Sampling Methods

Name	Description
First	Sequentially samples the first *n* instances, where *n* equals the defined maximum number of samples.
Normal	Probabilistically samples up to *n* instances, assuming a normal distribution (also called a Gaussian distribution).
Uniform	Probabilistically samples up to *n* instances, assuming a uniform distribution (also called a rectangular distribution).

8 Select the desired sampling method. Click to confirm your choice.

Figure 11-8. Creating a Data Set - Sampling Method Selected



No Data Set Definition for this concept...

Data Set Name: myDataSet1

Maximum number of samples: 100

Instance selection criteria: First

Probability:

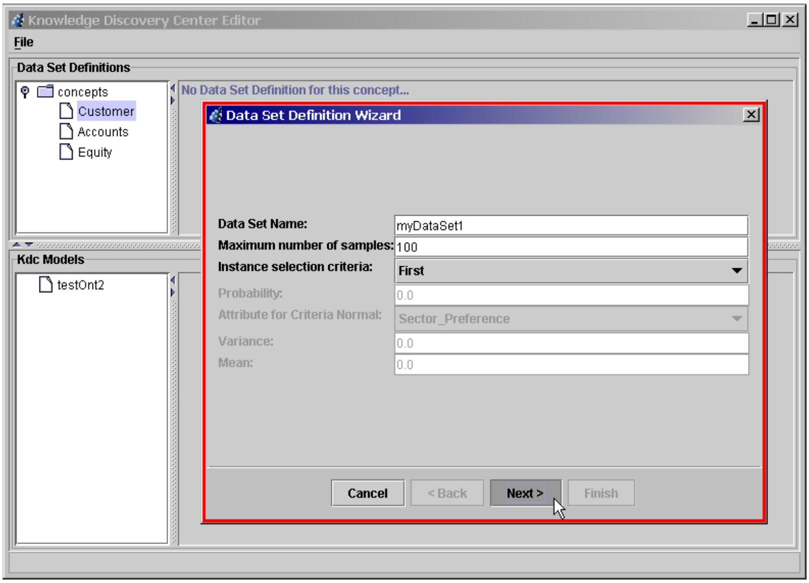
Attribute for Criteria Normal:

Variance:

Mean: 0.0

9 The Data Set Definition Wizard updates to reflect your selection of sampling method. Click the Next> button to continue.

Figure 11-9. Creating a Data Set - Selected Sampling Method Displays



Knowledge Discovery Center Editor
File
Data Set Definitions
concepts
Customer
Accounts
Equity
No Data Set Definition for this concept...
Kdc Models
testOnt2
Data Set Definition Wizard
Cancel
< Back
Next >
Finish

Data Set Name: myDataSet1
Maximum number of samples: 100
Instance selection criteria: First
Probability: 0.0
Attribute for Criteria Normal: Sector_Preference
Variance: 0.0
Mean: 0.0

Selecting the advanced probabilistic sampling methods, requires some additional statistical parameters:

Table 11-2. Data Set Statistical Parameters

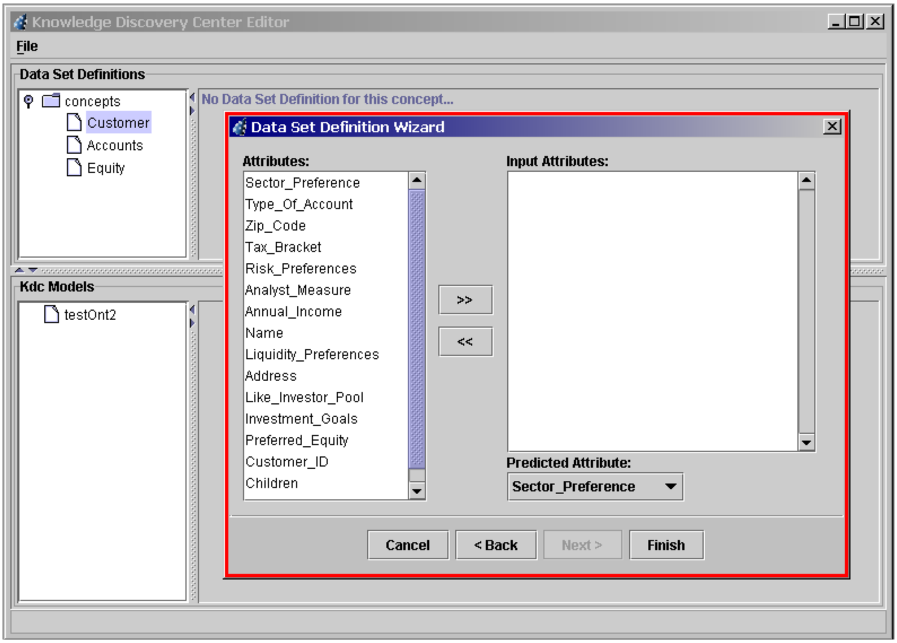
Name	Description	Sampling Method
Probability	Knowledge Broker applies a uniform probability to random data until it has selected the required number of instances.	Uniform
Attribute for Criteria Normal	This is the Normally distributed Attribute.	Normal
Variance	This is the square of the sample standard deviation.	Normal
Mean	This is the sample mean.	Normal

If you suspect your data is distributed within a classic “bell curve”, then you should use Normal Sampling as the Instance selection criterion, along with appropriate Variance and Mean values.

If, however, you suspect there is no bell curve, then you should select Uniform Probability Sampling as the Instance selection criterion.

10 The Data Set Wizard displays the list of available Attributes (that is, Concept Elements). You use this screen to select your input and predicted attributes.

Figure 11-10. Creating a Data Set - Available Attributes



Knowledge Discovery Center Editor

File

Data Set Definitions

concepts
Customer
Accounts
Equity

No Data Set Definition for this concept...

Kdc Models

testOnt2

Data Set Definition Wizard

Attributes:

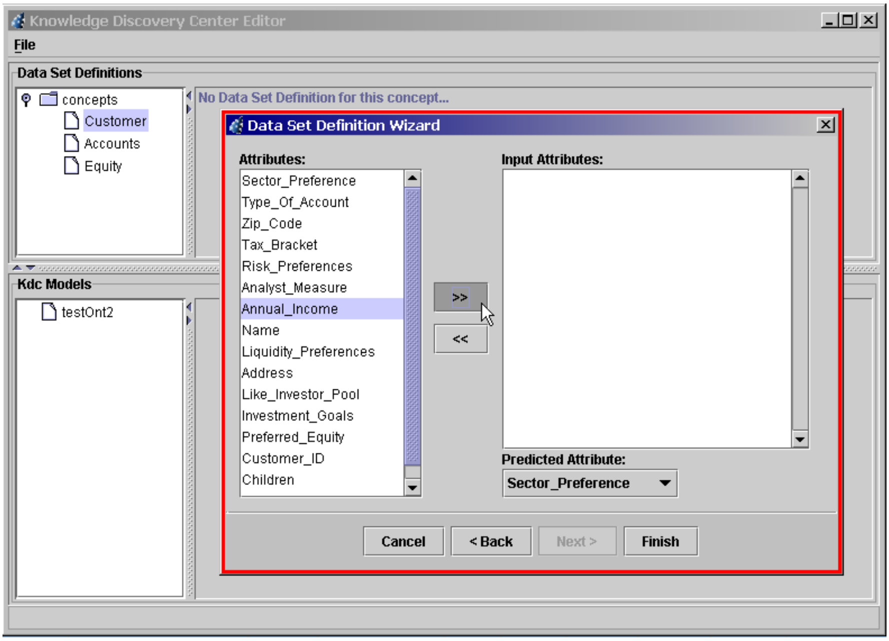
Sector_Preference
Type_Of_Account
Zip_Code
Tax_Bracket
Risk_Preferences
Analyst_Measure
Annual_Income
Name
Liquidity_Preferences
Address
Like_Investor_Pool
Investment_Goals
Preferred_Equity
Customer_ID
Children

Input Attributes:

Predicted Attribute: Sector_Preference

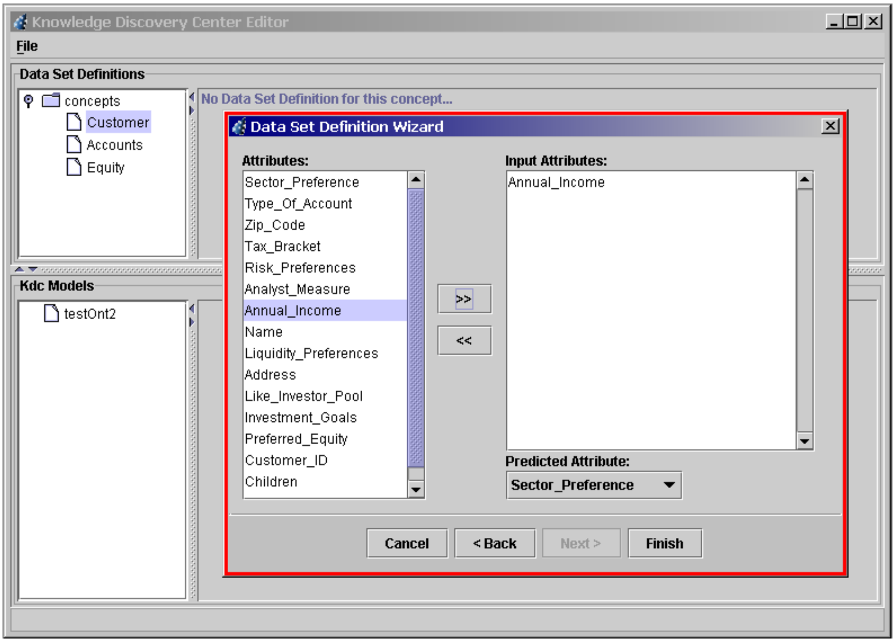
11 To select the Input Attributes, highlight an Attribute from within the Attributes list on the left-hand panel and click on the >> button.

Figure 11-11. Creating a Data Set - Input Attribute Selected



12 The selected Attribute moves into the right-hand Input Attributes panel.

Figure 11-12. Creating a Data Set - Input Attribute Confirmed


Knowledge Discovery Center Editor
File
Data Set Definitions
concepts
Customer
Accounts
Equity
No Data Set Definition for this concept...
Kdc Models
testOnt2
Data Set Definition Wizard
>>
<<
Cancel
< Back
Next >
Finish

Attributes:
Sector_Preference
Type_Of_Account
Zip_Code
Tax_Bracket
Risk_Preferences
Analyst_Measure
Annual_Income
Name
Liquidity_Preferences
Address
Like_Investor_Pool
Investment_Goals
Preferred_Equity
Customer_ID
Children

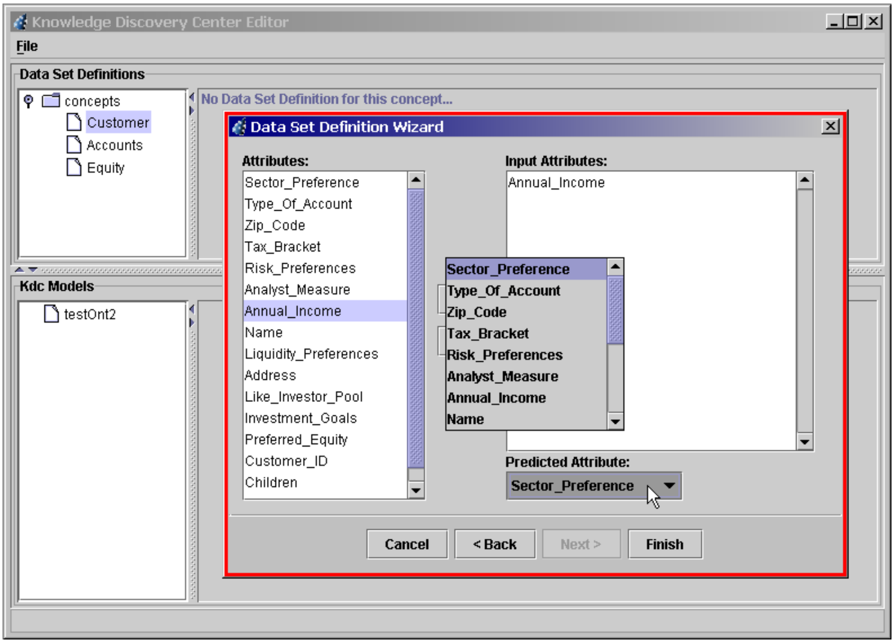
Input Attributes:
Annual_Income

Predicted Attribute:
Sector_Preference

13 To select multiple Input Attributes, repeat step 11 as required.

14 To select the Predicted Attribute, click the Predicted Attribute drop-down menu. A list of available Attributes displays.

Figure 11-13. Creating a Data Set - Predicted Attribute Drop-Down Menu Appears



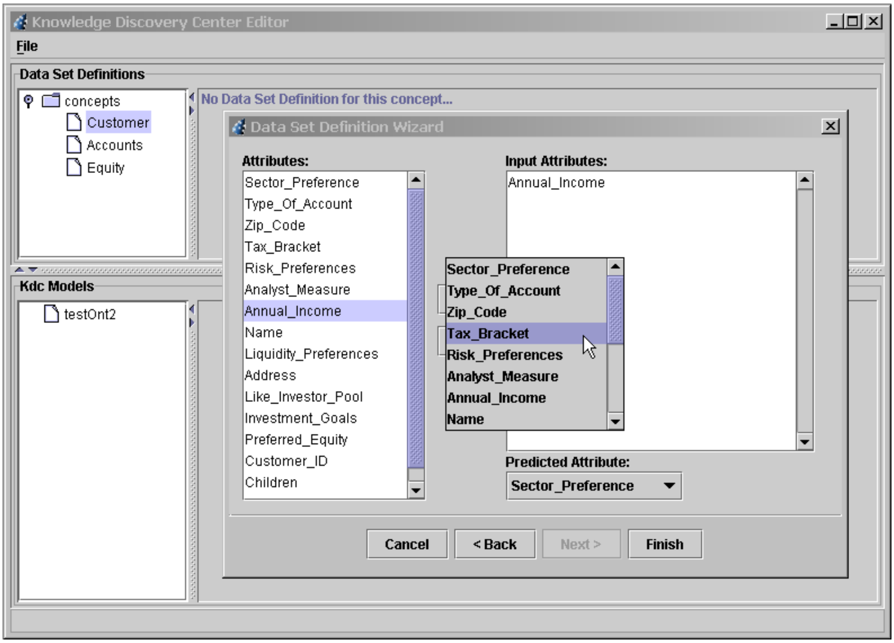
No Data Set Definition for this concept...

Input Attributes: Annual_Income

Predicted Attribute: Sector_Preference

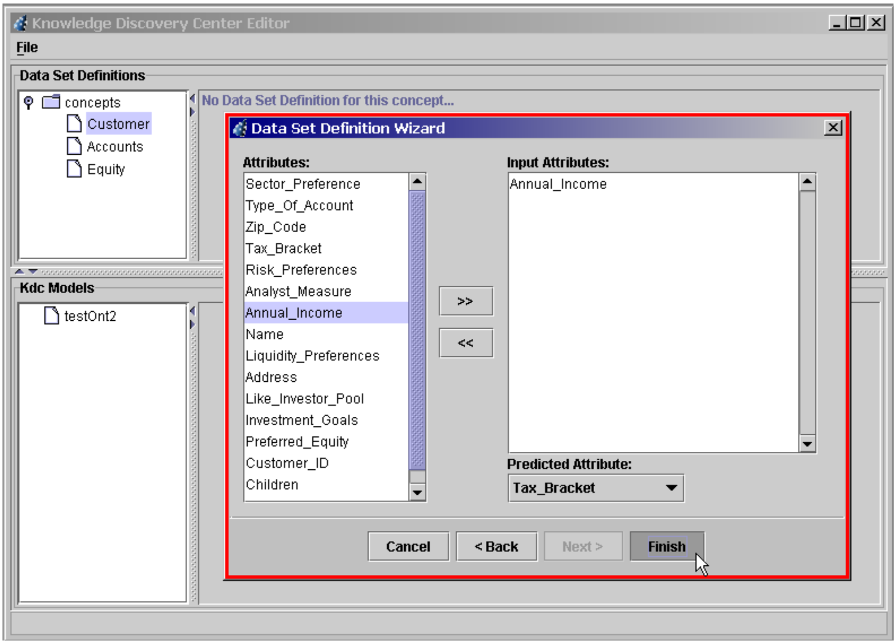
15 Use the cursor to select your desired Attribute.

Figure 11-14. Creating a Data Set - Selecting the Predicted Attribute



16 Click your selected Attribute to confirm your choice. The Data Set Definition Wizard display updates to reflect your choice of Predicted Attribute. Click Finish to complete.

Figure 11-15. Creating a Data Set - Predicted Attribute Displays



No Data Set Definition for this concept...

Attributes:

Sector_Preference
Type_Of_Account
Zip_Code
Tax_Bracket
Risk_Preferences
Analyst_Measure
Annual_Income
Name
Liquidity_Preferences
Address
Like_Investor_Pool
Investment_Goals
Preferred_Equity
Customer_ID
Children

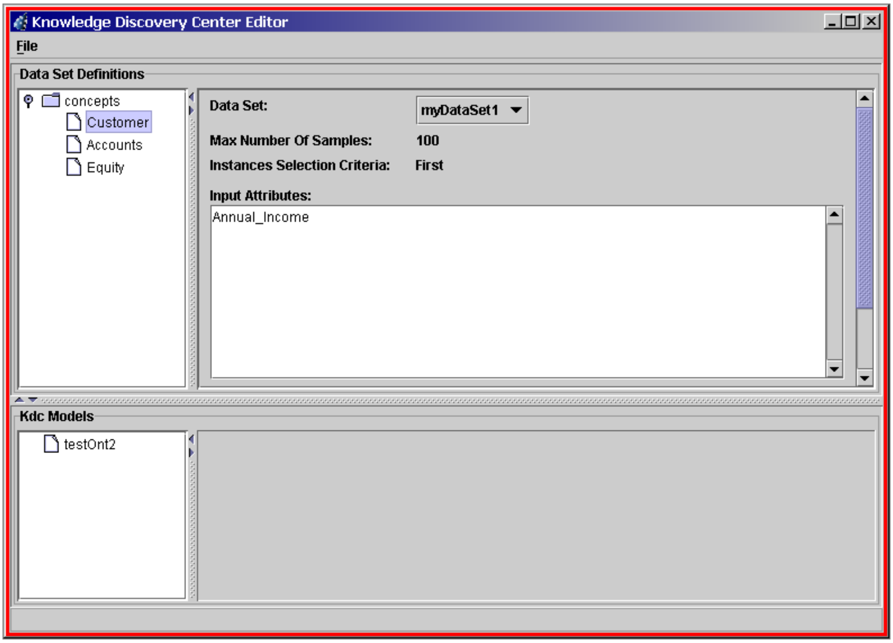
Input Attributes:

Annual_Income

Predicted Attribute: Tax_Bracket

17 The Knowledge Discovery Center display updates. The newly created Data Set is now visible in the right-hand panel. You have now finished creating a new Data Set.

Figure 11-16. Creating a Data Set - Completed Data Set Displays



Data Set: myDataSet1

Max Number Of Samples: 100

Instances Selection Criteria: First

Input Attributes:

Annual_Income

Deleting Data Sets

1 Open the Knowledge Discovery Center, if it is not currently open.

2 The available Concepts display in the Data Set Definitions upper left-hand panel. Click to select the Concept that contains the Data Set you want to delete.

3 The available Data Sets display in the upper right-hand Data Set drop-down panel. Click to select the Data Set you want to delete.

4 Select the menu item **File > Delete Data Set**.

5 The Knowledge Discovery Center display updates to reflect your Data Set deletion.

Editing Data Sets

1 Open the Knowledge Discovery Center, if it is not currently open.

2 The available Concepts display in the Data Set Definitions upper left-hand panel. Click to select the Concept that contains the Data Set you want to edit.

3 The available Data Sets display in the upper right-hand Data Set drop-down panel. Click to select the Data Set you want to edit.

4 Select the menu item **File > Edit Selected Data Set Definition...**

5 Use the Data Set Definition Wizard to make your changes. To complete your edits, click the Finish button.

6 The Data Set Definition Wizard display updates to reflect your modifications.

Using Models

You create and modify Models to train on selected Data Sets. Once trained, you can embed Models with the Ontology as Calculated Elements. These Calculated Elements can be thought of as sophisticated function calculators that will analyze input data and categorize it according to patterns discovered during training.

Knowledge Broker supports two types of model:

CBR

Case-based reasoning (CBR) works on the artificial intelligence principle that human reasoning does not stem solely from compiling a large set of rules and procedures. In real life, much reasoning is case-based rather than rule-based. People use their memories, or cases, to compare current situations to past situations, looking for similarities that will enable them to use their past experience in the new situation. It is reasoning through analogy.

In CBR, each case typically contains a description of the problem, plus a solution and/or the outcome. The knowledge and rules used by the domain expert to solve the problem are not explicitly recorded, but instead are implicit in the solution. To solve a current scenario, a CBR system matches the problem using similarity metrics against all stored cases and retrieves the most similar ones. The retrieved cases suggest a solution that is reused and tested for success. If necessary, the CBR system iterates, revising the current solution and resubmitting for success checking. After this process has concluded, the CBR system retains the current problem scenario and the final solution as part of a new case. The approach of CBR is summarized with the “4 Rs”: retrieval, reuse, revision, and retention.

Cases are stored as a description, a set of feature/value *n*-tuples, and an outcome. Testing for similarity means comparing the feature values and possibly using a ranking or precedence system to "promote" certain features.

Knowledge Broker's CBR implementation functions by:

1. Identifying a set of relevant problem descriptors.
2. Matching the case and returning a set of sufficiently similar cases (given a similarity threshold).
3. Selecting the best case from the set of cases returned.

MLP

A multi-layer perceptron (or neural network) emulates the way human brain cells learn pattern recognition and matching. The advantage of Knowledge Broker's MLP Models is that they can help you identify patterns in continuous data without requiring you to first build a knowledge base of rules or a case base to explain what is happening. In this sense, they are "dumb classifiers" that act as a black box classifier: they are unable to produce explanations for their output.

Creating a Model

1. Open the Knowledge Discovery Center, if it is not currently open.
2. The available root ontology within which to embed the Models display in the lower left-hand KDC Models panel. Click to select this root ontology.
3. Select the menu item **File > Create KDC Model**.
4. The Knowledge Discovery Center display updates to reflect your Model creation.

Choosing Which Model

The selection of which modeling approach to use is very domain-specific and requires careful analysis and experimentation. Some models may return better results for noisy data, or for incomplete data. Some models function better with discrete rather than continuous data.

CBR

A CBR model is advisable when:

- You have good data records of previously solved problems.
- You view historical cases and outcomes as an important asset with current applicability.

- You value experience and specific problem solving at least as highly as "book learning".
- Your domain experts or business users demonstrate or teach others by giving examples of "how to do things".

Case-based reasoning is often used where experts find it hard to articulate their thought processes when solving problems. This is because knowledge acquisition for a classical KBS would be extremely difficult in such domains, and is likely to produce incomplete or inaccurate results. When using case-based reasoning, the need for knowledge acquisition can be limited to establishing how to characterize cases.

CBR is advised when extracting rules-based problem-solving knowledge to create a knowledge base would be too challenging or time-consuming. Indeed, in this case a classical rules-based approach would probably produce inaccurate, incomplete, or "fragile" results. CBR's advantage here is in limiting the knowledge acquisition and formalization process as establishing the best way to categorize cases using the description, feature/value *n*-tuples, and outcome template.

Another advantage of CBR is that it enables incremental development of the case base. Additionally, the maintenance of the case base is relatively simple and can be easily done by your business domain experts, who may not welcome the complexity of instead maintaining a complex rules base.

Deleting a Model

1 Open the Knowledge Discovery Center, if it is not currently open.

2 The available Models display in the lower left-hand Kdc Models panel.

3 Select the menu item **File > Delete Kdc Model**.

4 The Kdc Model Wizard displays. When your model is complete, click the Finish button.

5 The Data Set Definition Wizard display updates to reflect your Model deletion.

Editing a Model

1 Open the Knowledge Discovery Center, if it is not currently open.

2 The available Models display in the lower left-hand Kdc Models panel.

3 Select the menu item **File > Edit Selected Kdc Model**...

4 The Kdc Model Wizard displays. When your model is complete, click the Finish button.

5 The Data Set Definition Wizard display updates to reflect your Model modifications.

Embedding a KDC Model

You can embed a KDC Model as a Calculated Element. See *Chapter 6, Understanding Calculated Elements on page 126*. The same restrictions for Calculated Elements apply to KDC Models. Both are only available to Business Concepts that are not associated with a pre-defined schema.

Part

IV

Tutorials

This section outlines various hands-on demonstrations of Knowledge Broker's user interface modules.

Chapter

12

Creating a Case Base

This chapter provides step-by-step instructions for using Knowledge Broker to build a Case Base. It uses sample data to demonstrate how to create data, make connections and mappings, and create and test a Case Base.

The steps you will follow in this chapter are:

1 Create Sample Data
2 Create a Connection
3 Create a Mapping
4 Create a Case Base
5 Test the Case Base

Before You Begin

Ensure that you have installed the Black Pearl Knowledge Broker.

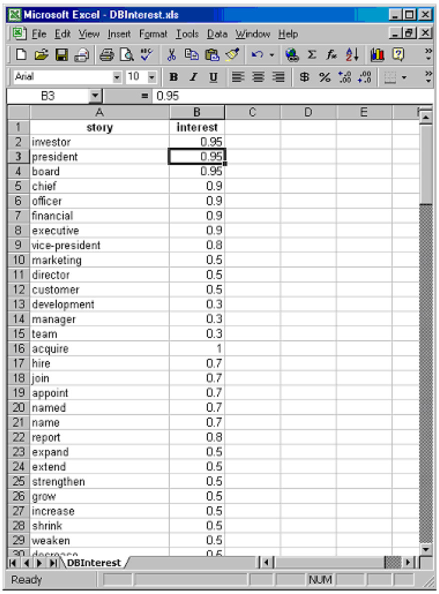
Creating Data

Creating Sample Data

To perform its role, Knowledge Broker must have access to data.

1 Create a sample data spreadsheet or use the `dbinterest.xls` located in the `[KBHOME]` directory.

Figure 12-1. Creating Data - Creating Sample Data - `dbinterest.xl`

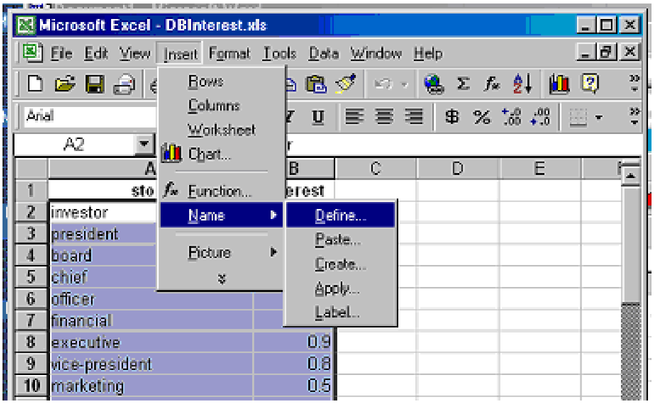

Microsoft Excel - DBInterest.xls
File Edit View Insert Format Tools Data Window Help
Arial
10
B3
= 0.95
DBInterest
Ready
NUM

	A	B	C	D	E
1	story	interest			
2	investor	0.95			
3	president	0.95			
4	board	0.95			
5	chief	0.9			
6	officer	0.9			
7	financial	0.9			
8	executive	0.9			
9	vice-president	0.8			
10	marketing	0.5			
11	director	0.5			
12	customer	0.5			
13	development	0.3			
14	manager	0.3			
15	team	0.3			
16	acquire	1			
17	hire	0.7			
18	join	0.7			
19	appoint	0.7			
20	named	0.7			
21	name	0.7			
22	report	0.8			
23	expand	0.5			
24	extend	0.5			
25	strengthen	0.5			
26	grow	0.5			
27	increase	0.5			
28	shrink	0.5			
29	weaken	0.5			
30	decrease	0.5			

2 You must define the data and rename it. In Windows Excel, high-light the data.

3 Select **Insert > Name > Define.**

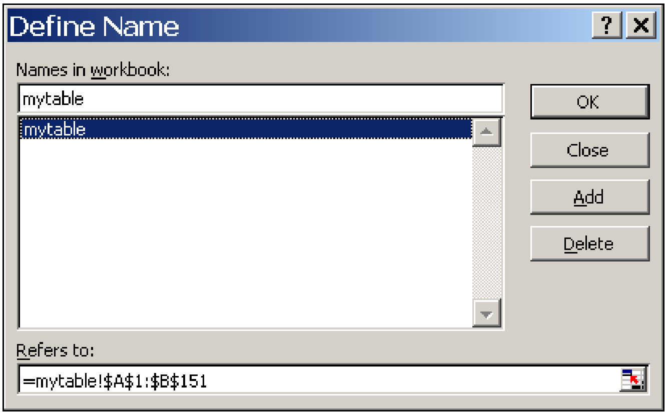
Figure 12-2. Creating Data - Creating Sample Data - Defining the Data



	A	B
1	sto	erest
2	investor	
3	president	
4	board	
5	chief	
6	officer	
7	financial	
8	executive	0.9
9	vice-president	0.8
10	marketing	0.5

The Define Name dialog displays.

Figure 12-3. Creating Data - Creating Sample Data - Define Data Name



Names in workbook: mytable

mytable

Refers to: =mytable!$A$1:$B$151

4 Type a name for the table and click the **OK** button to continue. You will use this sample data to predict a users interest in a text article based on keywords.

Configuring Sample ODBC Data Source Driver

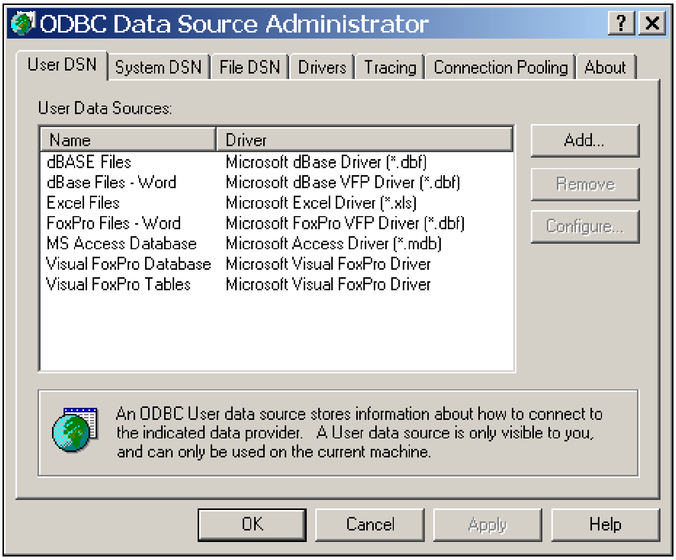
After you create data, you must configure a data source driver to point to the data.

Because it is possible to use Knowledge Broker with a variety of data source drivers, you must identify the driver before you can use Knowledge Broker. The Microsoft ODBC Data Source Administrator provides a convenient single location that aggregates many data source drivers for easy access. To create and and specify system-wide ODBC drivers, follow these steps:

1 **Select Start > Settings > Control Panel > ODBC Data Sources** (Windows NT4) or **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)** (Windows 2000). The ODBC Data Source Administrator displays, listing all currently configured data source drivers.

2 Click **User DSN** tab. The ODBC Data Source Administrator/System DSN dialog displays.

Figure 12-4. Creating Data - Configuring Sample ODBC - ODBC Data Source Administrator



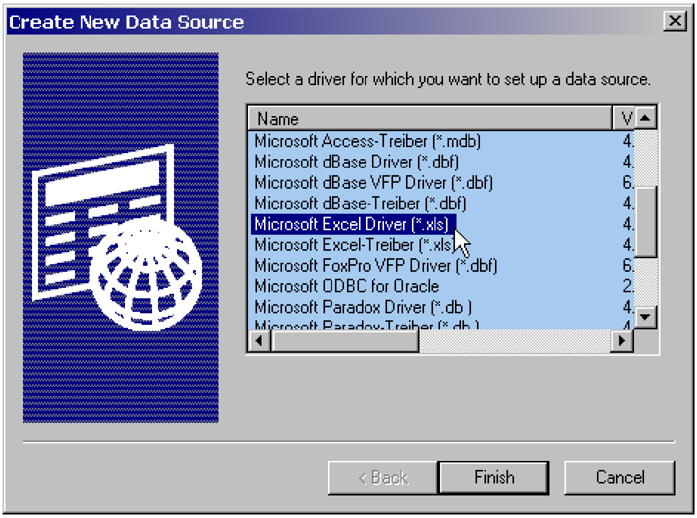
User Data Sources:

Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files - Word	Microsoft FoxPro VFP Driver (*.dbf)
MS Access Database	Microsoft Access Driver (*.mdb)
Visual FoxPro Database	Microsoft Visual FoxPro Driver
Visual FoxPro Tables	Microsoft Visual FoxPro Driver

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

3 Click the **Add** button. The Create New Data Source dialog displays.

Figure 12-5. Creating Data - Configuring Sample ODBC - Create New Data Source



Create New Data Source

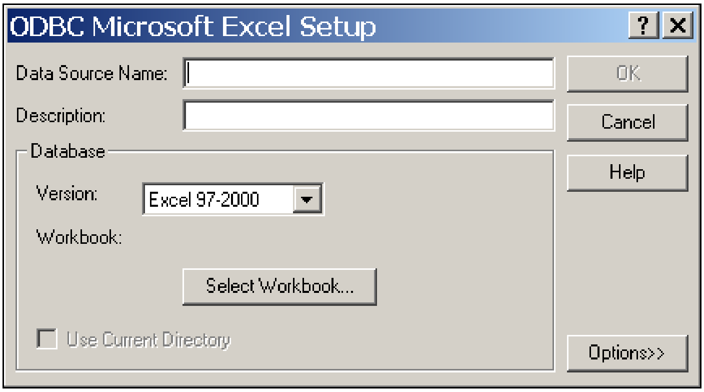
Select a driver for which you want to set up a data source.

Name	V
Microsoft Access-Treiber (*.mdb)	4.
Microsoft dBase Driver (*.dbf)	4.
Microsoft dBase VFP Driver (*.dbf)	6.
Microsoft dBase-Treiber (*.dbf)	4.
Microsoft Excel Driver (*.xls)	4.
Microsoft Excel-Treiber (*.xls)	4.
Microsoft FoxPro VFP Driver (*.dbf)	6.
Microsoft ODBC for Oracle	2.
Microsoft Paradox Driver (*.db)	4.
Microsoft Paradox-Treiber (*.db)	4.

4 Select the appropriate driver. In this example, select Microsoft Excel Driver (*.xls)

5 Click the **Finish** button. The ODBC Microsoft Excel Setup displays.

Figure 12-6. Creating Data - Configuring Sample ODBC - ODBC Microsoft Excel Setup



ODBC Microsoft Excel Setup

Data Source Name:

Description:

Database

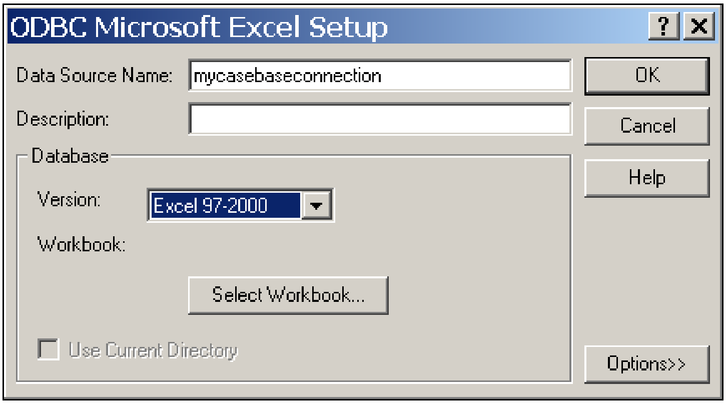
Version: Excel 97-2000

Workbook:

□ Use Current Directory

6 Type `mycasebaseconnection`, the name of your Data Source, in the **Data Source Name field**. Ensure the appropriate Excel Version is selected.

Figure 12-7. Creating Data - Configuring Sample ODBC - mycasebaseconnection



ODBC Microsoft Excel Setup

Data Source Name: mycasebaseconnection

Description:

Database

Version: Excel 97-2000

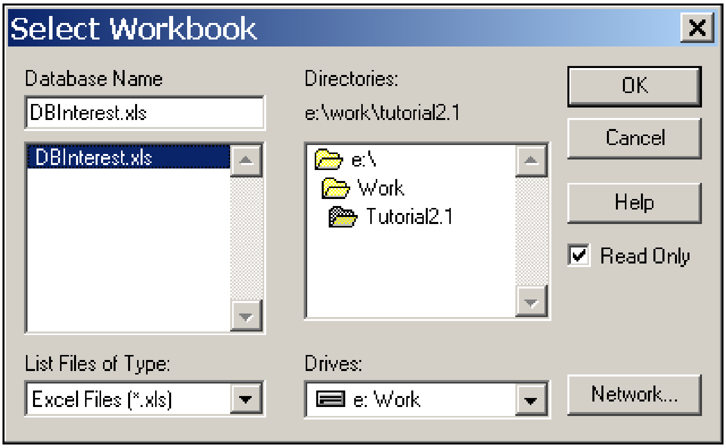
Workbook:

□ Use Current Directory

7 Click the **Select Workbook** button. The Select Workbook dialog displays.

8 Navigate to the location of the `DBInterest.xls` file, the sample data you created.

Figure 12-8. Creating Data - Configuring Sample ODBC - Select Workbook



Select Workbook

Database Name: DBInterest.xls

DBInterest.xls

Directories: e:\work\tutorial2.1

e:\
Work
Tutorial2.1

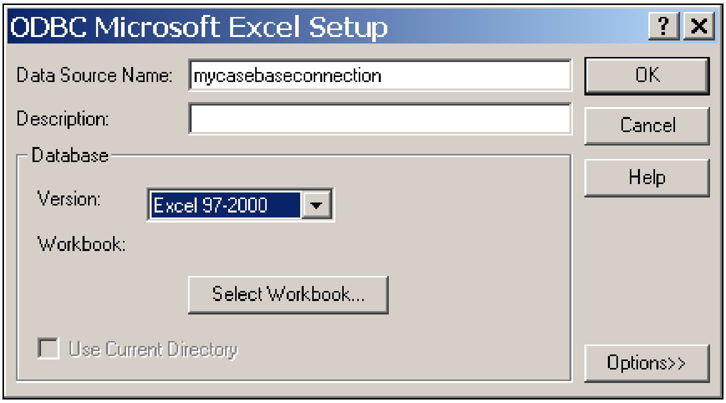
☑ Read Only

List Files of Type: Excel Files (*.xls)

Drives: e: Work

9 Click the **OK** button. The ODBC Microsoft Excel Setup dialog displays. You have just pointed your data source, `mycasebaseconnection`, to the sample data you created.

Figure 12-9. Creating Data - Configuring Sample ODBC - mycasebaseconnection2


ODBC Microsoft Excel Setup

Data Source Name: mycasebaseconnection

Description:

Database

Version: Excel 97-2000

Workbook:

☐ Use Current Directory

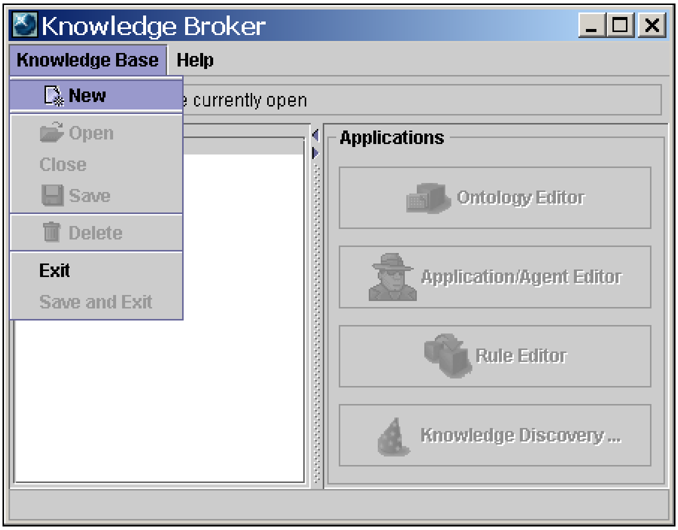
10 Click the **OK** button. Click the **OK** button again to exit the ODBC Data Source Administrator.

Creating a New Knowledge Base

After you point your data source to your data, you must create a knowledge base. You accomplish this in the Knowledge Broker Launcher.

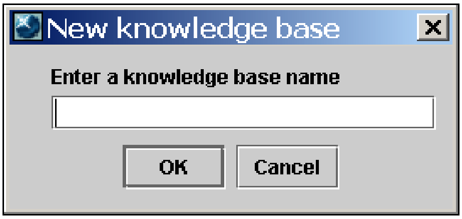
1 In the Launcher, select **Knowledge Base > New.**

Figure 12-10. Creating Data - Creating New Knowledge Base - Creating a New Knowledge Base



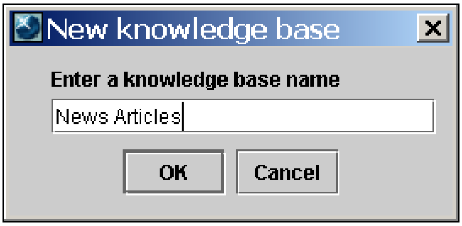
The New Knowledge Base dialog displays.

Figure 12-11. Creating Data - Creating New Knowledge Base - New Knowledge Base



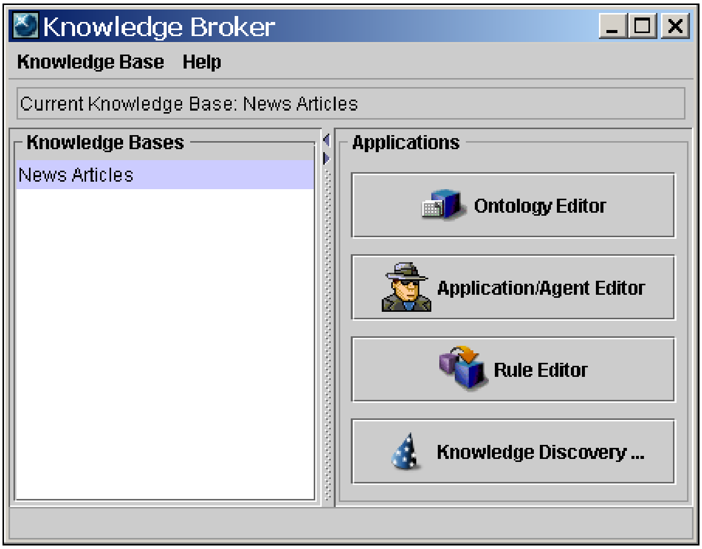
2 Type News Articles, the name of the new knowledge base, in the **Enter a knowledge base name** field.

Figure 12-12. Creating Data - Creating New Knowledge Base - Naming New Knowledge Base



3 Click the **OK** button. The new knowledge base displays in the Launcher.

Figure 12-13. Creating Data - Creating New Knowledge Base - Launcher - Knowledge Base

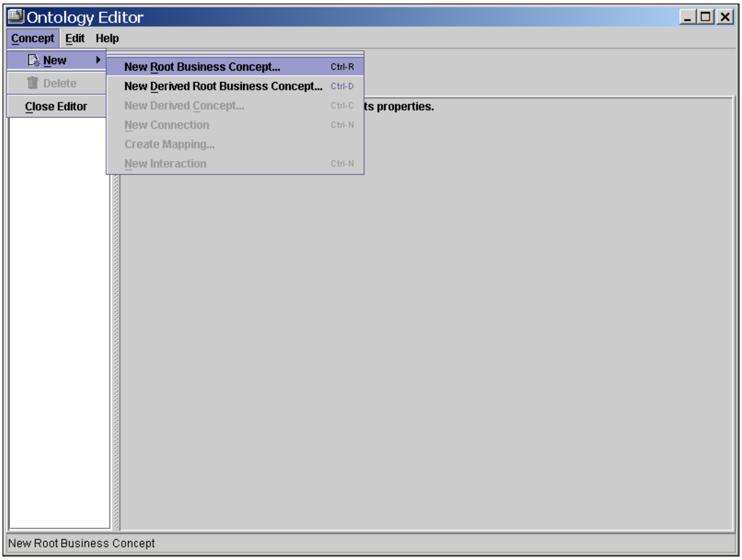


Creating a New Business Concept

After you have created a new knowledge base, you must give it meaning by creating business concepts. You accomplish this in the Ontology Editor.

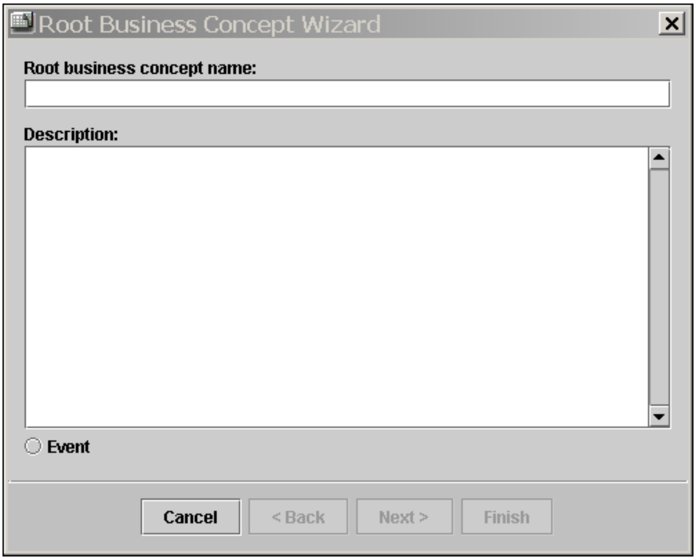
1 In the Ontology Editor, select **Concept > New > New Root Business Concept** (or select Concepts and option-click).

Figure 12-14. Creating Data - Creating a New Business Concept - New Root Business Concept


Ontology Editor
Concept
Edit
Help
New
Delete
Close Editor
New Root Business Concept... Ctrl-R
New Derived Root Business Concept... Ctrl-D
New Derived Concept... Ctrl-C
New Connection Ctrl-N
Create Mapping...
New Interaction Ctrl-N
ts properties.
New Root Business Concept

The Root Business Concept Wizard dialog displays.

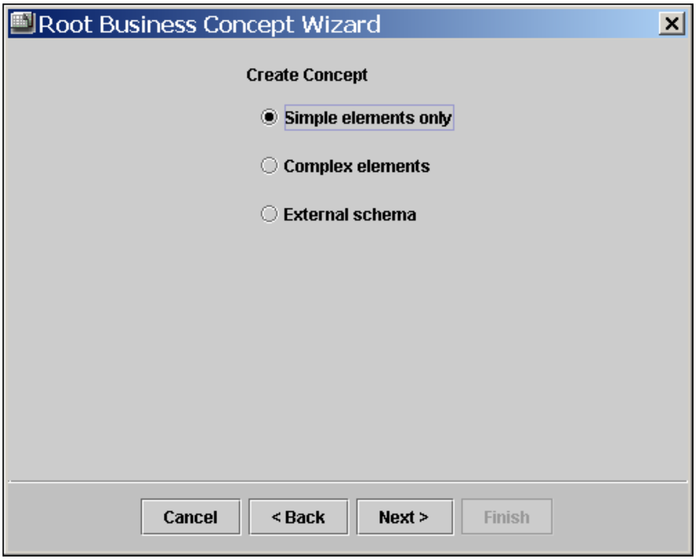
Figure 12-15. Creating Data - Creating a New Business Concept - Root Business Concept Wizard



2 In the **Root business concept name** field type `newsarticle`, the name of the new Business Concept.

3 Click the **Next** button. The Root Business Concept Wizard Schema Selector dialog displays.

Figure 12-16. Creating Data - Creating a New Business Concept - Schema Selector



Create Concept

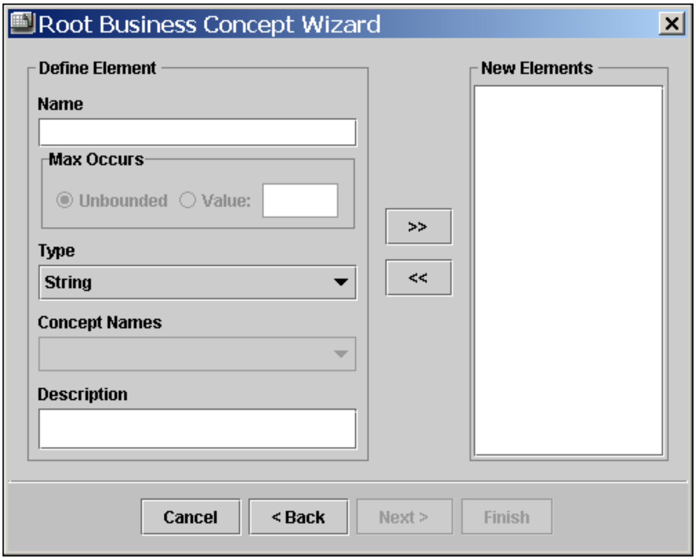
● Simple elements only

○ Complex elements

○ External schema

4 Select Simple Elements Only and click the **Next** button to continue. The Root Business Concept Wizard Define Element dialog displays.

Figure 12-17. Creating Data - Creating a New Business Concept - Define Elements

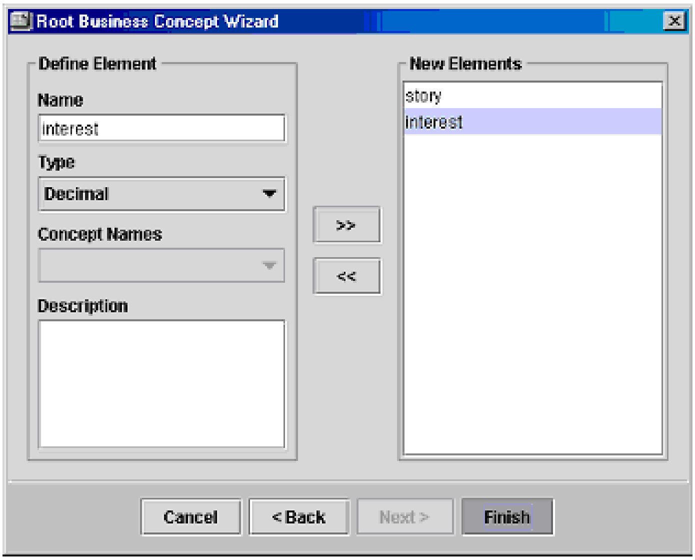


Max Occurs: ● Unbounded ○ Value:

Type: String

5 You must now define the elements of the Business Concept. In the **Name** field type `story`, the name of the first element. Select String as type from the Type pull-down menu and click the **>>** button to add the element to the Business Concept. The newly defined element displays in the New Elements field. Follow a similar procedure to create a new String as Type element named Interest.

Figure 12-18. Creating Data - Creating a New Business Concept - Define Elements



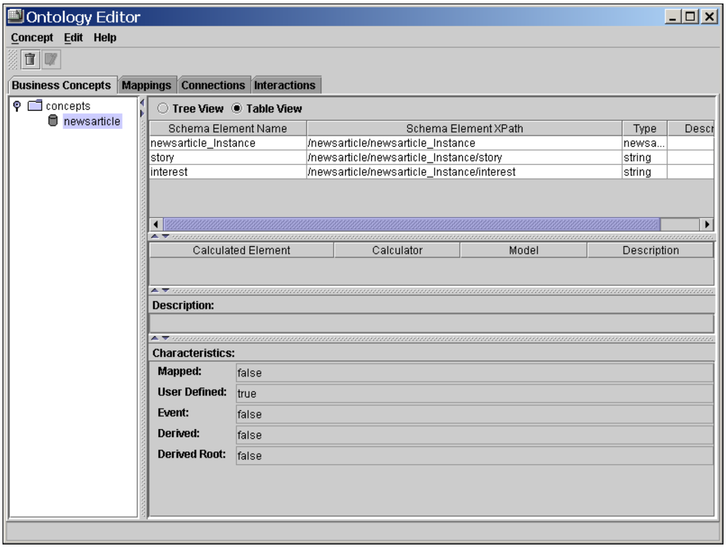
Name: interest

Type: Decimal

New Elements:
story
interest

6 Click the **Finish** button. The newly created Root Business Concept displays in the Ontology Editor.

Figure 12-19. Creating Data - Creating a New Business Concept - Ontology Editor - Business Concept



○ Tree View ● Table View

Schema Element Name	Schema Element XPath	Type	Descr
newsarticle_Instance	/newsarticle/newsarticle_Instance	newsa...	
story	/newsarticle/newsarticle_Instance/story	string	
interest	/newsarticle/newsarticle_Instance/interest	string	

Calculated Element	Calculator	Model	Description

Description:

Characteristics:

Mapped:	false
User Defined:	true
Event:	false
Derived:	false
Derived Root:	false

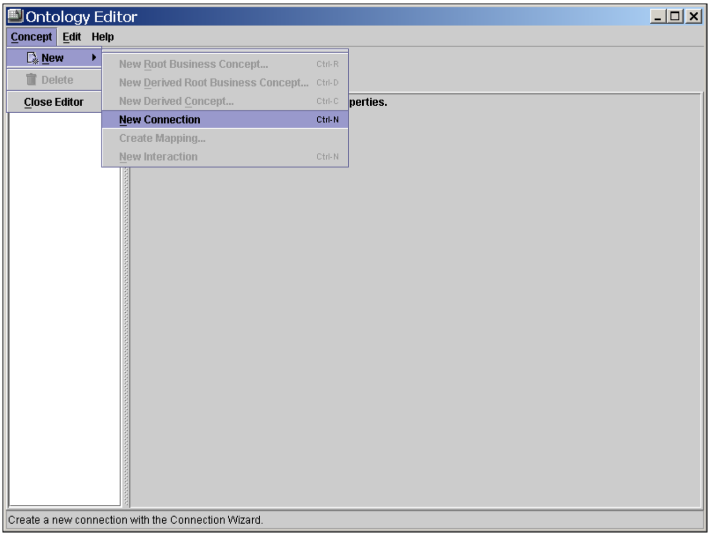
Creating a Connection

You must now create a connection in order to access the data you have created.

1 In the Ontology Editor, select the **Connections** tab.

2 Select **Concept > New > New Connection** (or select connections and option-click).

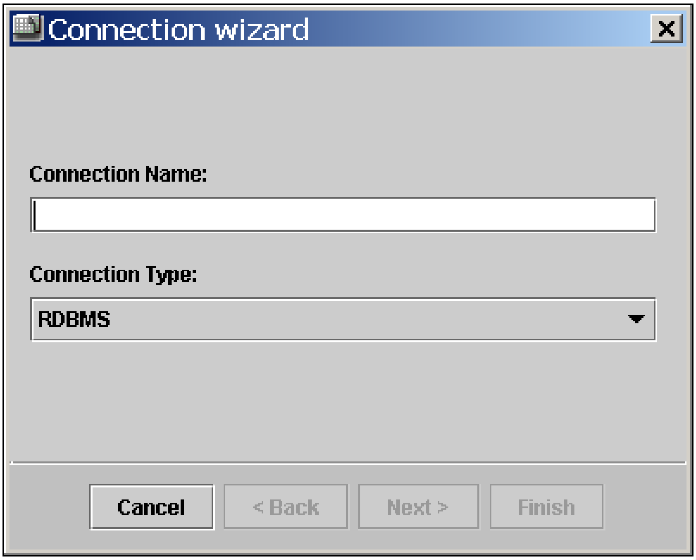
Figure 12-20. Creating a Connection - New Connection



Create a new connection with the Connection Wizard.

The Connection Wizard displays.

Figure 12-21. Creating a Connection - Connection Wizard



Connection wizard

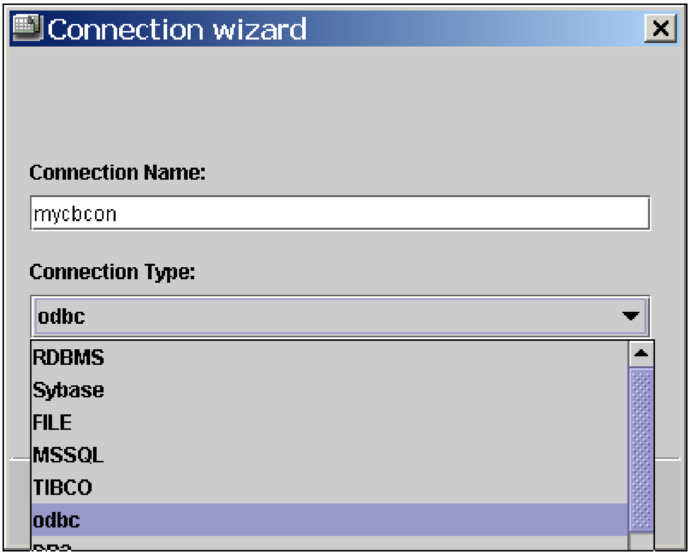
Connection Name:

Connection Type: RDBMS

3 Type `mycbcon`, the name of the new connection, in the **Connection Name** field.

4 Select ODBC from the Connection Type pull-down menu.

Figure 12-22. Creating a Connection - ODBC Connection



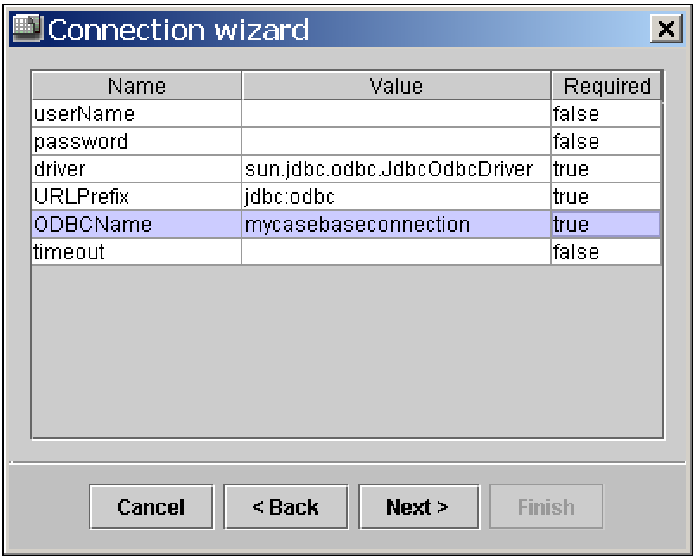
Connection Name: mycbcon

Connection Type: odbc

5 Click the **Next** button. The ODBC Connection Properties dialog displays.

6 Type `mycasebaseconnection`, the name of the ODBC Data Source in the ODBCName row.

Figure 12-23. Creating a Connection - ODBC Properties

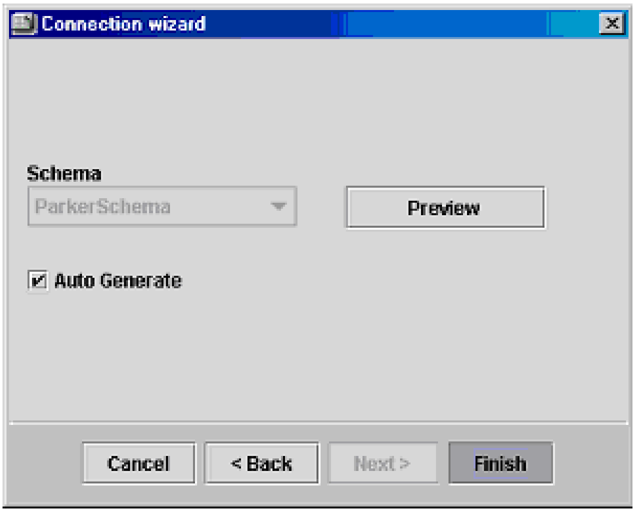


Name	Value	Required
userName		false
password		false
driver	sun.jdbc.odbc.JdbcOdbcDriver	true
URLPrefix	jdbc:odbc	true
ODBCName	mycasebaseconnection	true
timeout		false

7 Click the **Next** button. The Schema Selector dialog displays.

8 Select AutoGenerate.

Figure 12-24. Creating a Connection - Auto Generate Schema

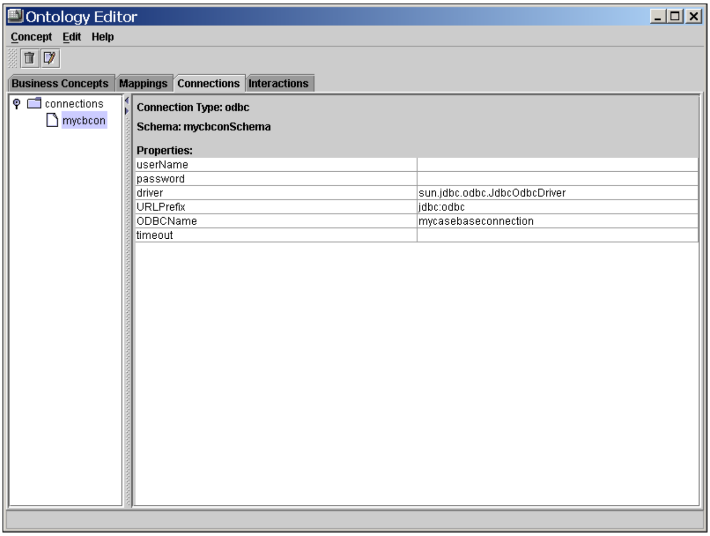


Schema
ParkerSchema

☑ Auto Generate

9 Click the **Finish** button. The newly created Connection displays in the Ontology Editor.

Figure 12-25. Creating a Connection - Ontology Editor - Connection



Connection Type: odbc

Schema: mycbconSchema

Properties:

userName	
password	
driver	sun.jdbc.odbc.JdbcOdbcDriver
URLPrefix	jdbc:odbc
ODBCName	mycasebaseconnection
timeout	

You have now created a Connection to the data.

Creating a Mapping

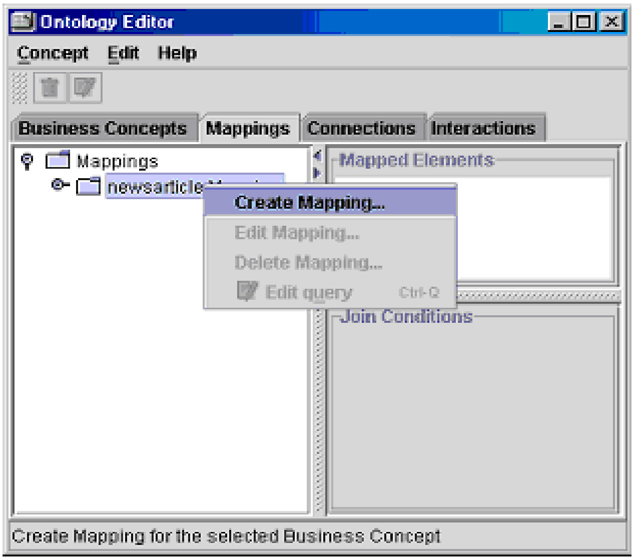
You must now “marry” the connection and the data by creating a Mapping.

1 In the Ontology Editor, select the **Mappings** tab.

2 Select the `newsarticle` Mapping.

3 Select **Concept > New > New Mappings** (or option-click) .

Figure 12-26. Creating a Mapping - New Mapping



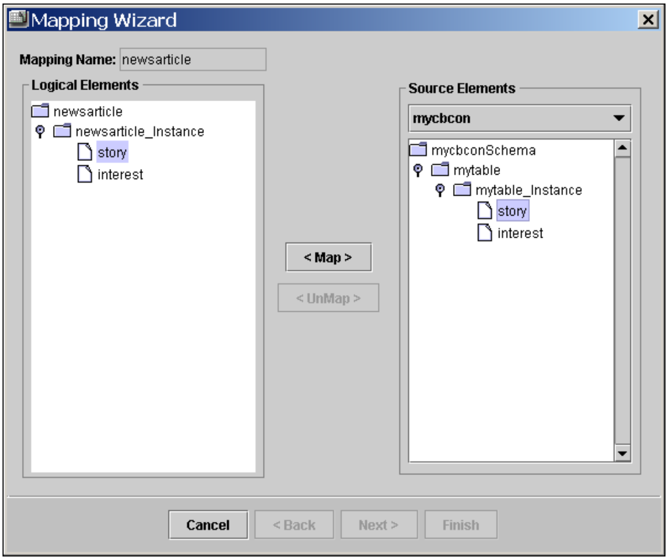
Create Mapping for the selected Business Concept

The Mapping Wizard displays. You must now select a Logical Element and map it to the corresponding Source Element.

4 On the left-side panel select story from the Logical Elements.

5 On the right-side panel select story from the Source Elements.

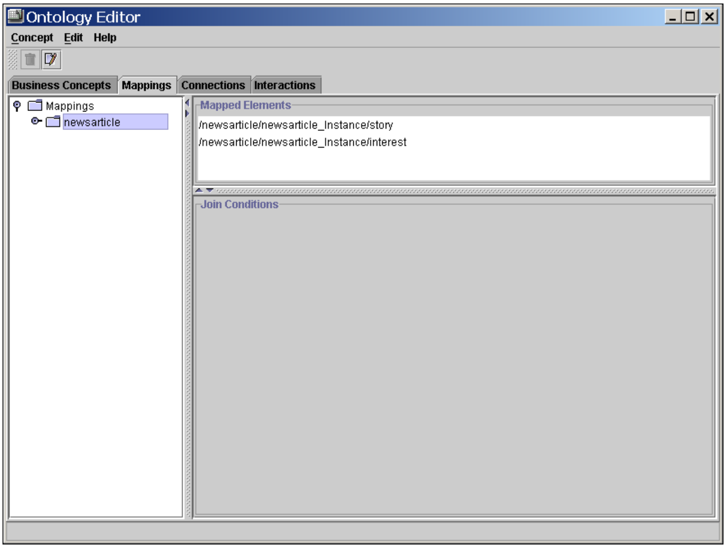
Figure 12-27. Creating a Mapping - Mapping Logical Elements to Source Elements



6 Click the **Map** button to map the two elements. To map interest, use a similar procedure.

7 Click the **Finish** button. The new Mapping displays in the Ontology Editor.

Figure 12-28. Creating a Mapping - Ontology Editor - Mapping



/newsarticle/newsarticle_Instance/story
/newsarticle/newsarticle_Instance/interest

You have now mapped a Business Concept, `newsarticle`, to your data.

Creating a Case Base

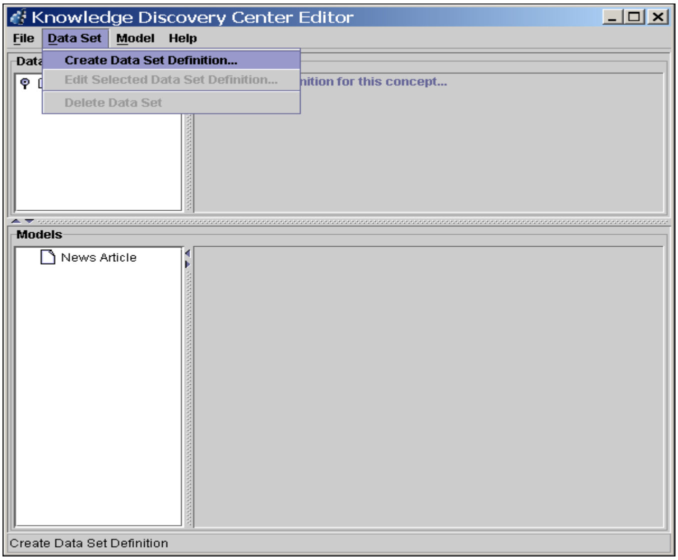
Creating a Data Set

You must now create a data set. A data set enables you to name the predictive model and determine the number of sample data elements you want to analyze.

1 From the Launcher, open the Knowledge Discovery Center.

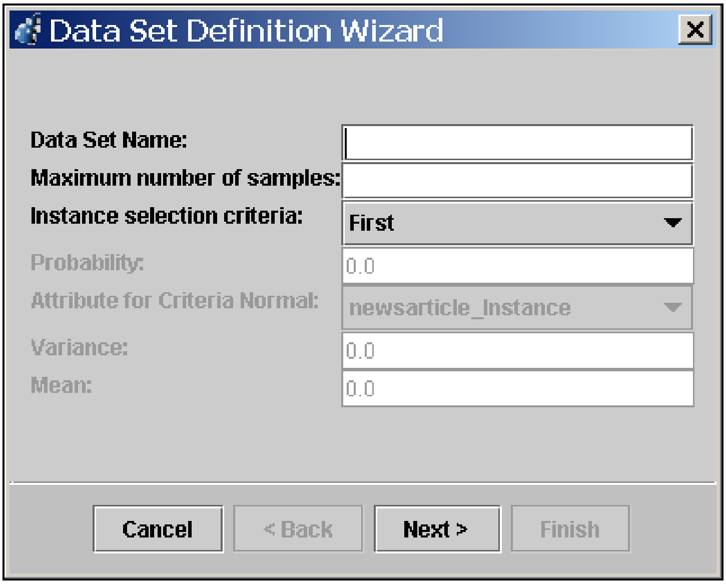
2 Select **newsarticle > Data Set > Create Data Set Definition** (or select **newsarticle** and option-click).

Figure 12-29. Creating a Case Base - Creating a Data Set - New Data Set


Knowledge Discovery Center Editor
File
Data Set
Model
Help
Create Data Set Definition...
Edit Selected Data Set Definition...
Delete Data Set
Data
nition for this concept...
Models
News Article
Create Data Set Definition

The Data Set Definition Wizard displays.

Figure 12-30. Creating a Case Base - Creating a Data Set - Data Set Definition Wizard



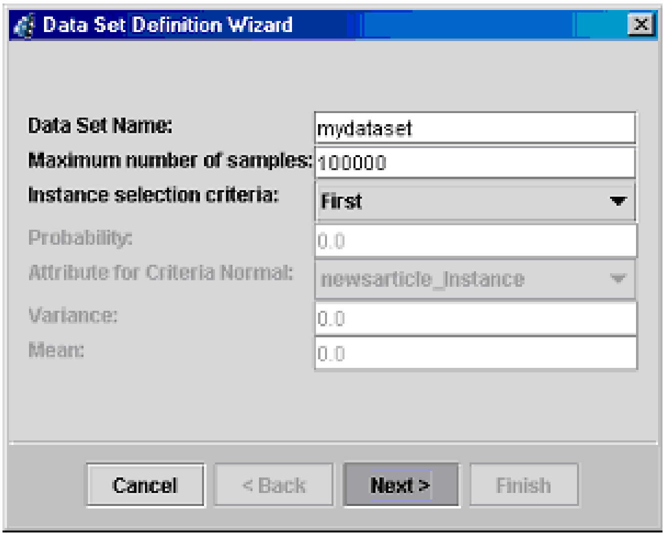
Data Set Name:
Maximum number of samples:
Instance selection criteria: First
Probability: 0.0
Attribute for Criteria Normal: newsarticle_Instance
Variance: 0.0
Mean: 0.0

You must now give your data set a name and tell it to build a data set from the first set of data.

3 Type `mydataset`, the Data Set Name, in the **Data Set Name:** field.

4 Set the number of samples to be larger than the size of your case base. In this example, type `10000` as the Maximum number of samples. Ensure that you have selected First from the Instance Selection Criteria pull-down menu.

Figure 12-31. Creating a Case Base - Creating a Data Set - Building a Data Set



Data Set Definition Wizard

Data Set Name: mydataset

Maximum number of samples: 100000

Instance selection criteria: First

Probability: 0.0

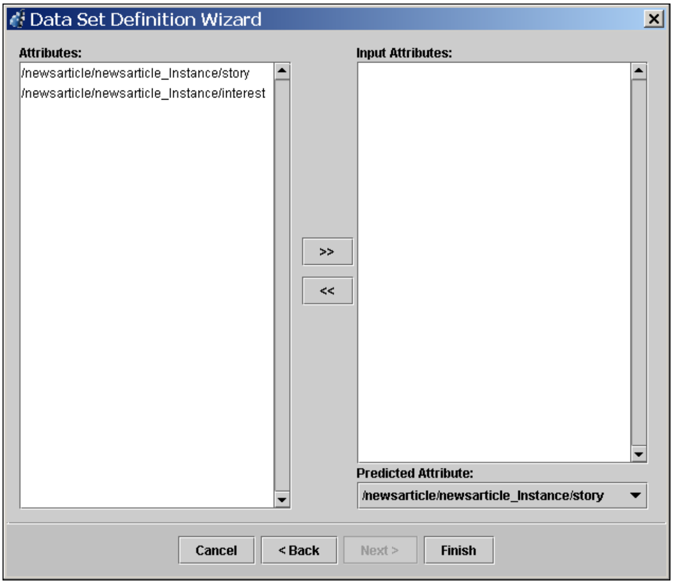
Attribute for Criteria Normal: newsarticle_Instance

Variance: 0.0

Mean: 0.0

5 Click the **Next** button. The Data Set Definition Wizard displays.

Figure 12-32. Creating a Case Base - Creating a Data Set - Building a Data Set



/newsarticle/newsarticle_Instance/story
/newsarticle/newsarticle_Instance/interest

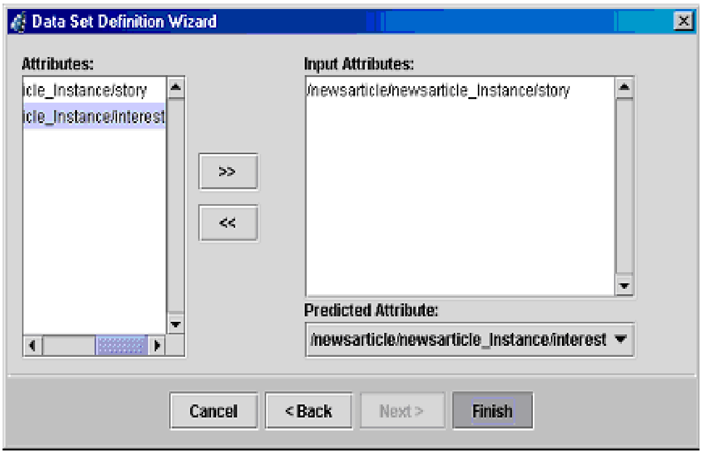
Predicted Attribute: /newsarticle/newsarticle_Instance/story

The model must be “trained” to make the most accurate predictions.

6 From the left-side panel select story, and click the **>>** button. Story displays as an Input Attribute.

7 From the Predicted Attribute drop-down menu select interest.

Figure 12-33. Creating a Case Base - Creating a Data Set - Select Attributes



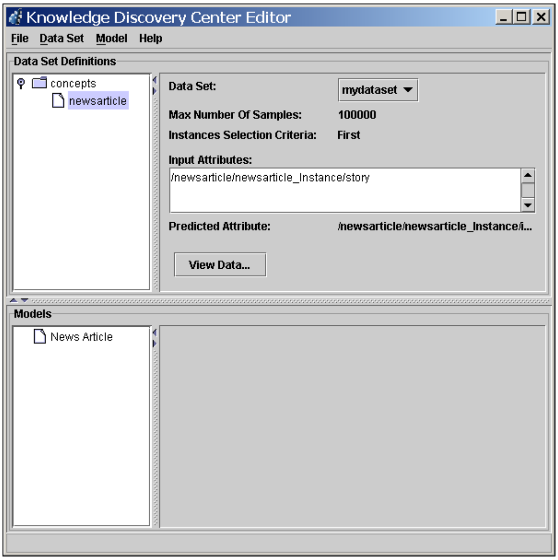
Attributes:
icle_Instance/story
icle_Instance/interest

Input Attributes:
/newsarticle/newsarticle_Instance/story

Predicted Attribute:
/newsarticle/newsarticle_Instance/interest

8 Click the **Finish** button. The newly created Data Set displays in the Knowledge Discover Center Editor.

Figure 12-34. Creating a Case Base - Creating a Data Set - KDC - Data Set



Data Set: mydataset

Max Number Of Samples: 100000

Instances Selection Criteria: First

Input Attributes:
/newsarticle/newsarticle_Instance/story

Predicted Attribute: /newsarticle/newsarticle_Instance/i...

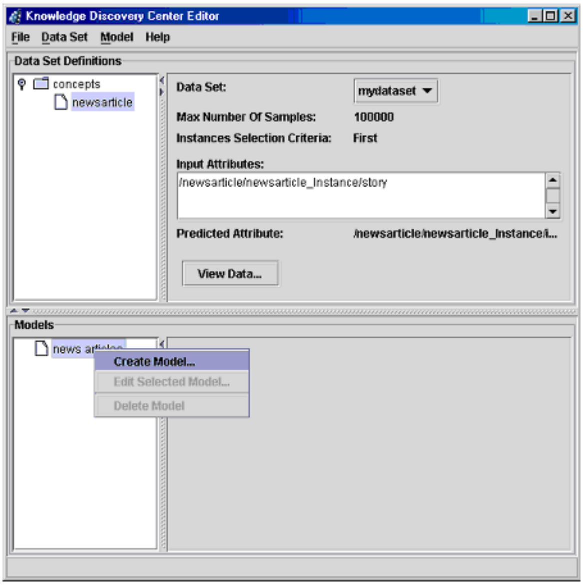
Building a Case Base

You must now build a case base from the data set model.

1 In the Knowledge Discovery Center Editor, Select News Article from the bottom-left Models panel.

2 Select **News Article > Model > Create Model** (or option-click on News Article).

Figure 12-35. Creating a Case Base - Building a Case Base - New Model



Data Set: mydataset

Max Number Of Samples: 100000

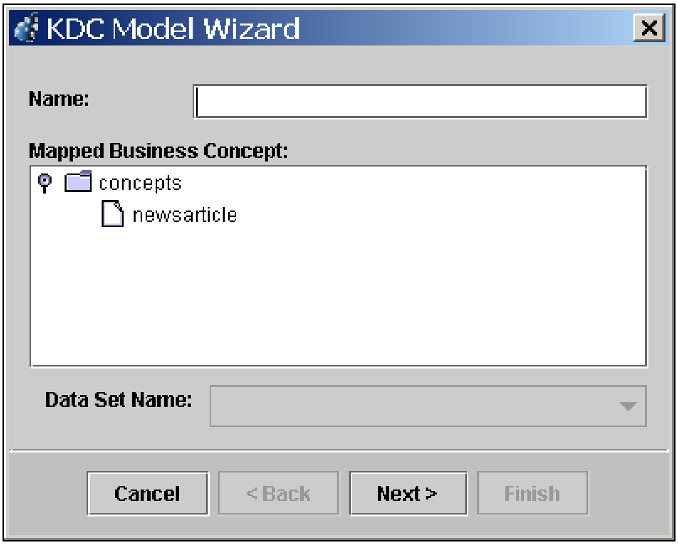
Instances Selection Criteria: First

Input Attributes:
/newsarticle/newsarticle_Instance/story

Predicted Attribute: /newsarticle/newsarticle_Instance/i...

3 The KDC Model Wizard displays.

Figure 12-36. Creating a Case Base - Building a Case Base - KDC Model Wizard



KDC Model Wizard

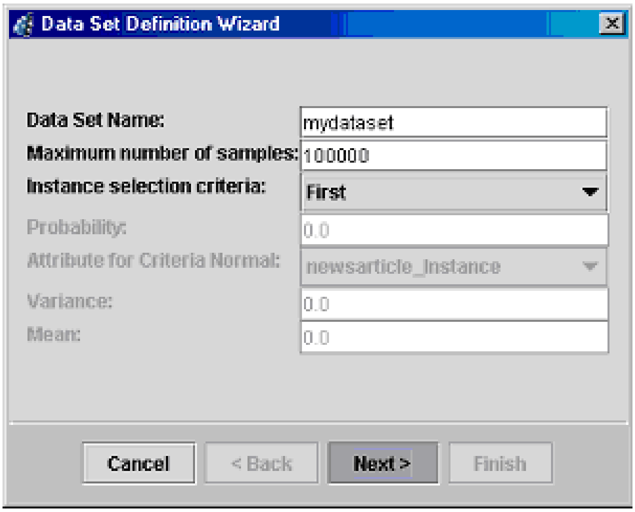
Name:

Mapped Business Concept:

4 You must now name your case base. Type `mycasebase`, the name of the new case base in the **Name:** field.

5 To train the data set, select the appropriate Business Concept. In this example, select `newsarticle`. Ensure that you have selected `mydataset` from the Data Set Name pull-down menu.

Figure 12-37. Creating a Case Base - Building a Case Base - Data Set Definition

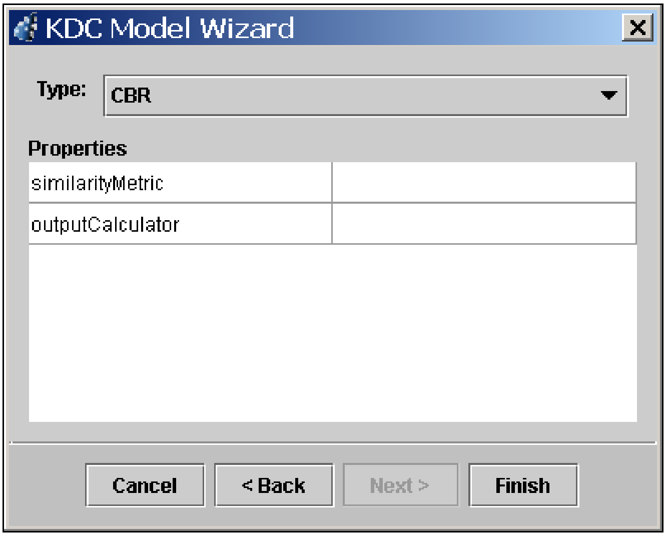


Data Set Definition Wizard

Data Set Name: mydataset
Maximum number of samples: 100000
Instance selection criteria: First
Probability: 0.0
Attribute for Criteria Normal: newsarticle_Instance
Variance: 0.0
Mean: 0.0

6 Click the **Next** button. The KDC Model Wizard displays.

Figure 12-38. Creating a Case Base - Building a Case Base - KDC Model Properties



Type: CBR

Properties

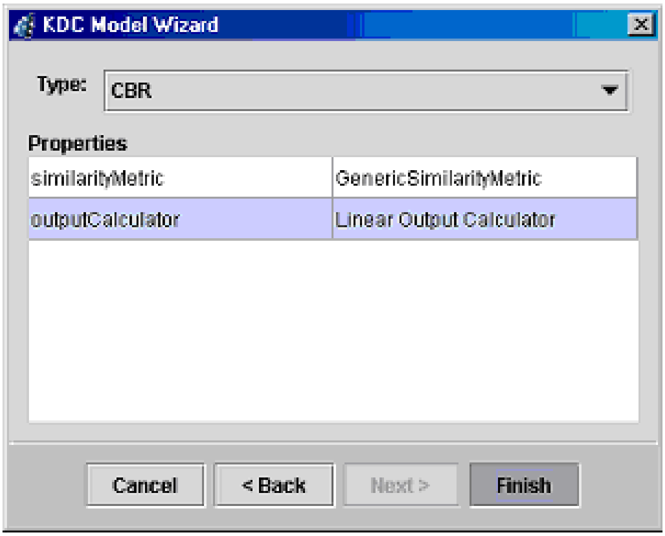
similarityMetric	
outputCalculator	

7 Select CBR from the **Type:** pull-down menu.

8 Select the similarityMetric row from **Properties** and select GenericSimilarityMetric from the pull-down menu.

9 Select the outputCalculator row from **Properties** and select Linear Output Calculator form the pull-down menu.

Figure 12-39. Creating a Case Base - Building a Case Base - KDC Model Properties2



KDC Model Wizard

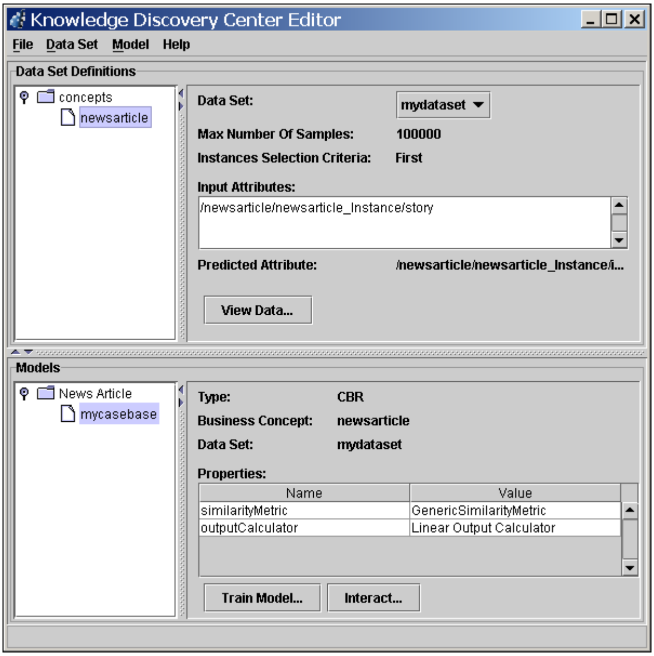
Type: CBR

Properties

similarityMetric	GenericSimilarityMetric
outputCalculator	Linear Output Calculator

10 Click the **Finish** button. The Case Base displays in the Knowledge Discovery Center.

Figure 12-40. Creating a Case Base - Building a Case Base - KDC Editor - Case Base



Data Set: mydataset

Max Number Of Samples: 100000

Instances Selection Criteria: First

Input Attributes:
/newsarticle/newsarticle_Instance/story

Predicted Attribute: /newsarticle/newsarticle_Instance/i...

Type: CBR

Business Concept: newsarticle

Data Set: mydataset

Properties:

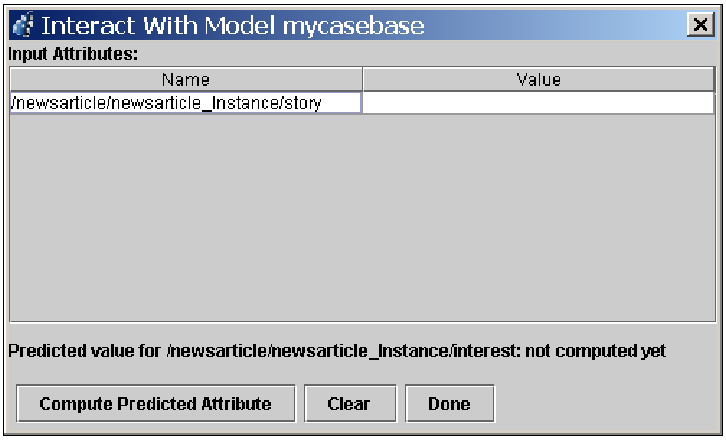
Name	Value
similarityMetric	GenericSimilarityMetric
outputCalculator	Linear Output Calculator

Testing the Case Base

After you have created a Case Base, you can test it.

1 In the Knowledge Discovery Center, click Interact. The Interact with Model mycasebase dialog displays.

Figure 12-41. Testing a Case Base - Interact with Model

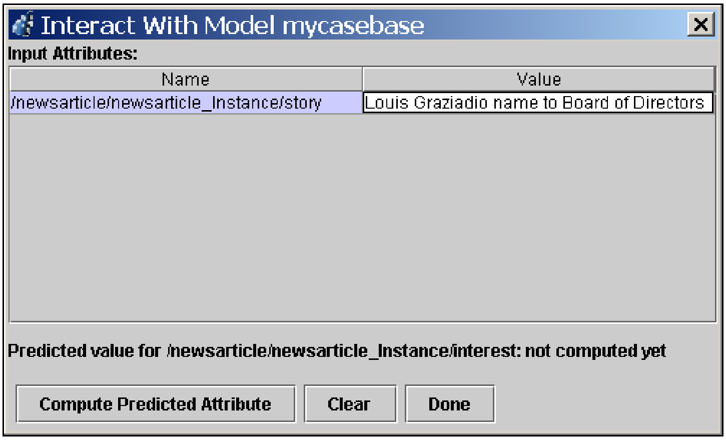


Name	Value
/newsarticle/newsarticle_Instance/story	

Predicted value for /newsarticle/newsarticle_Instance/interest: not computed yet

2 Type `Louis Graziadio named to Board of Directors`, the value for story, in the **Value** field.

Figure 12-42. Testing a Case Base - Interact with Model Value

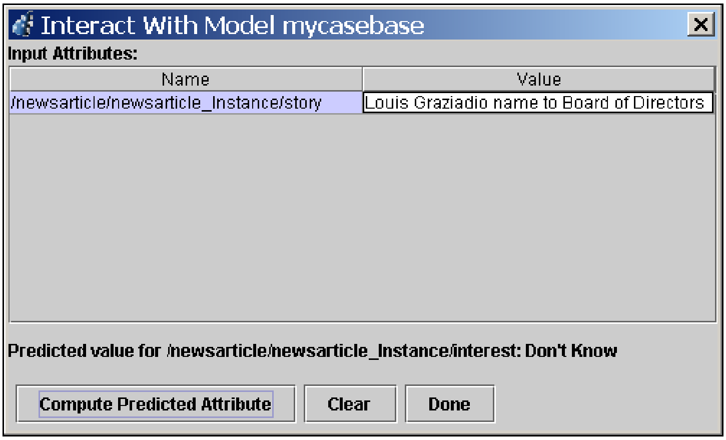


Name	Value
/newsarticle/newsarticle_Instance/story	Louis Graziadio name to Board of Directors

Predicted value for /newsarticle/newsarticle_Instance/interest: not computed yet

3 Click **Compute Predicted Attribute**. The Predicted Value for your request displays.

Figure 12-43. Testing a Case Base - Value displays



Input Attributes:

Name	Value
/newsarticle/newsarticle_Instance/story	Louis Graziadio name to Board of Directors

Predicted value for /newsarticle/newsarticle_Instance/interest: Don't Know

Part V

Appendices

The Appendices provide reference material, a glossary, and an index.

Appendix

A

JSP Interface

Java Server Pages (JSPs) enable application programmers to create and develop web-based interfaces to the run-time version of Knowledge Broker.

Examining Knowledge Broker's JSP Interface

The main class that defines the application program interface (API) for Knowledge Broker web access is:

```
com.blackpearl.api.web.WebEventHandler
```

The class that implements this interface is:

```
com.blackpearl.web.WebEventHandlerImpl
```

The public methods available are:

```
public void setHome(String home);
public void setApplication(String runtime);
public Cursor executeFlow(Map request) throws WebEventException.
```

The `setHome()` method sets the `com.blackpearl.home` variable
(for example, to `D:/blackpearl/config`).

The `setApplication()` method sets the com.blackpearl.application variable (for example, to `MortgageDemo`).

Both these setter methods must be called before calling the `executeFlow()` method. This execution method's argument is a Map that contains the following key-value pairs:

Table 0-1. The `executeFlow(Map request)` Method's Key-Valye Pairs

Key	**Value**	**Required**
conceptName	**Name of the web event concept**	**Yes**
attribute1	Value of attribute1	No
attribute2	Value of attribute2	No
...	...	...
attributeN	Value of *N*th attribute	No

All keys and values are case-sensitive.

Further information about JSPs is available here:

```
http://java.sun.com/products/jsp/
```

Using Knowledge Broker's JSP Interface

This section describes sample code that uses a small utility, Cursor2XMLConvertor, to convert an incoming Cursor (Knowledge Broker's in-memory data format) into XML.

The Cursor interface is described here:

```
com.blackpearl.api.connector.Cursor
```

Analyzing the JSP Code

```
<%@ page import="com.blackpearl.api.web.*,
                 com.blackpearl.web.*,
                 com.blackpearl.api.connector.Cursor,
         import com.blackpearl.dis2.common.Cursor2XMLConverter,
                java.rmi.*,
                javax.rmi.*,
                javax.ejb.*,
                javax.naming.*,
                java.util.*"%>
<%
    String concept= request.getParameter("conceptName");
    String id = request.getParameter("PortfolioID");
    String result = null;

    try
    {
                WebEventHandler handler = new WebEventHandlerImpl();
                handler.setHome("D:/fcs360/config");
                handler.setApplication("PortfolioExample");

                Map requestMap = new HashMap();
                requestMap.put("conceptName",concept);
                requestMap.put("PortfolioID",id);

                Cursor resultCursor = handler.executeFlow(requestMap);

                Cursor2XMLConverter conv = new Cursor2XMLConverter();
                conv.handleCursor(resultCursor);
                System.out.print("Results are:\n" + conv.getXML());
    }
    catch(WebEventException we)
    {
        out.println("Exception in JSP");
        we.printStackTrace(System.out);
    }
    catch(Exception e)
    {
        out.println("Exception in JSP");
        e.printStackTrace(System.out);
    }
%>
```

Transforming the Output

The raw XML output can be further transformed using an XSL Transformation (XSLT) user-defined stylesheet. These instructions assume you are using WebLogic 6.1.

1 From `[BEA_HOME]\ext\xmlx.zip`, extract a file called `xmxl-tags.jar` into any directory in your Web application.

  For example, you could extract the file into the following directory:

  ```
  [BEA_HOME]\config\mydomain\applications\DefaultWebApp\WEB-INF\lib
  ```

2 Using the WebLogic Console, navigate to the following location:

  mydomain→Deployments→WebApplications→KBRuntime→
  Edit Web Application Descriptor.

3 A new window pops up. Navigate the following location:

  Web Descriptor→Web App Descriptor→Tag Libs→Configure a new Tag Lib

  a Enter `taglib.tld` in the URI field.

  b Enter `/WEB-INF/lib/xmlx-tags.jar` in the location field. This location is relative to your Web Application's root.

  c Persist the changes.

4 Add the following lines to your JSP file:

```
<%@ taglib uri="taglib.tld" prefix="x"%>

<%
...
   JSP code here
...
%>

<x:xslt media="html">
        <x:xml><%=result%></x:xml>
        <x:stylesheet media="html" uri="myfile.xsl"/>
</x:xslt>
```

The `result` variable stores the XML string containing the inferencing results.

The `myfile.xsl` URI defines the location of the user-defined XSL file. This file must be in the same directory as the JSP and HTML files.

Further information about XSLTs is available here:

```
http://www.w3.org/TR/xslt
http://www.xml.com/pub/a/2000/08/holman/s1.html
```

Appendix

B

Java Exits

Java Exits enable advanced users to provide extra functionality within the Application/Agent Editor using programmatic Java code.

Examining Knowledge Broker's Java Exit Interface

The Application/Agent Editor is a GUI interface that enables users to configure and sequence commands that will be presented to the Flow Engine, a Knowledge Broker infrastructure component responsible for control flow.

Using the Java Exit command, advanced users can provide extra functionality not present in the existing Flow Engine commands.

Making a Java Exit

A Java Exit must contain a class that implements this interface:

```
com.blackpearl.api.flowengine.FlowJavaExit
```

The class must implement the `execute()` method.

Creating "Hello World"

This Java Exit will print "Hello World"

```
package examples.HelloWorld;
import com.blackpearl.api.flowengine.*;
public class HelloWorld implements FlowJavaExit {
    public Object execute(WorkingMemory memory) throws
FlowRuntimeException {
        System.out.println("Hello World");
        return null;
    }
}
```

To activate this program:

1. Select the Java Exit command within the flow
2. Enter `examples.HelloWorld`

Interacting with Flow Variables

The working memory passed to the Java Exit holds all references to variables defined in the rest of the flow. It is also possible to create new variables within the working memory that will be used in the rest of the flow.

Getting a Flow Variable

The working memory getter method is:

```
getUserVariable(String name).
```

For example, to retrieve a variable called age:

```
FlowVariable ageVariable = memory.getUserVariable("age");
String variableName = ageVariable.getName();
Object variableValue = ageVariable.getValue();
```

Note that the value is always an object and that casting is manual (that is, converting the above variableValue into a double).

Setting a Flow Variable

The working memory setter method is:

```
setUserVariable(FlowVariable variable).
```

For example, to create a variable called age:

```
FlowVariable ageVariable = new FlowVariableImpl("age", new Double(21));
    // Note:  FlowVariableImpl is not public,
    //        Users must implement own FlowVariables
memory.setUserVariable(ageVariable);
```

Modifying a Flow Variable

To modify a flow variable, grab the desired variable, then set the new value.

For example, to modify a variable called age:

```
FlowVariable ageVariable = memory.getUserVariable("age");
ageVariable.setValue(new Double(21));
```

Using System Variables

The WorkingMemory class contains a getSystemVariable() method. Certain commands (including Java Exits) require the existence of System Variables. These can be accessed similarly to the previously described User Flow Variables.

Because Knowledge Broker services such as the AdminService or the XQueryEngine use them, Black Pearl recommends you do not modify the System Variables. However, you can read them and duplicate their contents for your own use.

The class `FlowEngineConstants` contains several *VALUE*_VAR_NAME constants that define the names of the System Variables. Some key constants here are:

```
public static final String ADMIN_SERVICE_VAR_NAME = "AdminService";
```

The Admin Service accesses all Knowledge Broker functionality.

```
public static final String XQUERY_ENGINE_VAR_NAME = "XQueryEngine";
```

The XQueryEngine executes user-defined XQueries.

```
public static final String PROCESS_CONTEXT_VAR_NAME = "Process Context";
```

The process context enables you to set more variables, read values, and so on.

```
public static final String INTERACTION_LIST_VAR_NAME = "Interaction List";
```

Making Java Exits Thread-Safe

Because Java Exits could be accessed concurrently, you should make your code thread-safe. If you access or use System Variables, ensure you synchronize on the desired variable value. for example, if you are using the AdminService, synchronize on the adminService object.

Coding Java Exits

This Java Exit example executes a database update.

```
package com.blackpearl.application.dbab;

import com.blackpearl.api.flowengine.FlowJavaExit;
import com.blackpearl.api.flowengine.WorkingMemory;
import com.blackpearl.api.flowengine.FlowRuntimeException;
import com.blackpearl.api.flowengine.FlowEngineConstants;

import com.blackpearl.api.admin.AdminService;
import com.blackpearl.api.descriptor.ConnectionDescriptor;

import java.sql.*;

/**
 * Executes an update against a datasource.
 *
 * Expects two parameters in the WorkingMemory.
 * 1)  connectionDescriptor - points to the connectionDescriptor
 *                               of the rdbms datasource.
 * 2)  updateQuery which contains the SQL to execute.
 */
public class JdbcUpdate implements FlowJavaExit {

    public JdbcUpdate() {
    }
    public Object execute(WorkingMemory memory) throws FlowRuntimeException
{
        // pull the two parameters out of the memory
        String connectionDescriptor = null;
        String updateQuery = null;
        if (memory.getUserVariable("connectionDescriptor") == null) {
            throw new
           FlowRuntimeException("connectionDescriptor parameter required");
        }
        if (memory.getUserVariable("updateQuery") == null) {
            throw new
                    FlowRuntimeException("updateQuery parameter required");
        }
        connectionDescriptor = (String)
            memory.getUserVariable("connectionDescriptor").getValue();
        updateQuery = (String)
            memory.getUserVariable("updateQuery").getValue();
        AdminService admin = (AdminService) memory.

getSystemVariable(FlowEngineConstants.ADMIN_SERVICE_VAR_NAME).getValue();
        ConnectionDescriptor cxnDesc =
            admin.getConnectionDescriptor(connectionDescriptor);
        try {
            Connection cxn = (Connection) cxnDesc.getConnection();
            Statement stmt = cxn.createStatement();
            stmt.executeUpdate(updateQuery);
        } catch (com.blackpearl.api.descriptor.DescriptorException de) {
            throw new FlowRuntimeException("Could not get connection to
rdbms.  Nested Exception: " + de);
        } catch (SQLException ex) {
            throw new FlowRuntimeException("Could not execute update.
Nested Exception: " + ex);
        }
        return null;
    }
}
```

Appendix

C

XQuery Keywords

Because Knowledge Broker uses XQuery to search its datasources, some XQuery-specific keywords are reserved, and should not be used as user-defined names for ontology objects.

XQuery Reserved Keywords

These are the XQuery Reserved Keywords:

Table 0-1. XQuery Reserved Keywords

After	And	As	Ascending
Attribute	Before	Case	Comment
Default	DefineFunction	Descending	Div
Element	Else	Equals	Every
Except	Follows	For	Function
Group	If	In	Instanceof
Intersect	Item	Let	Mod
Node	Only	Or	Precedes
Return	Returns	Satisfies	Schema
Some	Sortby	Stable	Text
Then	To	Union	Where

The canonical list of "TERMINAL" XQuery words can be found here:

```
http://www.w3.org/TR/xquery/#id-terminals-table
```

Appendix

D

Writing XQueries

Knowledge Broker uses XQuery extensively to search and traverse complex data sources. Although the user interface (UI) wizards can auto-generate XQuery, advanced users must often input dedicated, custom XQuery. This appendix provides an introduction to writing XQueries for Knowledge Broker users.

Using Connections and Schemas

XQuery is a query definition and extraction language for XML documents that uses XPath for addressing. For more information, go here:

```
http://www.w3.org/TR/xpath
http://www.w3.org/XML/Query
```

All XQueries must use a connection or interaction function. This ensures they have a schema over which to execute.

Simple and Complex Type Elements

XML complex type elements can be empty, can only contain other elements, can contain only text, or can contain other elements and text; every element can itself also contain attributes.

This is a complex type element that contains other elements:

```
<areacode>
    <city>SF</city>
    <phone>415</phone>
</areacode>
```

This is a complex type element that contains both elements and text:

```
<mail>
Dear Ms.<recipient>Jane Lee</recipient>.
You will receive your purchase <purchaseid>3021</purchaseid>
on <duedate>2002-04-23</duedate>.
</mail>
```

This is declared using XSD as:

```
     <xs:element name="mail">
         <xs:complexType mixed="true">
             <xs:sequence>
                 <xs:element name="recipient" type="xs:string"/>
                 <xs:element name="purchaseid" type="xs:positiveInteger"/>
                 <xs:element name="duedate" type="xs:date"/>
             </xs:sequence>
         </xs:complexType>

</xs:element>
```

Using Source Schemas

The basic rule is that the result of an XQuery on a concept must agree with the schema associated with that concept.

To illustrate this, assume you have defined a `fileCustConnection` connection to an XML file that contains customer data. The connection references a schema with a structure that reflects the file's contents. Schemas used in connections are *source schemas*. The schema will look something like this:

```
<schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<element name="Customers">
<complexType>
    <sequence>
        <element name="Customers_Instance" type="CustomersType"/>
    </sequence>
</complexType>
</element>
<complexType name="CustomersType">
    <sequence>
        <element name="ID" type="string"/>
        <element name="firstName" type="string"/>
        <element name="lastName" type="string"/>
        <element name="balance" type="decimal"/>
    </sequence>
</complexType>
</schema>
```

The `Customers_Instance` element is an instance of a Customer.

Defining Concepts

During concept definition, you have two choices:

1 Define a schema for the concept.

These are user-defined business concepts (UBCs). Schemas created this way are restrictive because they can never contain complex type elements.

2 Associate the concept with a top-level element from a pre-defined schema.

These are structured business concepts (SBCs). Pre-defined schemas can be generated using XML Schema editors, provided by third parties, or using Knowledge Broker (see).

Derived concepts can be based on either UBCs or SBCs.

Schemas associated with business concepts are *target schemas*. These can be independent from any physical data source. This is an example of a target schema:

```
<schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<element name="People">
<complexType>
    <sequence>
        <element name="Person" type="PersonType">
        </element>
    </sequence>
</complexType>
</element>
<complexType name="PersonType">
    <sequence>
        <element name="SSN" type="string"/>
        <element name="Name" type="NameType"/>
    </sequence>
</complexType>
<complexType name="NameType">
    <sequence>
        <element name="forename" type="string"/>
        <element name="initial" type="string"/>
        <element name="surname" type="string"/>
    </sequence>
</complexType>
</schema>
```

Extracting Information Using XQuery

You want to generate a query that maps customers to people. That is, you want to enter an XQuery that extracts customer information from the XML file and present the results in a document that matches the target schema. Your XQuery should look like this:

```
<People> {
    FOR $i IN connection("fileCustConnection")/Customers/
Customers_Instance
    RETURN
    <Person>
        <SSN>{$i/ID/text()}</SSN>
        <name>
            <forename>{$i/firstName/text()}</forename>
            <surname>{$i/lastName/text()}</surname>
        <name>
    <Person>
}</People>
```

The connection function is a Knowledge Broker-specific extension to XQuery. For current versions of Knowledge Broker, it cannot reference a database connection (because the specified data source returns the complete data, unfiltered).

The query structure relates to the target schema: the top-level schema element (People) encloses the entire query. The FOR clause provides data to the RETURN clause. Beginning with the <People> tag, the RETURN clause's structure continues to build the result document; <Person> is the first element returned. The returned data structure matches the target schema.

The FOR clause produces *n* Customers_Instance elements, and the RETURN clause executes for every individual instance. Therefore, every Customers_Instance in the XML file produces a <Person> element in the result file. From the target schema, every person contains an SSN and Name element. The RETURN clause also contains these.

The instance document produced will look like this:

```
<People>
    <Person>
        <SSN>123456</SSN>
        <name>
            <forename>Joe</forename>
            <surname>English</surname>
        </name>
    </Person>
    <Person>
        <SSN>234567</SSN>
        <name>
            <forename>Eddie</forename>
            <surname>McGreal</surname>
        </name>
    </Person>

        ...

    <Person>
        <SSN>654321</SSN>
        <name>
            <forename>Cormac</forename>
            <surname>Keogh</surname>
        </name>
    </Person>
</People>
```

Selecting a Projection

Every reference in the RETURN clause to a source data element must slot into the document selected by the FOR clause.

This is a variation on the earlier examples:

```
<People> {
   FOR $i IN connection("fileCustConnection")/Customers
      RETURN
      <Person>
    <SSN>{$i/Customers_Instance/ID/text()}</SSN>
    <name>
        <forename>{$i/Customers_Instance/firstName/text()}</forename>
        <surname>{$i/Customers_Instance/lastName/text()}</surname>
    <name>
      <Person>
}</People>
```

The `Customers_Instance` element is absent from the `FOR` clause's XPath (after `connection`). This implies that the `FOR` clause now returns a single `Customers` element, instead of a list of `Customers_Instance` elements. But this `Customers` element contains a list of `Customers_Instance` elements.

Consequently, instead of referring directly to a source element, the query must prefix the element with `Customers_Instance`. This relative XPath addressing maintains the one-to-one correspondence between the source elements and the instance elements returned by the `FOR` clause.

The `i` variable referenced in a `FOR` clause is bound to the result nodes produced by the `FOR` clause. These result nodes are roots to individual instance documents, each with the same source schema identified by the `fileCustConnection` connection. The returned node structure is either a single root node or a list of top-level nodes. The exact structure depends on the XPath used after the `IN` keyword in the `FOR` clause.

This XPath binds a single `i` variable to the single `FOR` clause-produced `Customers` node:

```
FOR $i IN connection("fileCustConnection")/Customers
```

This XPath creates a variable binding of an `i` variable to every `FOR` clause-produced `Customers_Instance` node:

```
FOR $i IN connection("fileCustConnection")/Customers/Customer_Instance
```

Therefore, to correctly address the `i` variable in the `RETURN` clause (or the `WHERE` clause), you must use the the appropriate element XPath by noting the node set topology generated by the `FOR` clause.

Using the text() Function

The `text()` function directs a query to evaluate the scalar value of an element and ignore the element itself.

Therefore, if the source `ID` element has a value of `123456`, then binding the `ID` element to the `i` variable using `text()` will return only the value of the `ID` element:

```
<SSN>{$i/ID/text()}</SSN>
```

returns

```
<SSN>123456</SSN>
```

If `text()` is not used, then the ID element tags are returned as well:

```
<SSN>{$i/ID/}</SSN>
```

returns

```
<SSN><ID>123456</ID></SSN>
```

This breaks the example query because `ID` is not a child element of `SSN`.

Writing Heterogeneous XQueries

The earlier example was homogeneous: it used a single data source connection. Heterogeneous queries are XQueries that require simultaneous in-query connections to multiple data sources.

We can expand the earlier homogeneous example by assuming that the order details are stored on a web server in XML files, that a `webOrderConnection` connection refers to these files, and that the order details use this schema:

```
<schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<element name="Orders">
<complexType>
    <sequence>
        <element name="Order" type="OrderType"/>
    </sequence>
</complexType>
</element>
<complexType name="OrderType">
    <sequence>
        <element name="orderNo" type="string"/>
        <element name="custID" type="string"/>
        <element name="orderDetail" type="OrderDetailType"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="OrderDetailType">
    <sequence>
        <element name="productID" type="string"/>
        <element name="qty" type="decimal"/>
    </sequence>
</complexType>
</schema>
```

Each order may contain an unbounded quantity of orderDetail elements.

This is a sample instance for this schema:

```
<Orders>
    <Order>
        <orderNo>1</orderNo>
        <custID>123456</custID>
        <orderDetail>
            <productID>MLP23-87</productID>
            <qty>2</qty>
        </orderDetail>
        <orderDetail>
            <productID>ZAP03-9</productID>
            <qty>1</qty>
        </orderDetail>
    </Order>
    <Order>
        <orderNo>2</orderNo>
        <custID>234567</custID>
        <orderDetail>
            <productID>MLP23-87</productID>
            <qty>7</qty>
        </orderDetail>
    </Order>
        ...
        ...
</Orders>
```

Profiling customers who are making orders requires joining customer data from `fileCustConnection` with order details from `webOrderConnection`. The joined result requires a target schema:

```
<schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<element name="CustomerOrderProfile">
<complexType>
    <sequence>
        <element name="CustomerOrder" type="CustomerOrderType"/>
    </sequence>
</complexType>
</element>
<complexType name="CustomerOrderType">
    <sequence>
        <element name="CustomerOrderNo" type="string"/>
        <element name="CustomerFirstName" type="string"/>
        <element name="CustomerLastName" type="string"/>
    </sequence>
</complexType>
</schema>
```

Creating a heterogeneous query must follow the same rule as a homogeneous query: the query result must agree with the target schema. This heterogeneous query uses more than one connection to retrieve data:

```
<CustomerOrderProfile> {
   FOR $i IN connection("fileCustConnection")/Customers/
Customers_Instance
   FOR $j IN connection("webOrderConnection")/Orders/Order
   WHERE $i/ID = $j/custID
   RETURN
    <CustomerOrder>
        <CustomerOrderNo>{$j/orderNo/text()}</CustomerOrderNo>
        <CustomerFirstName>{$i/firstName/text()}</CustomerFirstName>
        <CustomerLastName>{$i/lastName/text()}</CustomerLastName>
    <CustomerOrder>
}</CustomerOrderProfile>
```

This query filters or *joins* the `FOR` clause-generated result sets using a `WHERE` clause. The `text()` function is not used on the elements specified in the `WHERE` clause.

This is a sample result for this query:

```
<CustomerOrderProfile> {
    <CustomerOrder>
        <CustomerOrderNo>1</CustomerOrderNo>
        <CustomerFirstName>Joe</CustomerFirstName>
        <CustomerLastName>English</CustomerLastName>
    <CustomerOrder>
    <CustomerOrder>
        <CustomerOrderNo>2</CustomerOrderNo>
        <CustomerFirstName>Eddie</CustomerFirstName>
        <CustomerLastName>McGreal</CustomerLastName>
    <CustomerOrder>
        ...
        ...
    <CustomerOrder>
        <CustomerOrderNo>99</CustomerOrderNo>
        <CustomerFirstName>Joe</CustomerFirstName>
        <CustomerLastName>English</CustomerLastName>
    <CustomerOrder>
}</CustomerOrderProfile>
```

The same customer can appear in different orders: the result set displays active customers associated with order numbers.

Understanding Filter Restrictions

There are several key restrictions on how filters can be specified within `WHERE` clauses:

- A literal must appear on the right-hand side of the relational expression. These are the valid relational expressions:

  =   !=   >   >=   <   <=

- Literals must be quoted if the element datatype used on the left-hand side of the relational expression is non-numeric.

- Filters can be logically combined using AND, OR, and NOT.
- Filters can use parentheses to indicate evaluation precedence.

XQuery Peculiarities

Because XQuery is a new and rapidly evolving standard, its syntax and behavior exhibit certain specific peculiarities.

Leading Minuses

The behavior of leading minuses is still undefined in the XQuery specification. It could be treated either as a unary operator or part of a numeric literal. Therefore, the outcome of this statement may not be what you expect.

For example, suppose you have a Balance record that contains only one element. Your application executes this query:

```
<RESULT>
{
    FOR $i IN interaction("GetChecking") /CheckingAccount/
CheckingAccount_Instance
    RETURN
        <RET>
            {$i/Balance[index]}
        </RET>
}
</RESULT>
```

These are the return characteristics of the query for different values of index:

Table 0-1. Concept Comparison Operators

index	Returns
2	Returns nothing. This behavior is as expected.
1	Returns data. This behavior is as expected.
0	Returns nothing. This behavior is as expected.
-1	Returns data. This behavior is not expected.
-2	Returns nothing. This behavior is as expected.

Leaf Elements Not Exposed During Variable Expansion

Variables are not expanded to expose their leaf elements. Thus, including or excluding a specific element requires that the element be visible to the initial query. This has implications for user-defined querys (UDQs) where the elements must be either explicitly exposed or wrapped within an element constructor.

For example, a simple `RETURN $i` is insufficient; instead use either of these two `RETURN` query forms:

```
RETURN
    <a>
        <a1>{$i/a1}
    </a>
```

(explicitly exposed)

or

```
RETURN
    <a>
        {i/a1}
```

(element constructor)

Include/Exclude Paths Applied Only To Concept Queries - No Propagation

Include and exclude paths apply only to concept queries and do not propagate “down” the query during execution.

For example, consider this UDQ:

```
FOR $i IN interaction("GetInterest")
    RETURN $i
```

with the associated Interaction query:

```
<mytable>
    {
    FOR $i IN /mytable/mytable_Instance
    RETURN
        <mytable_Instance>
            <story>{$i/story/text()}</story>
            <interest>{$i/interest/text()}</interest>
        </mytable_Instance>
}
</mytable>
```

In this scenario, the include and exclude functionality will be inoperative. This is because `$i` is not expanded and not propagated.

The include/exclude functionality operates as expected with this slightly altered Interaction query:

```
<mytable>
    {
    FOR $i IN interaction("GetInterest")/mytable/mytable_Instance
    RETURN
        <mytable_Instance>
            <story>{$i/story/text()}</story>
            <interest>{$i/interest/text()}</interest>
        </mytable_Instance>
}
</mytable>
```

Using Interactions

Like the connection function, the interaction function is a Knowledge Broker-specific extension to the XQuery standard. Interactions are high-level “connections” that describe both an XQuery expression and the associated execution connection. Previously, it was impossible to create interactions through the user interface but Knowledge Broker Version 2.1 enables this functionality.

Using XQueries in Interactions

Interactions, like connections, return data using XQueries. With connections, the specified data source returns the complete data, unfiltered. With interactions, the data source returns a set of data defined by the query expression described in the interaction. This filtering ability enables you to use interactions to extract specific subsets of information from databases (the connection function does not work with databases because without the filter ability it would attempt to retrieve all the stored data).

Interaction queries must follow the same rules as connection queries: the returned document must match the target schema and the source element references must slot into the document returned by the `FOR` clause or clauses. Additionally, the returned data is more complex: the outer XQuery author must understand the structure of the data provided by the internal XQuery. That is, the XQuery that calls the interaction must understand the XQuery inside the interaction.

For example, consider a `CustomerProfileInteraction` interaction that returns the `CustomerProfile` document. The XQuery to extract just the customer names and present them in the `People` schema looks like this:

```
<People> {
   FOR $i IN interaction("CustomerProfileInteraction")/CustomerOrderProfile
   RETURN
      <Person>
    <name>
         <forename>{$i/CustomerOrder/CustomerFirstName/text()}</forename>
         <surname>{$i/CustomerOrder/CustomerLastName/text()}</surname>
    <name>
      <Person>
} </People>
```

Note that the `RETURN` clause element references slot in to the document returned by the interaction function.

It's possible to rewrite the query in a simpler form:

```
<CustomerOrderProfile> {
    FOR $i IN interaction("CustomerProfileInteraction")/
CustomerOrderProfile
    RETURN
       $i
}</CustomerOrderProfile>
```

To accomplish this simplification, the query's target schema must be changed to match the target schema of the document returned by the interaction.

Writing XQueries for Interactions

All of the "matching" and "slotting" practices for writing good XQueries also apply to the XQuery expression contained within an interaction, that is, the "inner query." One difference is that the connection used to execute the query is not explicitly defined by a connection or interaction function, but instead implicitly implied by the interaction.

Suppose a database connection points to an Oracle instance that contains product information. The table name is `Product`. A query to return only those products with a price greater than $1,000 is:

```
<Products> {
    FOR $I IN /Product/Product_Instance
    WHERE $i/value > 1000
    RETURN
       <Expensive_Product>
        <productName>$i/name</productName>
        <productPrice>$i/cost</productPrice>
       </Expensive_Product>
} </Products>
```

Because this version of Knowledge Broker maintains a direct link between business concepts and their underlying XQuery mapping, the data retrieval process is exposed as a direct flow. This means that interactions should only be used as part of a business concept's outer query (except with the Query command, where interactions can be directly used).

Writing a Recursive Interaction

Writing an Interaction Using the Same Connection Function

Passing Variables Into XQueries

Using SQL in XQueries

Understanding Interaction Descriptors

Knowledge Broker functions by passing descriptors between its components. These descriptors are XML documents that describe the state of system variables. Descriptors are chained together using unique names that identify connection descriptors and interaction descriptors. Knowledge Broker triggers descriptors by creating "flow" documents, that is, XML documents of sequential, scripted actions.

Using Interaction Descriptors

Many Knowledge Broker operations create flows that encode their operation, using interaction descriptors and embedded XQuery expressions.

For example, an RDBMS Insert operation creates this flow:

```
<process name="Get Test">
    <flow>
        <get businessConcept="Client"/>
       <publish businessConcepts="Client" to="rdbmsTable" using="saveIxn"/>
    </flow>
    <schedule/>
</process>
```

This flow references this interaction descriptor using the `saveIxn` name:

```
<interactionDescriptor>
    <name>saveIxn</name>
    <connectionName>testDB</connectionName>
    <functionName>Insert</functionName>
    <expression/>
    <interactionVerb>SYNC_SEND</interactionVerb>
</interactionDescriptor>
```

Other interaction descriptors use a similar format.

This is a TIBRV interaction descriptor for a TIBRV connection:

```
<interactionDescriptor>
    <name>kbSysPutTibrvMessage</name>
    <system>false</system>
    <connectionName>kbSysTibrvCXN</connectionName>
    <functionName>Send</functionName>
    <expression/>
    <interactionVerb>SYNC_SEND</interactionVerb>
    <destination>blahSubject</destination>
</interactionDescriptor>
```

This is a JMS interaction descriptor for a JMS connection:

```
<interactionDescriptor>
    <destination>weblogic.examples.jms.exampleQueue</destination>
    <name>kbSysPutJMSMessage</name>
    <system>false</system>
    <connectionName>kbSysJMSCXN</connectionName>
    <functionName>Put</functionName>
    <interactionVerb>ASYNC_SEND</interactionVerb>
</interactionDescriptor>
```

Glossary

Action. An event that effects a change (*See also* Verb).

Action Knowledge. The ability to act on plans.

Agent. An intelligent, autonomous, and mobile entity capable of acting on behalf of a human, another agent, or a software program. *Intelligent* means that the agent is capable of perceiving its environment, learning, collaborating, communicating, adapting, and making decisions. *Autonomous* means that the agent acts in accordance with a set of programmed or learned functions that it attempts to optimize. *Mobile* means that the agent is capable of associating or de-associating with other agents in an attempt to optimize its actions.

API. Application Program Interface. A means by which two independently-written software programs can interact with one another to perform additional functions.

Application. Ties together agents and can output Knowledge Broker recommendations to a variety of external systems and feeds.

Association of ideas. The ability to unite in some manner two or more concepts, based on resemblance and/or contiguity.

Attribute. The character or property of an object.

Business Concept. A vocabulary term that is relevant to your industry or company. You define Business Concepts in the ontology's Business Concept Editor (*See also* Concept).

Common-Sense Rules. Rules that include opinions, hunches, and best practices, and that can be broken if circumstances warrant.

Client. A Client program requests data and services from a Server (*qv*), performs some processing, and presents the output to the Client user. Knowledge Broker can be installed as a Client version that requires software connection to a Knowledge Broker Server to function.

Compositionality. The ability to infer knowledge from the meanings, patterns, and relationships of individual concepts or information, based on a system of rules.

Concept. An abstract image about the who, what, when, and where of data and data relationships. Within Knowledge Broker, concepts are the vocabulary terms used for defining rules, querying disparate datasources, analyzing information patterns, recognizing implications, and making actionable recommendations.

Contiguity. Concepts frequently experienced together are mentally associated together.

Data. A specific person, place, thing, or event object.

Data Concept. A vocabulary term that maps directly to one or more datasource objects.

Dataset. The collection of datasource Instances.

Data Source. The disparate Web, relational database, and legacy files containing data pertinent to your business.

Data Source Connection. The URLs, strings, or pathnames to your Web, relational database, and legacy files.

Data Element. A word, number, measurement, or fact that represents or refers to an object.

Domain-Specific Rule. Rule that is defined by a specific business or field of knowledge (including physics, chemistry, and law), cannot be broken unless amended by the issuing domain.

Dynamic Knowledge. Makes use of existing data elements, concepts, contexts, information, and knowledge to create addition knowledge.

Editor. A User Interface (UI) that enables a person to communicate with Knowledge Broker to select options, set parameters, and save, modify, or delete information.

Explicit Knowledge. A set of codified experience and know-how that is communicated through formal language, including data elements, concepts, contexts, and information (*See also* Tacit Knowledge).

Filter. A device specifying the parameters that Knowledge Broker will use to evaluate, process, and sort data.

Graphical User Interface (GUI). A program that enables users to log in to a Knowledge Broker Server to begin sessions and reactive and edit knowledge bases.

Implication Knowledge. The recognition of the implications of those patterns on future behaviors.

Implicit Knowledge. A set of personal experiences know-how, and mental images that is rarely (if ever) communicated with others.

Information. The association of concepts and contexts whose meanings are understood.

Instance. A specific record associated with a Data or Business Concept. For example, every person identified as a customer is an Instance of the Customer Data Concept, while every person identified as a male customer is an Instance of the Male Business Concept.

Intelligent Agent. An agent is considered to be intelligent if it is capable of perceiving its environment, learning, collaborating, communicating, adapting, and making decisions.

Iterative. An iterative process occurs when the process repeats, taking its result output and feeding it back into itself to arrive at a progressively more refined or detailed result.

JDBC. A Java computer language-based technology that Knowledge Broker uses to gain access to information stored in tabular-format databases (*See also* ODBC).

Knowledge. A group of four interrelated things: the understanding of patterns of information that consistently and completely repeat themselves over time; the understanding of the implications of those patterns on future behavior; the ability to make plans and recommendations based on those implications; and the ability to act on those plans.

Knowledge Base. A collection of datasource connections, ontology concepts and concepts relations, rules, and predictive models related to a particular industry or business domain.

Knowledge Discovery. A Knowledge Broker process that analyzes historical data to determine patterns, relationships, and trends that can be used to predict future values and real-time recommendations.

Learning. Knowledge Broker has an ability to adapt or learn from experience. The learning agents adapt to a customers likes and dislikes, which agents to cooperate with and trust and which ones to avoid. Learning is based on Knowledge Broker's ability to recognize situations it has seen before and improve its performance based on prior experience.

Legal Rules. Rules that are mandated by a federal, state, or city, or other warrantable agency (including SEC, OSHA, FCC, EPA, and unions) cannot be broken unless the issuing agency amends the rule.

Map. The link between a Datasource Connection and a Data Concept.

Mean. The representative average for the entire dataset.

Metadata. The description of the location, format, and meaning of data stored in your disparate datasources.

Model. An abstract conception or representation of a portion of reality.

Multi-layer Perceptron (MLP). A predictive algorithm.

ODBC. A Microsoft Windows-centric technology that Knowledge Broker uses to gain access to information stored in tabular-format databases (*See also* JDBC). Ontology

Ontology. A collection of concepts and contexts used to provide a common vocabulary for defining rules, querying disparate datasources, analyzing information patterns, recognizing implications, and making actionable recommendations.

Pattern Knowledge. The process of understanding patterns of information that consistently and completely repeat themselves over time.

Persistence. An object that stays in memory and can fulfill multiple requests. Knowledge Broker supports the persistence of concept profiles and relationships.

Policy. A business practice or rule.

Population. Data and Business Concepts used as the antecedent portion in a rule.

Predictive Model. A data-mining process whereby a mathematical function is used to analyze your data to discover metadata (*qv*) and make predictions about actions, behavior, or requirements. It is a method for "learning by example.".

Property. A characteristic of an object.

Real-time Recommendation. Advice that recognizes the implications of patterns on future behavior.

Reasoning. A set of rules and facts; a working memory of derived data and facts and an inference engine containing reasoning logic used to process rules and data. Knowledge Broker's reasoning engine supports both forward- and backward-chaining to derive and test facts from data.

Recommendation. A suggestion for products, services, or actions.

Resemblance. When two concepts are considered to be similar, whatever properties associated with one concept are automatically associated with the other concept.

Rule. Describes how you expect a person or thing to behave in a particular situation. Rules are the means for "teaching" the computer how to respond to situations relevant to your business requirements, and are the basis for generating actionable recommendations. In Knowledge Broker, rules take a familiar "if-then" format.

Rule Base. A collection of rules.

RuleSet. A collection of rules sharing a common set of verbs.

Server. A Server program provides data and services to a Client (*qv*). Knowledge Broker can be installed as a Server version that supports a number of Knowledge Broker Clients.

Standalone. Knowledge Broker can be installed as a Standalone version that combines both Client and Server within one program and provides limited functionality, compared with the Application Server version.

Strategy Knowledge. The capacity to make plans and recommendations based on those implications.

Tacit Knowledge. A set of personal experiences, know-how, and mental images that is rarely (if ever) communicated with others (*See also* Explicit Knowledge).

Verb. An event that effects a change (See also Action). Knowledge Broker's Inference Engine infers over verbs, using sets of verbs to tie together Rules sharing a common purpose.

An event that effects a change (See also Action). Knowledge Broker's Inference Engine infers over verbs, using

Index