SUN SEEBEYOND

JAVA™ COMPOSITE APPLICATION PLATFORM SUITE PRIMER

Release 5.1.1

Sun microsystems

Version 20060619212437

Contents

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

List of Figures

Chapter 1

Introduction

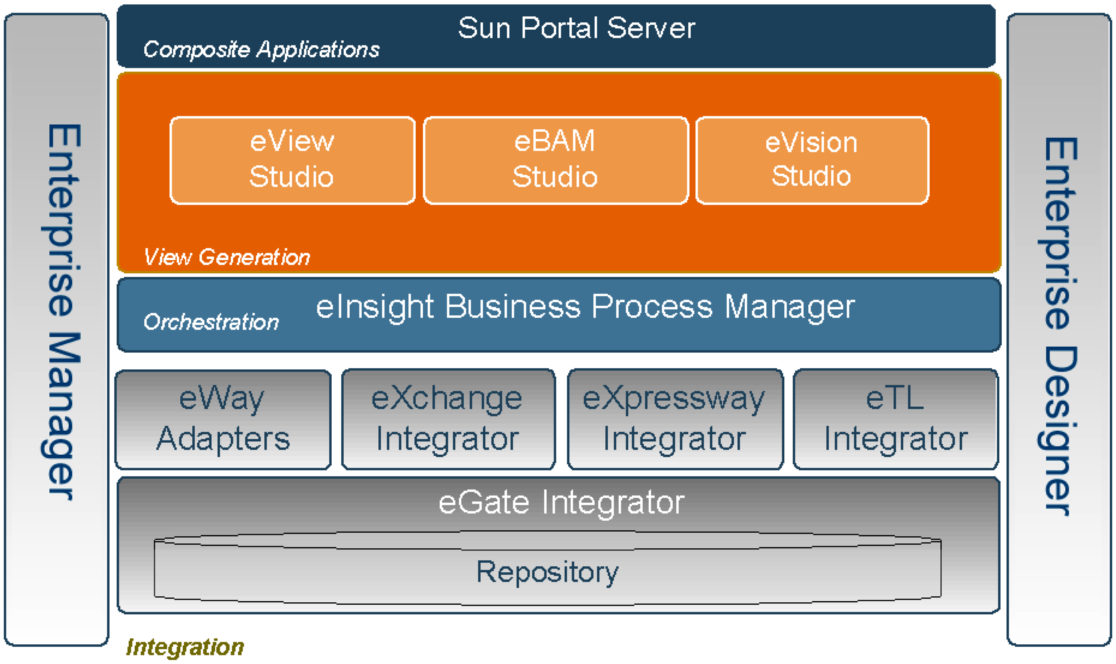
This chapter provides an overview of the Sun Java™ Composite Application Platform Suite (Java CAPS).

What's in This Chapter

1.1 About the Sun Java Composite Application Platform Suite

In today's economic climate, every business must optimize operations. This means that organizations will profit more by making better use of systems already in use, connecting seamlessly with partners, suppliers, and customers. At the same time organizations must adjust to tremendous transaction volumes while managing inventories at peak efficiency.

Figure 1 Sun Java Composite Application Platform Suite



With Java CAPS, you can:

- **Connect** to partners, suppliers, and customer organizations rapidly and securely to enable the exchange of business critical information. Quickly establish trading partner relationships and dynamically manage those relationships.
- **Integrate** all critical business data, systems, and disparate applications among multiple independent business units into a common platform to deliver real-time information to partners, suppliers, and customers.
- **Optimize** core internal and external business processes to create an agile organization able to quickly react to changing business conditions in real-time.

Java CAPS represents the most comprehensive, unified eBusiness infrastructure available for companies seeking to improve business operations and drive rapid return on investment. It delivers seamless application integration, dynamic business-to-business connectivity, and robust business process management capabilities across the extended enterprise for customers, suppliers, and partners.

1.2 What's in This Document

This document overviews products in Java CAPS. It includes the following chapters:

- Chapter 1 **"Introduction"** provides an overview of this document.
- Chapter 2 **"eGate Integrator"** describes the architecture, components, and functionality of the Sun SeeBeyond eGate Integrator application.

- **Chapter 3** **"Sun Java System Portal Server"** This chapter describes the Sun Java System Portal Server and its uses with Java CAPS.
- **Chapter 4** **"eInsight Business Process Manager"** This chapter describes Sun SeeBeyond eInsight Business Process Manager (eInsight) and how it interacts with Java CAPS.
- **Chapter 5** **"Sun SeeBeyond eVision Studio"** describes the features of the eVision graphical design studio, a graphical design studio for the WYSIWYG creation of integrated Web applications.
- **Chapter 6** **"eView Studio"** describes the features of the eView Studio. eView Studio is an application building tool that enables you to design, configure, and create a master index that will uniquely identify and cross-reference the business objects stored in your system databases.
- **Chapter 7** **"eIndex Single Patient View"** describes the features of eIndex Global Identifier Composite Application (eIndex). eIndex is a healthcare-oriented, enterprise-wide master person index that maintains the most current information about the patients in your organization.
- **Chapter 8** **"eTL Integrator"** describes the features of the eTL Integrator. eTL Integrator provides the functionality to *extract*, *transform*, and *load* data between databases.
- **Chapter 9** **"eBAM Studio"** describes the features of eBAM. eBAM provides *business activity monitoring* functionality for collection, aggregation, and presentation of business activity data according to specified key performance indicators. eBAM Studio provides the tools for generating custom digital dashboards for defining and monitoring these key performance indicators.

The **Glossary** on page 62 lists eGate Integrator terms and their definitions.

1.3 Sun Microsystems, Inc. Web Site

The Sun Microsystems web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

http://www.sun.com

1.4 Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

CAPS_docsfeedback@sun.com

Chapter 2

eGate Integrator

The Java Composite Application Platform Suite (CAPS) is built around Sun SeeBeyond eGate™ Integrator, the most widely deployed Enterprise Application Integration (EAI) platform on the market. eGate Integrator quickly solves complex connectivity issues and enables the dynamic delivery of information across applications and systems to partners and customers with unmatched performance, flexibility, and ease of use.

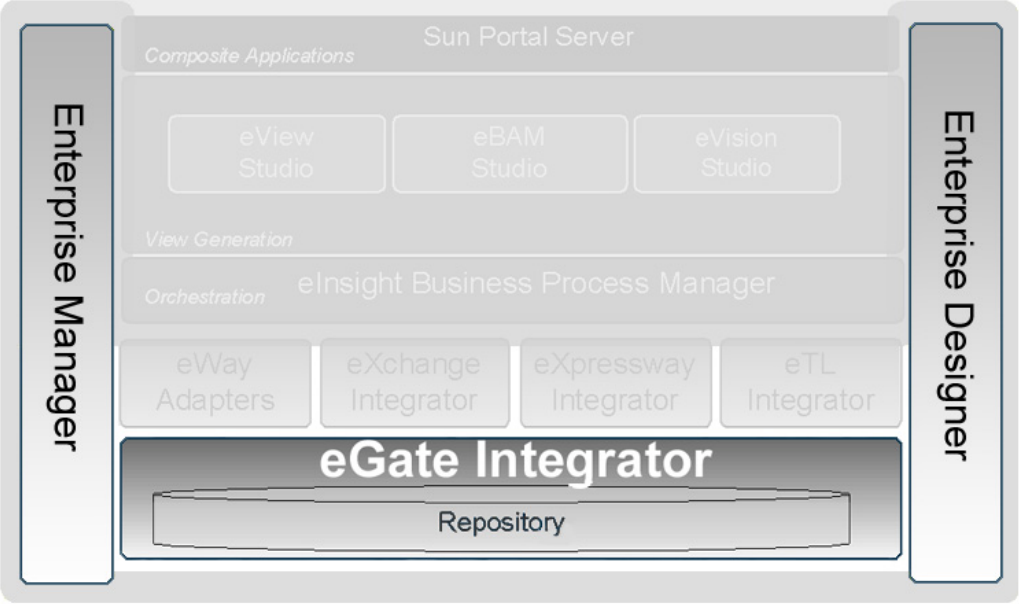
What's in This Chapter

2.1 Introduction

eGate Integrator is a fully J2EE-certified, web services-based integration platform that serves as the foundation of the Java Composite Application Platform Suite. It provides the core integration function, incorporating comprehensive systems connectivity, guaranteed messaging, and robust transformation capabilities. eGate Integrator also provides a unified, single sign-on environment for integration development, deployment, monitoring and management.

As shown in the following figure, the heart of eGate Integrator is the Repository, which is a comprehensive store of information common to the entire Java Composite Application Platform Suite. A separate UDDI registry allows publication and discovery of web services.

Figure 2 eGate Integrator



The run-time environment employs J2EE-compatible integration servers as operational engines and JMS-compatible message servers for the propagation of messages. The flexibility of the eGate Integrator system allows the option of deployment to a Sun SeeBeyond run-time environment or to third-party application servers, across a distributed network of hardware platforms.

Enterprise Designer provides a unified, graphical development environment for integrating systems and developing composite applications using web services. Enterprise Manager provides a unified, web browser-based framework for managing all aspects of the run-time environment.

eGate Integrator can communicate with and link multiple applications and databases across a variety of different operating systems. eGate Integrator performs with a wide variety of hardware, message standards, operating systems, databases, and communication protocols in both real-time and batch (scheduled) integration modes.

2.2 Integration Model

The Java Composite Application Platform Suite addresses application integration by means of an integration Project, which contains the business logic required to solve the specific problem. The Project contains the various logical components and supporting information required to perform the routing, processing, and caching of messages containing the relevant data from one application to another. All Project information is stored in the Repository.

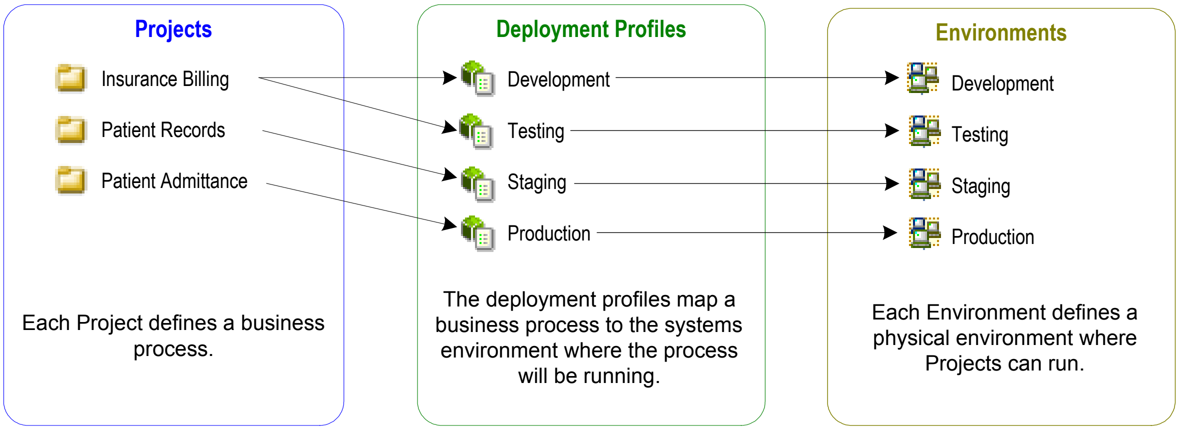
Projects are created using tools contained within Enterprise Designer and, once deployed, can be run and monitored using Enterprise Manager. Projects can also be set up to be run from the business process level using the Sun SeeBeyond eInsight Business Process Manager, if that product is also installed.

Projects are run within individual sets of system definitions, referred to as Logical Hosts. These are defined within Environments, which represent the physical resources required to implement the Project. Projects are mapped to the individual Environments by means of deployment profiles, which are defined within Enterprise Designer and become part of the Project. Activating the deployment profile deploys the Project to the associated Environment.

This structure of Projects, Environments, and deployment profiles isolates each implementation into logical and physical realms. This provides you with extensive flexibility and efficiency in implementing integration Projects. For example, once you build your Projects and Environments, you have the flexibility to change each realm without having to make changes to the other realm.

The finished Project, of course, will run in your production Environment; separate Environments, having the same structure as the production Environment, should be created for development and testing. You may also want some additional Environments, such as staging. The following figure illustrates the eGate Integrator implementation model using a healthcare-related example.

Figure 3 eGate Integrator Implementation Model


Projects
Insurance Billing
Patient Records
Patient Admittance
Each Project defines a business process.
Deployment Profiles
Development
Testing
Staging
Production
The deployment profiles map a business process to the systems environment where the process will be running.
Environments
Development
Testing
Staging
Production
Each Environment defines a physical environment where Projects can run.

Any of the Projects shown in the figure can be deployed to any of the Environments via the mapping defined in the deployment profiles. The example in the figure above shows that the patient admittance Project is already in the production phase and therefore was deployed using the production deployment profile. The patient records Project is in the staging phase and was therefore deployed to the staging Environment using the staging deployment profile. The insurance billing Project is still being developed and tested, and therefore it is deployed to development and testing via the development and testing profiles.

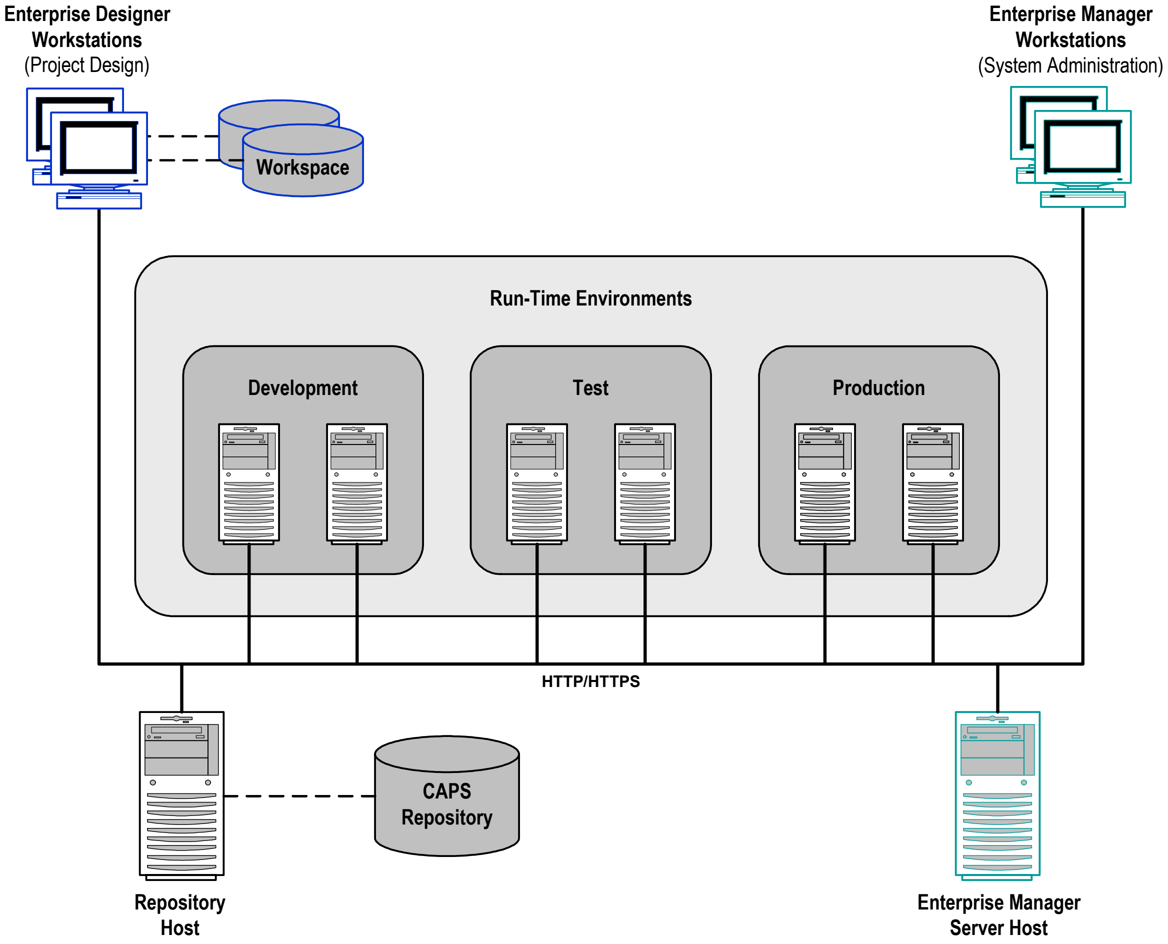
Typically, a particular Project will exist in different versions—a mature version in the production environment, and a follow-on version in the development environment, for example. Isolating these different versions from each other is handled by maintaining separate branches of the Repository.

2.3 System Architecture

eGate Integrator employs a versatile architecture that is ideally suited to distributed computing environments. As a result, the various components of an eGate Integrator system can reside on the same hardware platform (assuming adequate system resources), or be distributed across several different hardware platforms in the enterprise network.

Figure 4 shows an example of a distributed system implementation having six separate run-time servers in addition to the Repository server and, optionally, an Enterprise Manager server. Each run-time server can host multiple domains, each consisting of an application server-message server pair.

Figure 4 Distributed eGate Integrator System


Enterprise Designer Workstations (Project Design)
Workspace
Enterprise Manager Workstations (System Administration)
Run-Time Environments
Development
Test
Production
HTTP/HTTPS
CAPS Repository
Repository Host
Enterprise Manager Server Host

Note: *This scenario assumes that all instances of eGate Integrator are of the same release.*

2.3.1 Repository

The setup, components, and configuration information for the elements of a Project are stored in the Repository. As shown in Figure 4, a single Repository serves the entire enterprise—this common Repository is used for development, testing, and production purposes. Communication between the Repository and other Java CAPS components can be configured to use either HTTP or HTTPS. The Enterprise Designer and Enterprise Manager clients can communicate with the Repository through a firewall.

2.3.2 Run-Time Environments

An Environment represents the total system required to implement a Project. It consists of a collection of Logical Hosts, capable of hosting components of Java CAPS, along with information about external systems involved in the implementation.

- **Logical Hosts**
  Each Environment contains one or more system definitions. Each definition must include one or more **application servers** such as the Sun SeeBeyond Integration Server, which are the engines that run Collaborations and eWays, and one or more **message servers** such as the Sun SeeBeyond JMS IQ Manager, which manage JMS topics (publish-and-subscribe messaging) and queues (point-to-point messaging). Each collection of integration servers and message servers, plus additional software modules, comprise what is known as a *Logical Host*.
- **External Systems**
  An external system is a representation of a real, physical system that exists within the specific Environment, with configuration properties for locating and accessing that system.

In the example system shown in Figure 4, the production environment is split across two hardware platforms. Separate environments for development and testing should duplicate the structure of the production environment. The test environment should be supported by hardware similar to that supporting the production environment, to allow performance and load testing to give representative throughput results. The hardware supporting the development environment, however, does not usually have the same performance requirements as that supporting the test and production environments.

An integration Project is created within the development environment, then migrated to the test environment, and finally to the production environment. This migration path is a necessary and highly critical practice in implementing a working system.

Note again that there is no requirement for the components shown in Figure 4 to run on separate systems; all could run on a single system, provided that resources (processing, memory and storage) are sufficient to support concurrent usage.

2.4 Projects

A Project represents the logical system designed to implement either all or part of a business integration scenario. Projects are created using tools contained within the Enterprise Designer such as the Connectivity Map Editor, shown in **Figure 5 on page 18**.

The Connectivity Map contains information about how components are connected. Connectivity Maps include one or more services (such as Collaborations), queues, topics, external applications, and eWays. An eWay is a component that connects eGate Integrator to an external business application. These components implement high-performance, distributed exchange of data with external systems.

2.4.1 Project Components

External Application Proxies

The basic purpose of eGate Integrator is to facilitate the interchange of data between external business applications. These business applications are collectively referred to as *external applications*, and are represented in the Project by logical proxies for the specific applications involved. An external application can be identified with an ERP application such as SAP or PeopleSoft, a DBMS such as Oracle or SQL, or with a particular communications protocol, such as TCP/IP or HTTPS.

Collaborations

A *Collaboration* is a logical operation taking place between a message destination and an external application, two message destinations, or two external applications, having a publication and subscription relationship with those entities. The operation is defined by a Collaboration Definition, which can be encoded in either Java or XSLT.

Collaboration Definitions

A Collaboration Definition defines the logical operation taking place in the related Collaboration. It is created in either the Java Configuration Editor or the XSLT Configuration Editor, and is based on an Object Type Definition.

Object Type Definitions

Object Type Definitions (OTDs) are a set of rules that define the encoding of an object. They describe:

- Messages that are propagated through eGate Integrator, and the methods available for operating on them.
- Interactions with external APIs.

OTDs can function as instances in eGate Collaboration Definitions, or as services in eInsight Business Process Manager. They are created using the OTD Wizard and customized in the OTD Editor.

Message Destinations

A Message Destination is a container for stored data, and can follow either the topic or queue JMS model.

- A topic is a message destination that conforms to the publish-and-subscribe messaging paradigm.
- A queue is a message destination that conforms to the point-to-point messaging paradigm.

Logical Connections

An eWay Intelligent Adapter provides the logical connection between external systems (applications or protocols) and a service. The identity of the External Application placed in a Connectivity Map dictates the type of eWay adapter assigned to the link between it and a Service such as a Collaboration, and can be accessible via multiple transport mechanisms.

The JMS Client Connection configuration specifies the connection properties for the linked service (for example, publisher or subscriber).

2.5 User Interfaces

eGate Integrator provides two main graphical user interfaces (GUIs), which generally address different sets of users.

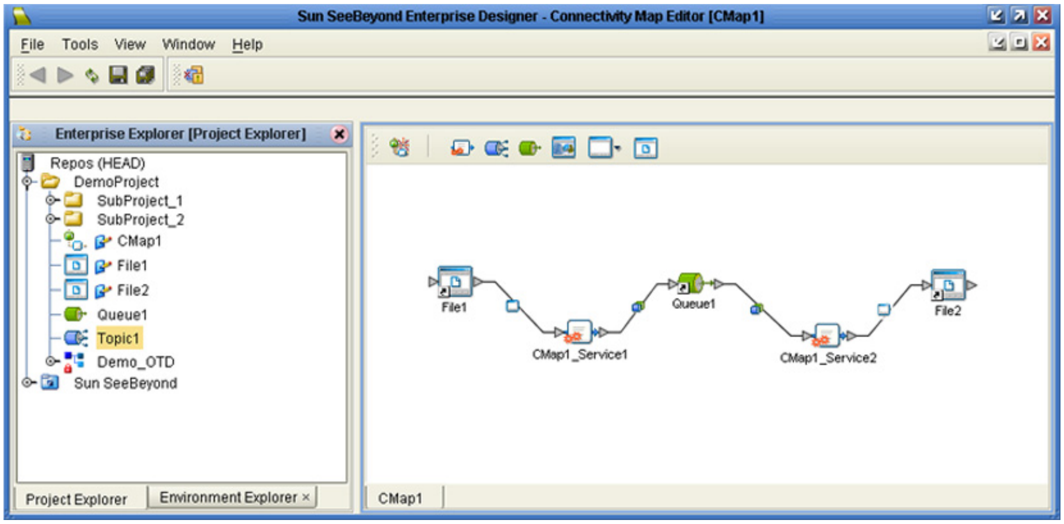
Enterprise Designer is used by personnel who are involved in designing and configuring Projects for integrating the various enterprise applications, using eGate Integrator and other Java CAPS products.

Enterprise Manager is a web browser-based interface used by system administrators.

2.5.1 Enterprise Designer

Enterprise Designer is used to create and configure the logical components and physical resources of an integration Project. Through this GUI (see Figure 5), you can develop Projects to process and route data between external applications or systems.

Figure 5 Enterprise Designer



The major features of Enterprise Designer are Enterprise Explorer on the left, and an editor panel on the right—which is initially blank. Enterprise Explorer follows the familiar Windows Explorer format, displaying a tree structure. The editor panel displays a variety of editors, depending upon what component is selected in Enterprise Explorer.

The Connectivity Map Editor shown in Figure 5 provides a graphic example of one of these editors, in which logical components of a Project can be created and connected. eGate Integrator uses Connectivity Maps to intuitively configure the end-to-end flow of messages within an integration scenario. You can drag and drop the various Collaborations, Intelligent Queues, and eWay adapters onto the Connectivity Map canvas, and link them together to specify message flow.

Enterprise Designer also includes the design-time functionality for other Java CAPS products, such as eInsight and eXchange. For more information on using these products with Enterprise Designer, see the product documentation for those products.

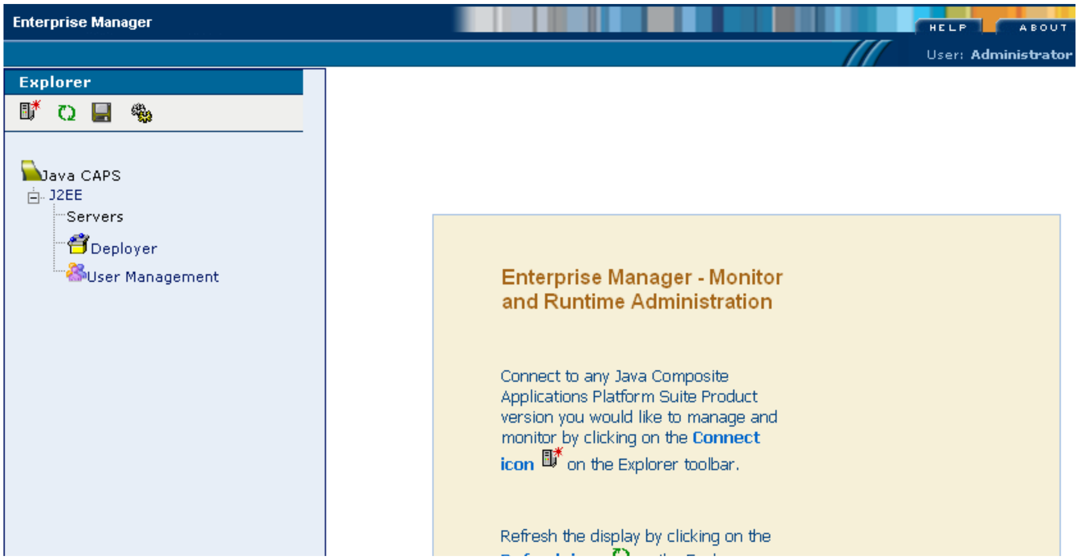
2.5.2 Enterprise Manager

Enterprise Manager is a Web-based interface with which you can manage running Java CAPS applications for both the Schema Runtime Environment (SRE) and the Java 2 Enterprise Edition (J2EE).

Enterprise Manager contains an Explorer panel on the left and a Details panel on the right.

Figure 6 shows the home page of Enterprise Manager.

Figure 6 Enterprise Manager



Enterprise Manager - Monitor and Runtime Administration

Connect to any Java Composite Applications Platform Suite Product version you would like to manage and monitor by clicking on the **Connect icon** on the Explorer toolbar.

Refresh the display by clicking on the

Chapter 3

Sun Java System Portal Server

This chapter describes the Sun Java™ System Portal Server and its use with Java Composite Application Platform Suite.

What's in This Chapter

3.1 Overview of Portal Server

Portal Server delivers capabilities and components necessary for advanced portal solutions. Using Portal Server, you are able to find relevant information and personalize your portal environments to best meet your needs.

Portal Server Basic Benefits

Portal Server's basic benefits include:

Portal Server's major benefits include:

- Aggregation
- Personalization
- Security, including single sign-on (SSO)
- Collaboration and Communities
- Secure remote access
- Mobile access
- Search
- Content management and delivery
- You can refer to Sun's Portal Server Web site for information on additional available features at the following URL:

http://www.sun.com/software/products/portal_srvr/index.xml

These benefits and their related features securely deliver the Portal Server capabilities that you require. The overall result is that you can work more efficiently and flexibly, doing a variety of business-related activities, for example, allowing:

- Employees to collaborate
- Sales personnel to access and update customer information on the road
- Partners to get advance access to the latest product designs

See **Portal Server Features and Benefits** on page 22 for a more detailed list of Portal Server's major features and benefits.

Portal Server and the Java CAPS

You can integrate Portal Server with the Java CAPS, which supports the following Portal Server releases:

- Version 6 2005Q4
- Version 7

Note: *Portal Server is part of the Sun Java Enterprise System (Java ES). Use Portal Server Version 6 2005Q4 when deploying on a Windows platform. Use Version 7 for all other platforms.*

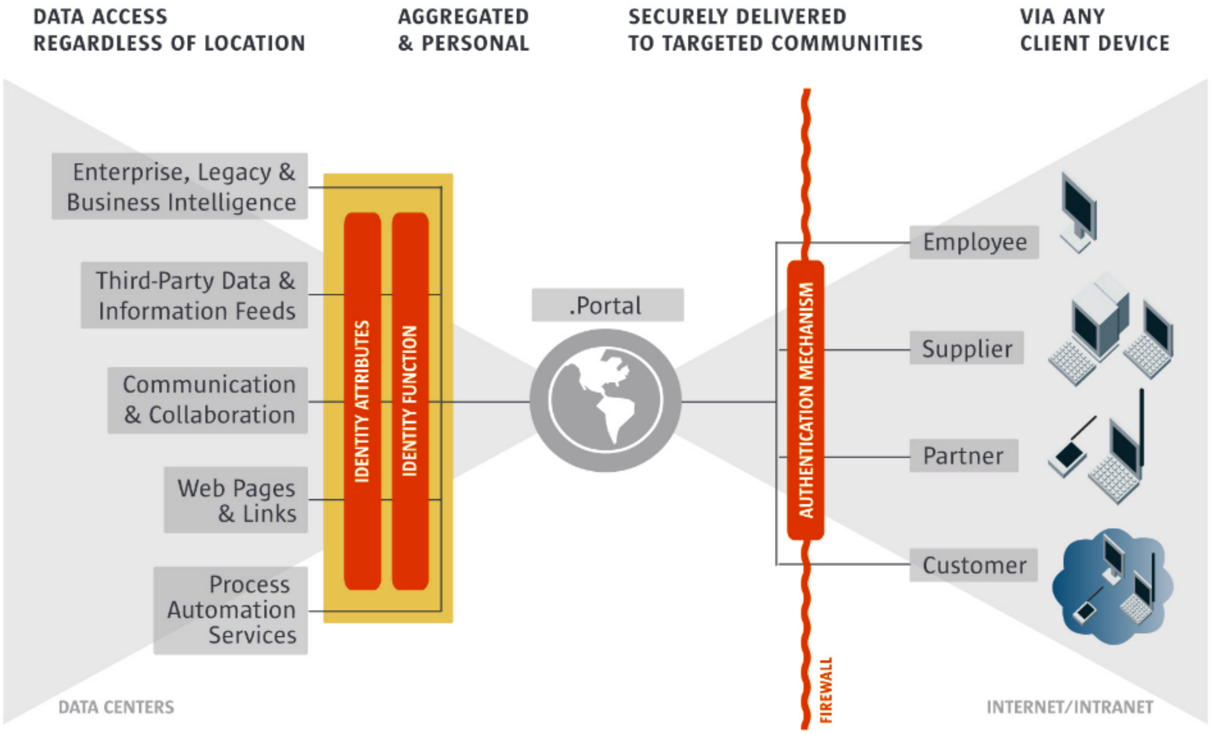
Portal Server supports the following Java CAPS applications:

- Sun SeeBeyond eVision Studio
- Sun SeeBeyond eBAM Studio

Note: *For more information on these applications, see* **Using Portal Server With eVision and eBAM** *on page 23, as well as the chapters in this Primer that describe these applications in greater detail.*

Figure 7 shows a diagram of Portal Server's basic features and general operation.

Figure 7 Portal Server Features and Operation


DATA ACCESS REGARDLESS OF LOCATION
AGGREGATED & PERSONAL
SECURELY DELIVERED TO TARGETED COMMUNITIES
VIA ANY CLIENT DEVICE
Enterprise, Legacy & Business Intelligence
Third-Party Data & Information Feeds
Communication & Collaboration
Web Pages & Links
Process Automation Services
IDENTITY ATTRIBUTES
IDENTITY FUNCTION
.Portal
AUTHENTICATION MECHANISM
Employee
Supplier
Partner
Customer
DATA CENTERS
FIREWALL
INTERNET/INTRANET

3.2 Portal Server Features and Benefits

Table 1 shows a list of Portal Server's major features and benefits, for both administration- and user-related operations.

Table 1 Portal Server Features and Benefits

Feature	Function	Benefit
Identity-based content delivery	Integrated identity management capabilities deliver information and services according to a user's identity, including access control, policy enforcement, and single sign-on (SSO) support.	Provides an increased control of relevant business information and personalized experience, allowing users to easily perform their tasks.
Wikis	Users can create wikis from within a portlet and edit them using the toolbar.	Increases team productivity.
Shared files	Multiple members in a group can share files.	Increases users' abilities to share and participate.

Table 1 Portal Server Features and Benefits (Continued)

Feature	Function	Benefit
Group calendaring and task management	Users can create community/group calendars, update them, and track the status of project tasks.	Improves effective interactions among stake-holders of a project.
Community surveys and polls	Users can make team decisions with polls and surveys they have created.	Speeds consensus-driven decisions.
Secure remote access and mobile access	Remote access to community resources, for example, e-mail, calendars, address books, conference calling, files, and portlets.	Helps improve user productivity by enabling broad access to information.
Convenient administration	For example, one-click portal creation, remote administration, and delegated administration.	Simplifies administration by helping administrators to focus on the delivery of business value to the enterprise.
Content management	Now includes a full, unrestricted FatWire license for creating, managing, and delivering content.	Reduces the integration costs and time needed to deploy content.
Inter-portlet communication	Portlets share information with each other, allowing them a greater degree of flexibility.	Reduces the deployment time and costs of coming up with integrated portal solutions.
Full-text search engine with federated search and taxonomy capabilities	A single request brings results from multiple search engines, as well as enterprise information repositories and databases.	Accelerates user access to relevant information.
Application design that provides service-oriented architecture (SOA)	Provides a unified service view into new business processes and composite applications that result from SOA and integration projects.	Helps deliver relevant services based on identity; also leverages existing IT resources.

3.3 Using Portal Server With eVision and eBAM

Using Portal Server with Java CAPS allows you to fully leverage its features and benefits with the eVision and eBAM applications.

3.3.1 Portal Server and eVision

eVision is a graphical environment that enables you to rapidly create interfaces for composite applications, without the need for advanced programming abilities. When you perform drag-and-drop tasks using the eVision tools, the underlying code is automatically generated.

You can use these tools to create application user interfaces within your integrated business systems, allowing convenient access to any needed data flowing through these systems. eVision can then expose that data to users via Web-based views. These views are able to give users the means to perform necessary tasks.

Portal Server, by exposing these Web interfaces, allows users capabilities to aggregate the views provided by eVision. The resulting collection of views, for example, on a series of tabbed Web pages on one site, gives users easy access to the business systems they need by way of a single-entry point.

3.3.2 Portal Server and eBAM

eBAM allows you to create business activity monitoring applications that intercept the flow of data through other components of Java CAPS, to produce visual presentations of data analysis based on Key Performance Indicators (KPIs) and alerts. eBAM can generate views for users, which provide high-level information on the status of the business.

Portal Server, by exposing these views via Web interfaces, gives users capabilities to aggregate the views provided by eBAM. The resulting collection of views, for example, on a series of tabbed Web pages on one site, allows users easy access to the high-level information they need by way of a single-entry point.

Chapter 4

eInsight Business Process Manager

This chapter describes Sun SeeBeyond eInsight™ Business Process Manager (eInsight) and how it interacts with Java CAPS.

What's in This Chapter

4.1 Java CAPS Overview

Java CAPS allows companies to assemble largescale applications built on existing systems and infrastructure. Java CAPS is an application-level network that unifies connectivity among people, application systems, and devices in different locations and across organizations.

Business services facilitate the implementation of extended applications. Service-oriented architectures (SOA) clarify design and enable reuse by sharing logic and data among different client systems and users.

4.2 Summary of Features

eInsight provides your business with a powerful assortment of features:

- Maximizes Business Process efficiency by enabling Business Process owners to directly model, monitor, manage, analyze, and optimize Business Processes using an easy-to-use, drag and drop graphical user interface.
- Manages long-lived Business Processes and ensures process integrity, including the ability to compensate for failed processing steps.
- Abstracts the complexities of the technical integration using open standards for the graphical notation of a Business Process, elevating the business logic into the process layer to ensure a flexible, Business Process-driven implementation.

- Automates web services orchestration implementing BPEL4WS to assemble web services into larger composite application processes.
- Automatically provides all the interoperability benefits of web services standards without requiring developers to learn SOAP, WSDL, UDDI, and BPEL4WS.
- Guarantees process integrity and eliminates processing errors by ensuring that every step in the Business Process either completes successfully with full traceability and auditability or is handled by robust workflow and exception handling functionality.
- Accelerates decision making and human involvement through robust workflow support, including support for custom task assignment, user roles, and organizational hierarchies.

4.3 eInsight and Java CAPS

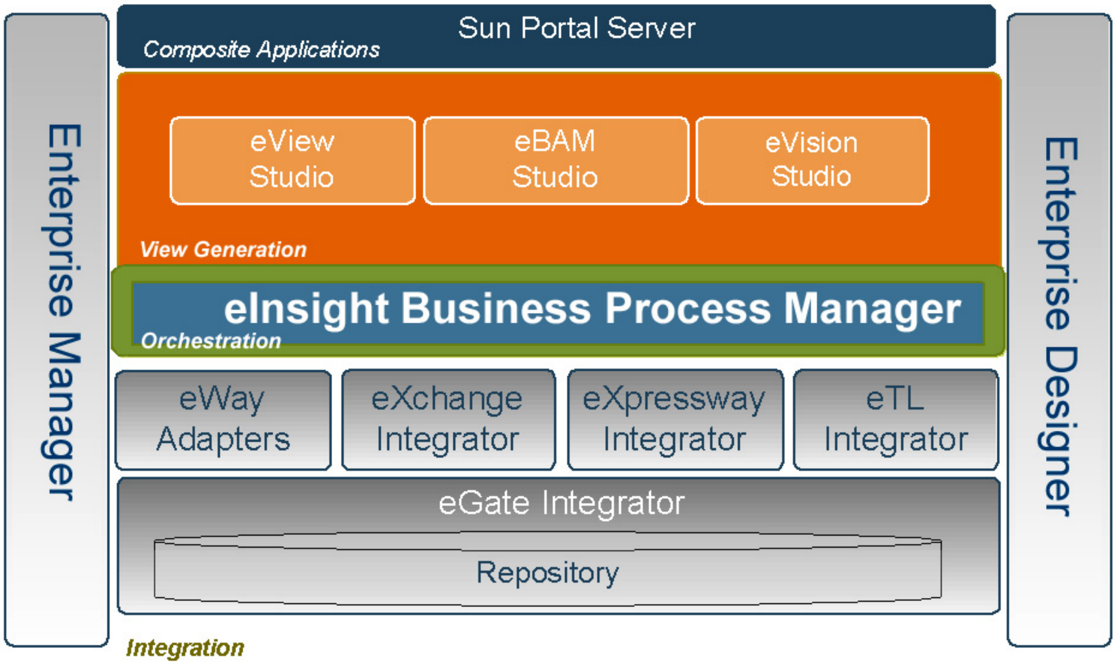
eInsight is a component of Java CAPS. eInsight delivers Business Process management features and functions to Java CAPS.

Business Process management is a strategic orchestration of the movement of information and the flow of complex processes between participants (systems, users, and organizations) to accomplish larger business objectives.

4.3.1 Java CAPS Integration

eInsight is tightly integrated with Java CAPS and runs as a component within the Java CAPS environment. The following figure illustrates some of the eInsight and Java CAPS components that work together.

Figure 8 eInsight and Java CAPS



- The Business Process Designer runs as a component within **Enterprise Designer**.
- Business Process definitions, components, and deployment profiles are stored in the **Java CAPS Repository**.
- The eInsight Engine, which coordinates all Business Process related activity of a deployed project, runs within the **Java CAPS Integration Server.**
- Web-based Business Process monitoring is available throughout **Enterprise Manager**, which provides an interface to access current instance data.

4.3.2 Java CAPS Services

Java CAPS has a wide range of functions that it shares with all Java CAPS products. eInsight can leverage many platform level services, such as:

- **Resource Management** - Java CAPS uses a distributed and open architecture that enables components to access system resources (memory and processing power) as needed, in conjunction with other components.
- **Security** - Java CAPS provides a security module to fulfill security needs such as authentication and authorization access to eInsight functions.
- **Repository storage and access** - The setup, component, and configuration information for the elements of a Project, including Business Process and related components, are stored in the Repository.
- **Deployment abilities** - Java CAPS provides deployment profiles that contain the information necessary to activate eInsight Project Business Processes and associated

components. When a deployment profile is activated, eInsight's active Business Processes are made available as web services.

- **Monitoring** - The Enterprise Manager lends web-based monitoring abilities to eInsight, allowing you to observe and correct Business Process activity.
- **Connectivity Mapping** - The Connectivity Map maintains the relationships between eInsight and other system components. The Connectivity Map specifies the topology of services that will be invoked, by doing the following.
    - Identifies the nature of services that are invoked
    - Depicts relationships between the components, including the publish/ subscribe information for data routing
    - Defines the partners fulfilling the services that are invoked
- **Version Control** - This feature maintains a history of Business Process versions, through a check-in and check-out process.
- **Impact Analysis** - Impact Analysis allows you to view how changes to one component or Business Process will impact other components or Business Processes of a Project or all Projects in the Repository.
- **Import and Export of Business Process Models** - The ability to import and export Business Process models makes it possible to recreate the processes on other systems or to reuse processes that have common components.

4.4 eInsight Architecture

eInsight provides you with a clear view into the internal and external processes of an organization. These processes can be executed by either computer systems or employees.

eInsight speeds the design and deployment of Business Processes by providing an open process modeling environment using Business Process Modeling Notation (BPMN) for the graphical notation of a Business Process. eInsight automatically generates the code needed to implement the Business Process across all participating web services such as applications and business partners.

You drag and drop components into the process model and then specify the additional flow control and business rules that manage what services get called at what time. eInsight supports importing and exporting Business Process Execution Language for Web Services (BPEL4WS) to share processes with third-party tools.

The technologies that carry out eInsight's functions are entirely based on industry standards.

eInsight's architecture uses the following standards.

- Business Process Modeling Notation (**BPMN**), from the Business Process Management Initiative (**BPMI**) standards body, provides a standard graphical view for Business Process Execution Language for Web Services (**BPEL4WS**).

- Web Services Business Process Execution Language (**BPEL4WS**) is the underlying code generated when creating a Business Process.
- Web Services Description Language (**WSDL**) is an XML-based language used to define web services and describe how to access them. All eInsight Business Processes are automatically described using generated WSDL.
- J2EE™ Connector Architecture (**JCA**) provides a mechanism to access external applications and data. The JCA engine is implemented as a standard JCA 1.5 module that plugs into the Java CAPS Integration Server.
- The eInsight Engine uses JCA and Java Enterprise Edition 1.4.

4.5 Process Overview

There are two phases of Business Process Management. The first phase, *design*, is described in the **"Design Phase Overview"**. The design phase begins before you start using eInsight and ends once the Business Process is deployed.

The second phase is called *runtime*, which is discussed in the section **"Runtime Phase Overview"**. Runtime refers to the tasks that you perform after the Business Process is deployed.

4.5.1 Business Process Modeling and Design

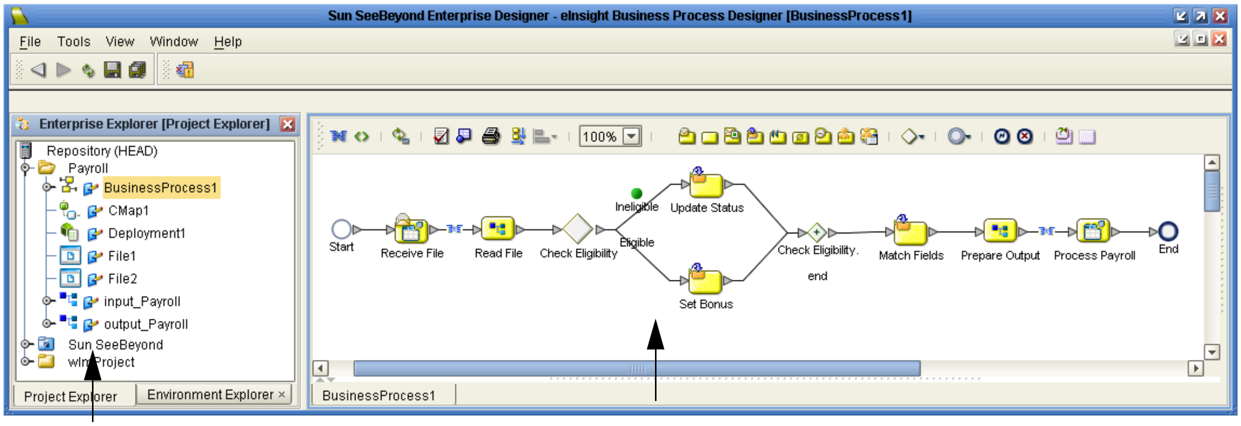
You can use eInsight to streamline operations by creating business logic that helps you reach outward to include customers and trading partners. Using eInsight to implement Business Process Management removes inefficiencies by orchestrating a unified work flow. This flow can include multiple systems/users, therefore extending to customers.

The eInsight graphical user interface (GUI) allows you to model the Business Processes that your department or even your entire company performs on a regular basis. The tools provided allow for various scenarios and events that may take place in your process.

4.5.2 Business Process Designer

The Business Process Designer serves as the front-end design tool used to create a visualization of your business workflow and increase understanding of the Business Processes involved. As the business user, you are able to integrate logic into the Business Process. When used with eInsight, the Enterprise Designer includes the following areas, as shown in Figure 9.

Figure 9 eInsight Business Process Designer



Project Explorer

Business Process Designer

- **Project Explorer** - Displays a hierarchical representation of all the Business Process models and related Java CAPS components. This view shows what is currently displayed on the modeling canvas.
- **Business Process Designer** - Used to graphically create the Business Process model in the form of an activity diagram.

4.5.3 Design Phase Overview

The basic steps that you will perform to design a Business Process model are as follows:

1. Plan and design a model that represents a business process taking place in your company.
2. Set up a Project and related components necessary for your Business Process model.
3. Create the new Business Process model in the Business Process Designer, using activities, links, decision and exception handling logic, and any other modeling elements that express the actual Business Process.
4. Validate, generate, and save the Business Process code to the Java CAPS Repository server, where Business Process configuration and deployment information is stored.
5. Create a Connectivity Map to configure the relationship between your Java CAPS components.
6. Select or create an Environment where your Business Process will run.
7. Select or create a Deployment Profile and activate it to complete the deployment process. The eInsight Engine is now deployed to the Integration Server.

4.5.4 Advanced design phase tasks

You can perform the following tasks in the design phase, based upon your configuration.

- Create human workflow tasks using an eVision interface and User Activity element. These custom web pages can be configured to interact with Business Processes or track and view system exceptions or employee assignments. Human workflow are the actual tasks and assignments that a person performs.
- eInsight supports the different ways an organization defines its company structure. You may define expression-based task assignments to users, groups, and/or roles.
- Configure and maintain persistence and recoverability functions. The instance data is stored in a database that you configured in the design phase. The eInsight Engine writes to this database at runtime.

4.5.5 Runtime Phase Overview

After you have completed all of the design phase tasks and your system is running, you can monitor and manage Business Process activity and the overall Project with the Enterprise Manager. These tasks are only available if you use a database.

eInsight Engine

The eInsight Engine provides process coordination that enables the execution of Business Processes, Activities, and Tasks. During the runtime phase, the eInsight Engine does the following:

- Receives messages that instantiate Business Process instances
- Writes monitoring, persistence, and recoverability data to a database (if available)

eInsight Database

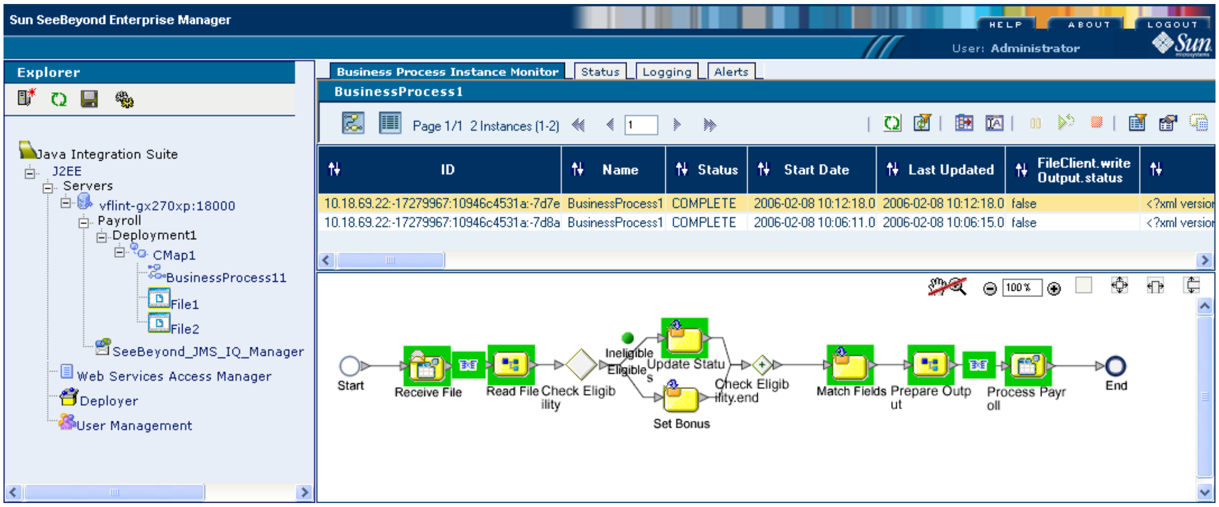
eInsight components connect to the optional database to provide:

- **Persistence** - The eInsight Engine writes instance data to a database to ensure that data is able to persist in the system.
- **Recoverability** - Using a database allows you to recover data from the last state of consistency.
- **Monitoring** - Instance data is written to a database and then read by the Enterprise Manager to provide current and historical system information.

4.5.6 Business Process Monitoring and Management

Once Business Processes are up and running, the Enterprise Manager provides visibility into the state of each Business Process Activity. This interface allows organizations to monitor and manage all aspects of the Business Process.

Figure 10 Business Process Monitor



ID	Name	Status	Start Date	Last Updated	FileClient.write Output.status	
10.18.69.22:-17279967:10946c4531a:-7d7e	BusinessProcess1	COMPLETE	2006-02-08 10:12:18.0	2006-02-08 10:12:18.0	false	<?xml versior
10.18.69.22:-17279967:10946c4531a:-7d8a	BusinessProcess1	COMPLETE	2006-02-08 10:06:11.0	2006-02-08 10:06:15.0	false	<?xml versior

The actions that you can perform on a Business Process from the Enterprise Manager are called *Business Process Management* tasks. Some actions that you can take from the Enterprise Manager include viewing and identifying errors originating from erroneous data. You can ensure that processes continue to run and work properly with these powerful monitoring tools.

Chapter 5

Sun SeeBeyond eVision Studio

This chapter describes Sun SeeBeyond eVision™ Studio, a graphical design studio for the creation of integrated web applications.

What's in This Chapter

5.1 Summary of Features

eVision Studio is a graphical environment that enables business analysts and web developers to rapidly create interfaces for composite applications, without the need for advanced programming abilities. When drag-and-drop tasks are performed using the eVision tools, the underlying programming code is automatically generated.

eVision provides two main tools: the Page Layout Designer and the Page Flow Designer. Both tools are integrated within Enterprise Designer.

The Page Layout Designer is used to create web pages. This tool includes a design canvas, GUI object palettes, a properties window, and a toolbar. Among the objects in the palettes are check box groups, text boxes, hyperlinks, image maps, progress bars, submit buttons, and charts. The properties window enables the developer to specify a variety of attributes for the GUI objects and the overall web page. The toolbar includes functions such as cut, copy, and paste; left align, right align, top align, and bottom align; and preview.

The Page Flow Designer is used to manage the web pages within a composite application. The web pages are combined with specialized elements to create a Page Flow. Page Flows can include data mappings from one component to the next. These mappings are configured using a business rule designer.

eVision Studio can be used in a variety of ways as a front end to Java CAPS solutions. For example:

- In conjunction with eInsight Business Process Manager, eVision Studio enables users to participate in business processes. The users perform workflow tasks with eVision Studio web pages that are tailored for specific organizational roles.

- eVision Studio can provide the login pages in which users enter their user name and password, which are then authenticated.

5.2 eVision Studio and Java CAPS

eVision Studio is part of the integration suite of products. eVision Studio provides web application design and deployment features and functions to Java CAPS.

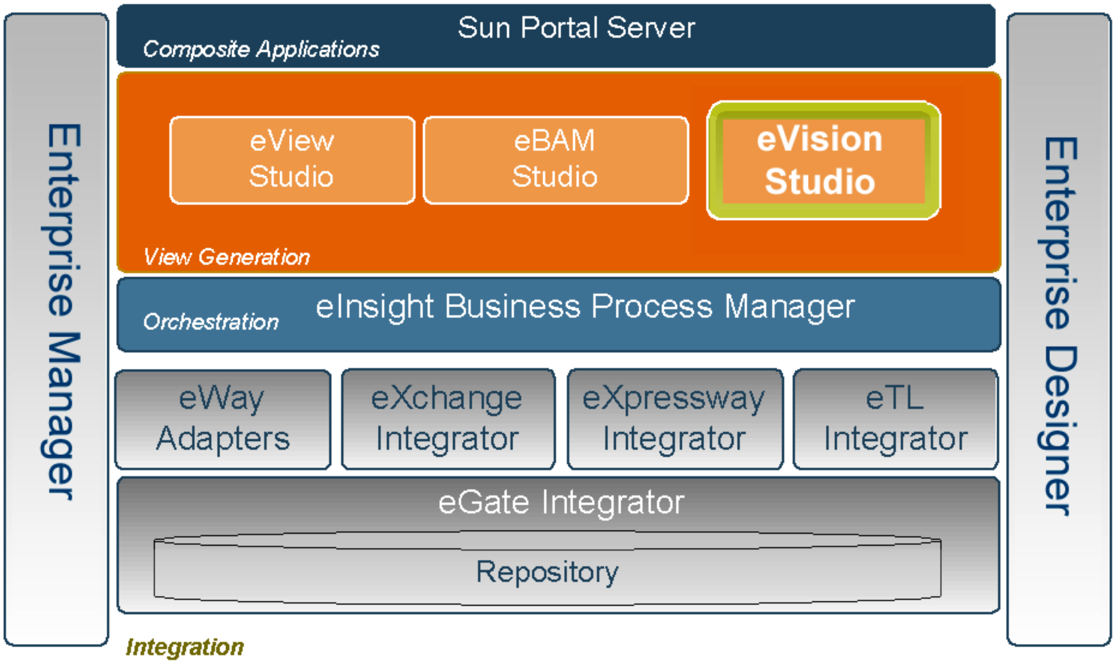
eVision Studio simplifies the task of developing web applications by using patterns and metaphors that are familiar to web developers, providing graphical abstractions of back-end data, and modeling of user/system interactions.

5.2.1 Suite Integration

eVision Studio is tightly integrated with Java CAPS and runs as a component within the suite's environment. The Page Layout Designer and Page Flow Designer run as components within Enterprise Designer. eVision Studio web applications are stored in the Sun SeeBeyond Repository.

Figure 11 shows how eVision Studio is integrated with Java CAPS.

Figure 11 eVision Studio Integration With Java CAPS



- eVision Studio runs as a component within Enterprise Designer.
- Page Layouts, Page Flows, GUI objects, and Deployment Profiles are stored in the Sun SeeBeyond Repository.

- The Page Flow Engine, which coordinates all process-related activity of a deployed project, runs in a Sun SeeBeyond Integration Server.

5.3 eVision Studio Overview

eVision Studio applications receive data from and send data to Java CAPS components, allowing users to interact with the data at run time. Web applications enable the distribution of integrated business processes across the enterprise and allow real-time user interaction with those processes. eVision Studio allows the organization to present a single, unified view of enterprise data and applications to employees, customers, and partners.

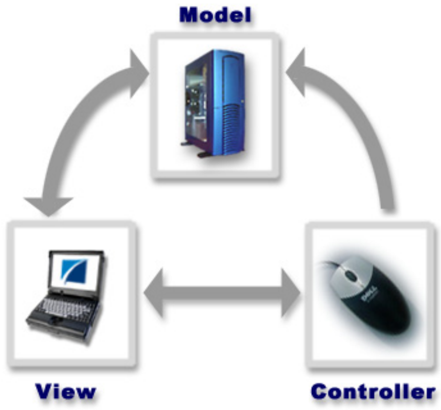
With eVision Studio, the web developer can create personalized views of business information, enabling web application users to interact with running business processes in real time, while working with only the information that they need to see. Web applications can be structured to allow employees to see what tasks are assigned to them, then to interact with and complete the tasks.

5.3.1 MVC Architecture

eVision Studio applications are based on the Model/View/Controller (MVC) architecture. MVC is a software development paradigm that enhances the task of building software systems, particularly those that generate multiple, synchronized presentations of the same data. For example, MVC is ideal for the development of a graphical statistical presentation application that requires simultaneous rendering of the same data in bar, line, and pie chart formats.

The MVC architecture consists of three types of objects: the Model, the View, and the Controller.

Figure 12 Model/View/Controller (MVC) Architecture



The Model object represents the data in a program, which manages behaviors and data within an application. The Model responds to requests for information about its current state (typically requested by the View) and responds to instructions to change its state (typically requested by the Controller).

The View object manages the visual display of the Model data; for example, displaying graphics and text to users in a browser.

The Controller object enables user interaction with the Model data; for example, mouse and keyboard inputs from the user, which instruct the Model and/or View to perform an action.

eVision Studio's application architecture fully supports the MVC paradigm. In an eVision Studio web application, user input, modeling of the external world, and visual feedback are managed by MVC objects, where each object is specialized for its task. For example:

- The Model, represented by the Page Flow, contains the business logic (Object Type Definitions and Collaborations) that interacts with the back-end system applications.
- The View contains the JavaServer Pages™ software components that are generated with the Page Layout Designer.
- The Controller is the web-enabled Page Flow created with the Page Flow Designer. The Controller orchestrates the sequence of pages being sent to the browser in response to user actions.

5.4 Process Overview

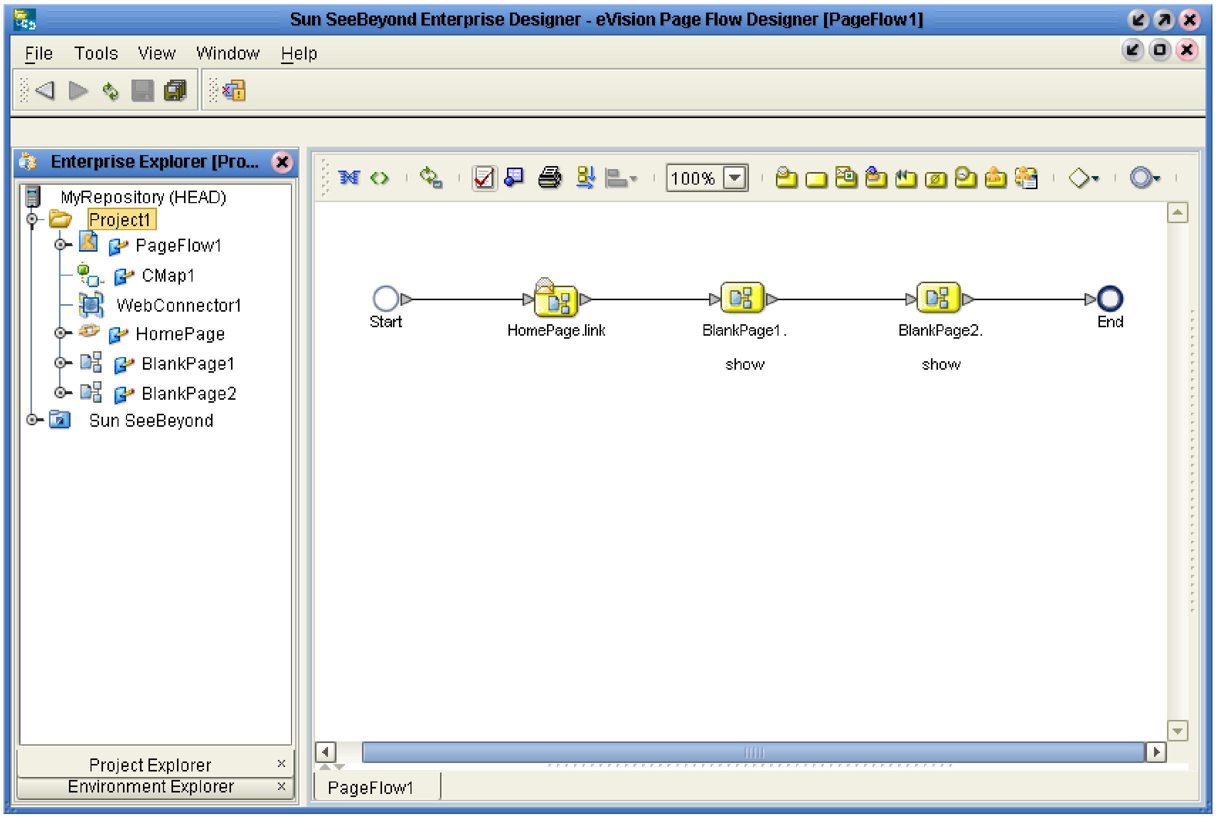
The steps for the web application development process are:

- Page Layout - Using the Page Layout Designer to create web pages with pre-built GUI objects.
- Page Linking - Using the Page Layout Designer to create links to web pages and to pass parameters from one page to another.
- Page Flow - Using the Page Flow Designer to connect finished web pages to create a logical flow for the underlying business process.
- Binding and Deployment - Deploying web applications into an Environment for integration with other integration suite run-time components.

5.4.1 User Interface Components

eVision Studio leverages the familiar Enterprise Designer interface to create, manage, deploy, and integrate web applications.

Figure 13 eVision Studio in Enterprise Designer



Page Layout Designer

The Page Layout Designer enables the web designer to specify the details of individual pages in the application by using familiar drag-and-drop techniques to place GUI objects (web page design objects) onto a design canvas.

The Page Layout Designer provides a comprehensive collection of pre-built GUI objects and web page editor.

Page Flow Designer

The Page Flow Designer facilitates the flow of web-based business activities. The functions include Page Flow modeling, monitoring, and execution, as well as the ability to analyze how data messages flow from page to page.

The Page Flow Designer enables the business analyst to lay out the user workflow of a web application by modeling the high-level, logical, page-by-page flow that users will follow through the web application to complete a given task (the Page Flow). The business analyst can lay out a web application's page flow by using the Page Flow Designer's set of graphical modeling tools. At run time, the Page Flow drives the display of eVision Studio web pages and orchestrates the interactions with back-end systems.

Through the deployment of web pages that are developed with the Page Layout Designer and Page Flow Designer, the business analyst can enable enterprise-wide, internal and external access to web applications across an intranet or the Internet.

5.4.2 Page Flow Engine

The Page Flow Engine orchestrates the system responses to the execution of web page component code. At run time, the Page Flow Engine executes Page Links, receives and processes user input, and, based on human interaction, moves the viewer from page to page until the underlying process is complete.

Chapter 6

eView Studio

This chapter explains the basic operation and features of the Sun SeeBeyond eView™ Studio and of the indexing applications created by eView Studio.

What's in This Chapter

6.1 Summary of Features

eView Studio provides a flexible framework that allows you to create matching and indexing applications, known as enterprise-wide master indexes. A master index uniquely identifies and cross-references the business objects stored in your system databases, creating a single view of all like objects. Business objects can be any type of entity about which you store information, such as customers, members, vendors, businesses, inventory items, and so forth. eView Studio provides your business with a powerful assortment of features that you can use to create master indexes.

6.1.1 eView Studio

The eView Studio tools provide your business with flexibility in designing and creating indexing applications. This flexibility allows you to perform the following tasks.

- Rapidly develop a master index for any type of business entity using a wizard to create the framework and using XML documents to configure the attributes of the index.
- Automatically create the primary components of the master index.
- Customize the strategies that determine the field values to populate into the single best record (SBR).
- Configure the matching algorithm and logic by specifying the fields to standardize, the fields to use for matching, and the matching logic to use for each field.

- Incorporate a Java API that is customized to the object structure you define. You can call the operations in this API in the Collaboration Definitions or eInsight Business Processes of different Projects.

6.1.2 Master Index

The components of the master index are designed to uniquely identify, match, and maintain information throughout a business enterprise. These components are highly configurable, allowing you to create a custom master index suited to your specific data processing needs. Master indexes created with eView Studio provide the following features.

- **Centralized Information** - The master index maintains a centralized database, enabling the integration of data records throughout the enterprise while allowing local systems to continue operating independently. The index stores copies of local system records and of SBRs, which represent the most accurate and complete data for each object.
- **Configurability** - Before deploying the master index, you define the components and processing capabilities of the system to suit your organization's processing requirements. You can configure the object structure, matching and standardization rules, survivorship rules, queries, Enterprise Data Manager (EDM) appearance, and field validation rules.
- **Cross-referencing** - The master index is a global cross-indexing application that automates record-matching across disparate source systems, simplifying the process of sharing data between systems. The master index uses the local identifiers assigned by your existing systems as a reference for cross-indexing, allowing you to maintain your current systems and practices.
- **Data Cleansing** - The master index uses configurable matching algorithm logic to uniquely identify object records and to identify duplicate and potential duplicate records. The index provides the functionality to easily merge or resolve duplicates. The index can be configured to automatically merge records that are found to be duplicates of one another.
- **Data Updates** - The master index provides the ability to add, update, deactivate, and delete data in the database tables through messages received from external systems. Records received from external systems are checked for potential duplicates during processing. Updates can be performed in real time or as batch processes.
- **Identification** - The master index employs configurable probabilistic matching technology, using a matching algorithm to formulate an effective statistical measure of how closely records match. Using a state-of-the-art algorithm in real-time mode and establishing a common method of locating records, the index consistently and precisely identifies objects within an enterprise.
- **Matching Algorithm** - The matching algorithm and logic used by eView Studio is highly configurable. eView Studio uses the Sun SeeBeyond Match Engine (SBME) for standardization and matching, but it can also be implemented with other match engines that provide a compatible API. The matching and standardization logic of the match engine is customizable.

- **Unique Identifier** - Records from various systems are cross-referenced using an enterprise-wide unique identifier (EUID) that the master index assigns to each object record. The index uses the EUID to cross-reference the local IDs assigned to each object by the various computer systems throughout the enterprise.

6.2 eView Studio and Java CAPS

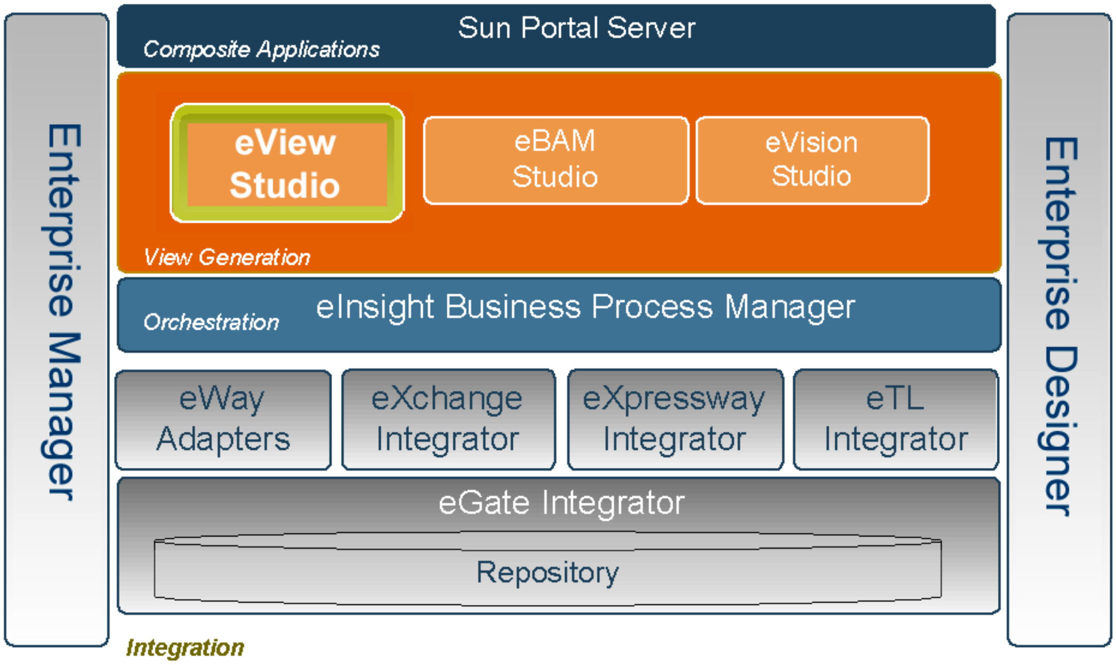
eView Studio is a part of the Java Composite Application Platform Suite. eView Studio is highly configurable, allowing you to define the data structure of the information to be indexed and to define the logic that determines how data is updated, standardized, weighted, and matched in the master index database. The master indexes created by eView Studio provide accurate identification of objects throughout your organization and cross-reference an object’s local identifiers using an enterprise-wide unique identification number (EUID) assigned by the master index. The master index also ensures accuracy by identifying potential duplicate records and providing the ability to merge or resolve duplicate records. All information is centralized in one shared index. Maintaining a centralized database for multiple systems enables the indexing application to integrate data throughout the enterprise while allowing local systems to continue operating independently.

6.2.1 Java CAPS Products

eView Studio and the indexing applications it creates are tightly integrated with Java CAPS and run as components within the Java CAPS environment. You can create multiple indexing applications, each uniquely identifying different types of business objects.

Figure 14 illustrates some of the eView Studio and Java CAPS components that work together.

Figure 14 eView Studio and Java CAPS



- The eView Wizard, eView Studio editors, and Custom Plug-in module run as components within **Enterprise Designer**.
- Scripts for creating and customizing the master index databases are stored in, and can be run from, **Enterprise Designer**.
- eView Studio configuration files, connectivity components, and deployment profiles are stored in the **Sun SeeBeyond Repository**.
- The eView Manager Service, which coordinates the activities of a deployed indexing application, runs within the **Sun SeeBeyond Integration Server**. or the **Sun Java System Application Server**.
- Messages sent to the indexing applications can also be processed through **eInsight** for enhanced business management functions.
- Information in eView Studio databases can be accessed via Web pages created by **eVision**, as well as through the eView Studio database maintenance and monitoring system.
- Web-based process monitoring is provided by the **Enterprise Manager Monitor**.

6.2.2. Java CAPS Services

The Java Composite Application Platform Suite has a wide range of functions that it shares with all of the suite products. eView Studio is able to leverage many platform level services, including the following.

- **Resource Management** - The suite uses a distributed and open architecture that enables components to access system resources (memory and processing power) as needed and in conjunction with other components.
- **Security** - The suite provides a security module for eView Studio, to fulfill security needs such as user authentication and authorization of access to eView Studio functions and to the functions of the indexing applications created by eView Studio.
- **Repository storage and access** - The setup, component, and configuration information for the elements of a Project, including database scripts, application files, runtime files, and related eView Studio components, are stored in the Repository.
- **Deployment abilities** - The suite provides deployment profiles that contain all the information necessary to activate an eView Studio Project and deploy the indexing applications to the server. When a deployment profile is activated, the eView Studio master indexes are deployed to the server, enabling access to the database.
- **Monitoring** - The Enterprise Manager Monitor lends web-based monitoring abilities to eView Studio, allowing you to observe and correct message transfer and routing activities.
- **Connectivity Mapping** - The Connectivity Map maintains the relationships between the eView Studio master indexes and other system components.
- **Version Control** - This feature allows you to maintain multiple versions of the same components through a check in and check out process.
- **Impact Analysis** - Impact Analysis allows you to view how changes to one component will impact other components of a Project or all Projects in the Repository.

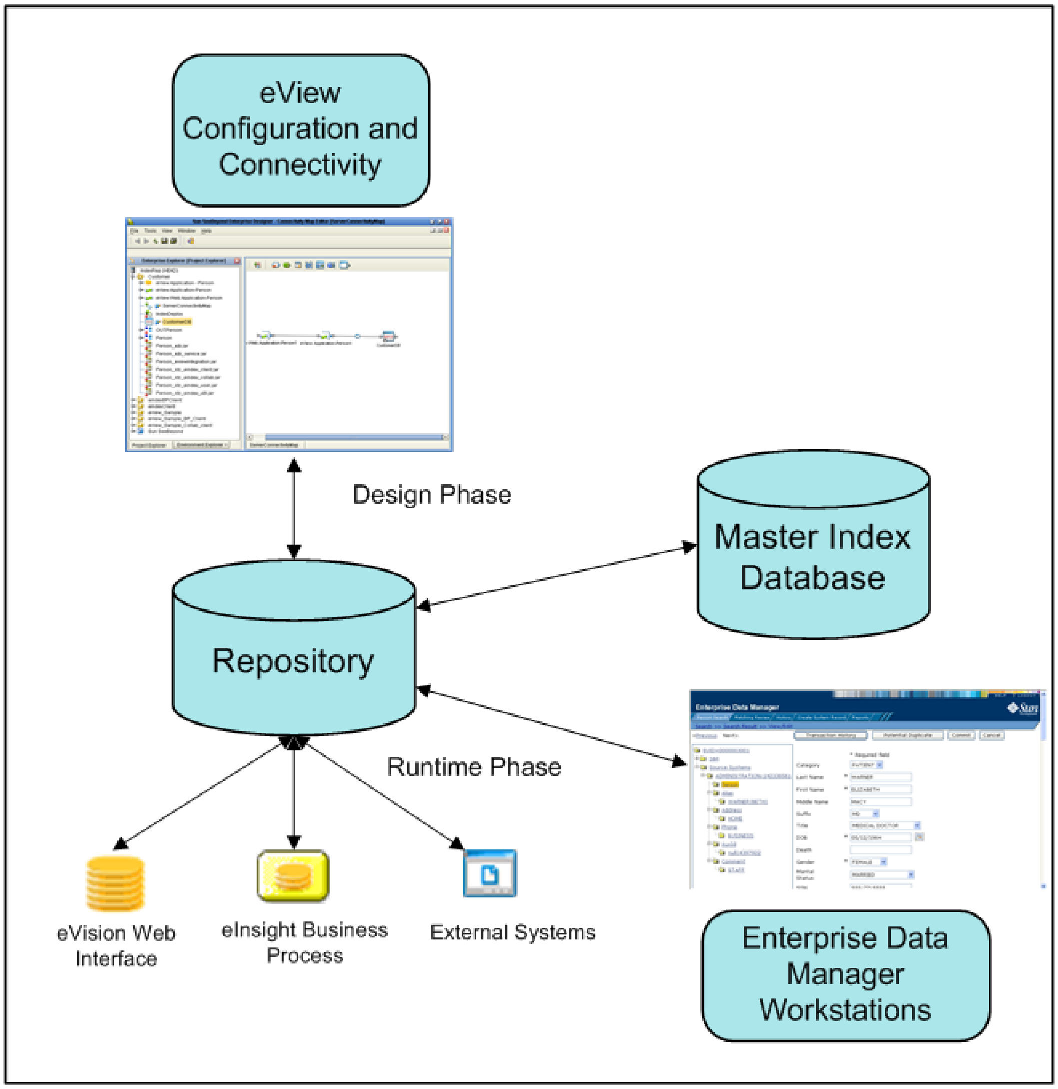
6.3 eView Studio Overview

eView Studio provides the tools that allow you to create and configure an enterprise-wide master index for any type of data. The eView Wizard guides you through the initial setup steps and special editors are provided so you can further customize the configuration, processing rules, and database structure of the master index. The eView Wizard automatically generates the components you need to implement a master index.

6.3.1. eView Studio Architecture

The components of eView Studio are designed to work within Enterprise Designer to create and configure the master index and to define connectivity between external systems and the master index. In Figure 15, you can see how the different components interact.

Figure 15 eView Studio Architecture


eView Configuration and Connectivity
Design Phase
Master Index Database
Repository
Runtime Phase
eVision Web Interface
eInsight Business Process
External Systems
Enterprise Data Manager Workstations

6.4 Process Overview

Several eView Studio tools in Enterprise Designer help you create the master index, including the eView Wizard, application generator, and eView Studio editors. The Projects associated with eView Studio will also include several standard eGate components, such as OTDs, Collaborations, queues, topics, a deployment profile, and so on. You can use the standard Enterprise Designer editors to create these components.

6.4.1 Analysis and Design Phase Overview

The process of creating a master index begins with a thorough analysis of the data you plan to store in the master index database and to share among external systems. The results of this analysis define the structure of the information stored in the master index database and provide information to help you customize the processing and matching logic for the master index.

From this analysis you can design the object structure, matching and standardization logic, any required custom processing, and the connectivity components for the indexing system. The analysis results can also be used to design the physical components of the system, such as the Logical Hosts, networking components, and master index database.

6.4.2 Development Phase Overview

The development phase consists of standard tasks for creating an indexing application and advanced tasks for further customizing the applications you create.

Standard Development Tasks

The following steps outline the basic procedure for developing a master index using eView Studio.

1. Create an eGate Project, and then create a new eView Studio application within that Project.
2. Using the eView Wizard, define the data and message structures, the operating environment, and external systems sharing data with the master index.
3. Customize the configuration files and generate the application.
4. Customize the database scripts, and then create the database and load any startup data.
5. Create and define connectivity components, and then create a Connectivity Map to configure the relationship between these components.
6. Define the Environment and configure the application server and database connections.
7. Create the deployment profile and enable it to complete the deployment process. The indexing application is now deployed to the Integration Server.
8. In the Enterprise Manager Monitor, define EDM security.

Advanced Development Tasks

You can perform additional tasks during the development phase to customize your indexing application further.

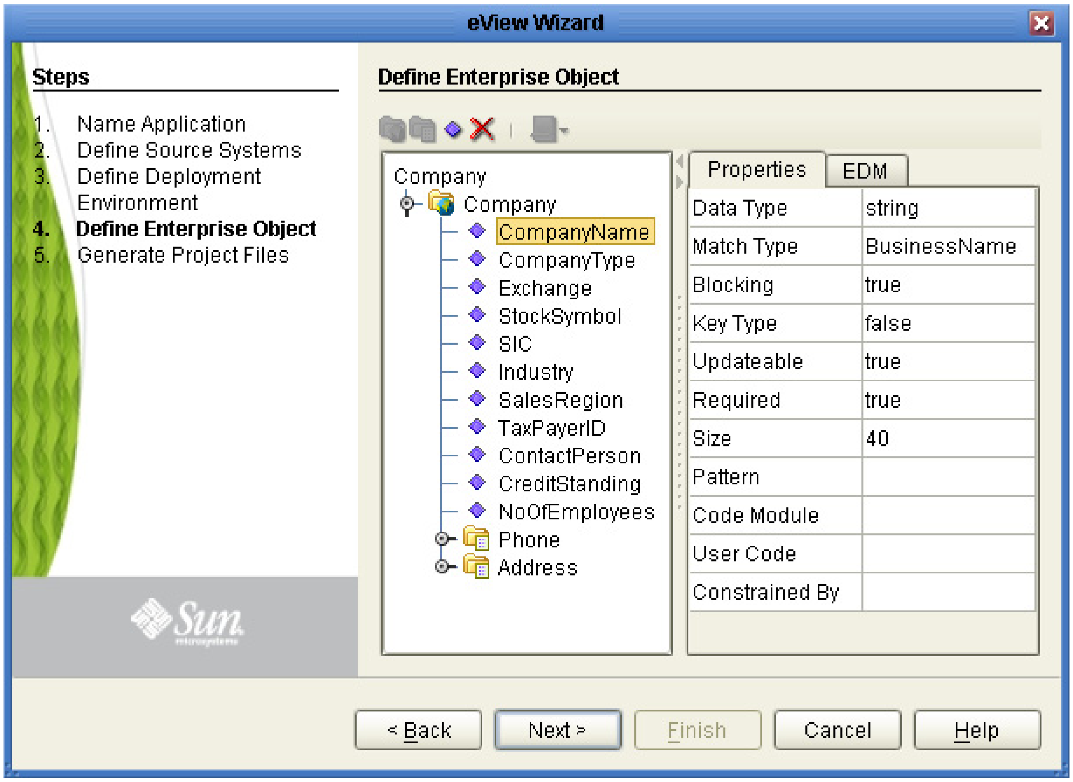
- **Custom Plug-ins** - Create Java classes to perform custom processing during the matching process and once the matching process is complete (such as performing additional operations before finalizing a transaction or validating certain field values).
- **Database Distribution** - Before running the predefined scripts against the database, create additional tablespaces to distribute the tables of the master index.

- **Match Engine Configuration** - Customize how weighting is performed by modifying the match engine configuration files. For the SBME, these files are included in the eView Studio Project.
- **eInsight Integration** - When you generate the eView Studio Project, a set of operations are created that are specifically tailored to the object structure you defined. Use these operations to integrate the indexing application with an eInsight Business Process.

eView Wizard

The eView Wizard takes you through each step of the master index setup process and, based on the information you specify, creates the XML files that define the configuration of the application.

Figure 16 eView Wizard



eView Wizard

Steps

1. Name Application
2. Define Source Systems
3. Define Deployment Environment
4. **Define Enterprise Object**
5. Generate Project Files

Define Enterprise Object

Properties	EDM
Data Type	string
Match Type	BusinessName
Blocking	true
Key Type	false
Updateable	true
Required	true
Size	40
Pattern	
Code Module	
User Code	
Constrained By	

The eView Wizard guides you through the following steps to set up the master index configuration.

1. **Name Application** - Create a unique name for the master index.
2. **Define Source Systems** - List each external system to be connected to the master index.
3. **Define Deployment Environment** - Specify the database platform, match engine type, and date format.

4 **Define Enterprise Object** - Define the data structure of the master index.

5 **Generate Project Files** - Create the framework of the master index.

6.4.3 Runtime Phase Overview

Once all of the analysis, design, and development tasks are complete and the system is running, you can perform any of these maintenance tasks.

- Transform and route data between external systems and the master index application (where the matching process occurs)
- Monitor and manage activities and the overall Project with the Enterprise Manager Monitor
- Monitor and maintain the indexed records in the master index database using the EDM

eView Manager Service

The eView Manager Service provides a session bean to all components of the master index, such as the EDM, Query Builder, Update Manager, and so on. During the runtime phase, the eView Manager Service performs the following tasks.

- Manages connectivity to the master index database
- Specifies the query to use for the match process and the system parameters that control the match process
- Coordinates the activities of the various components of the master index, including queries, updates, object persistence, system parameters, and so on

Master Index Database

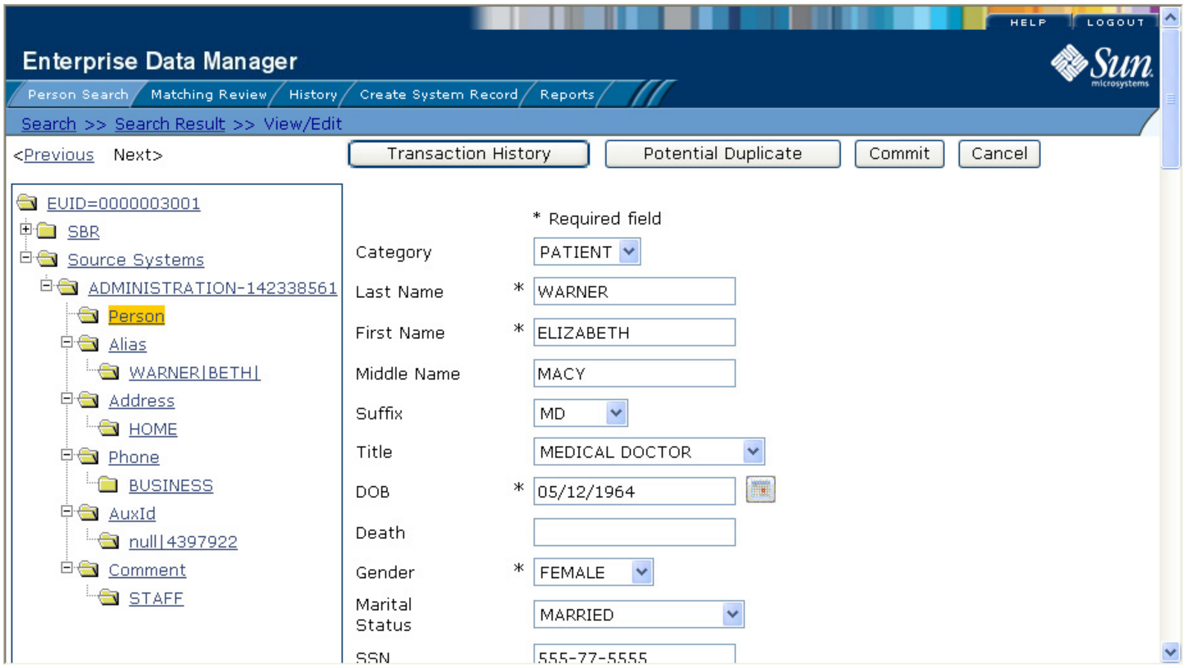
The components of a master index connect to the database to provide these features.

- **Persistence** - The Object Persistence service writes instance data to database tables to ensure that data is able to persist in the system.
- **Recoverability** - The eView Studio database allows you to recover data from the last state of consistency.
- **Transaction History** - The database stores a description of the changes that occur for each transaction. This allows you to view a complete history of changes to each record in the database.

Database Monitoring and Maintenance

The EDM is a web-based interface that allows you to monitor and maintain the data in your master index database. The appearance and search capabilities of the EDM are highly configurable to allow you to view and search for information in the way that best suits your business needs. A sample View/Edit page from the EDM is displayed in Figure 17.

Figure 17 Enterprise Data Manager



* Required field

Category: PATIENT

Last Name *: WARNER

First Name *: ELIZABETH

Middle Name: MACY

Suffix: MD

Title: MEDICAL DOCTOR

DOB *: 05/12/1964

Death:

Gender *: FEMALE

Marital Status: MARRIED

SSN: 555-77-5555

The EDM allows you to perform these primary functions to monitor and maintain the data in a master index database.

- **Transaction History** - You can view a complete history of each object for both the local system records and the single best record.
- **Data Maintenance** - You can add new records; view, update, deactivate, or reactivate existing records; and compare records for similarities and differences.
- **Search** - You can perform searches against the database for a specific object or a set of objects. For certain searches, the results are assigned a matching weight indicating the probability of a match.
- **Potential Duplicate Detection and Handling** - Using matching algorithm logic, the master index identifies potential duplicate records and provides the functionality to correct the duplication.
- **Merge and Unmerge** -You can merge records you find to be actual duplicates of one another at either the enterprise-wide unique identifier (EUID) level or the system record level. Merges made in error can easily be unmerged.

Chapter 7

eIndex Single Patient View

This chapter explains the basic operation and features of the Sun SeeBeyond eIndex™ Single Patient View (eIndex SPV).

What's in This Chapter

7.1 Summary of Features

The components of eIndex SPV are designed to uniquely identify, match, and maintain patient information throughout a healthcare enterprise. These components are highly configurable, allowing you to create a custom master patient index suited to your specific data processing needs. eIndex SPV provides your business with a powerful assortment of features, including:

- **Centralized Information** - eIndex SPV maintains a centralized database, enabling the integration of data records throughout the enterprise while allowing local systems to continue operating independently. eIndex SPV stores copies of local system records and of single best records (SBRs), which represent the most accurate and complete data for each patient.
- **Cross-referencing** - eIndex SPV is a global cross-indexing application that automates record-matching across disparate source systems, simplifying the process of sharing data between systems. eIndex SPV uses the local identifiers assigned by your existing systems as a reference, allowing you to maintain your current systems and practices.
- **Data Cleansing** - eIndex SPV uses configurable matching algorithm logic to uniquely identify object records and to identify duplicate and potential duplicate records. eIndex SPV provides the functionality to easily merge or resolve duplicates, and can be configured to automatically merge records that are found to be matches of one another.
- **Data Updates** - eIndex SPV provides the ability to add, update, deactivate, and delete data in the database tables through messages received from external systems.

Records received from external systems are checked for potential duplicates during processing. Updates can be made in real time or as batch processes.

- **Flexibility** - eIndex SPV is provided in a highly flexible framework, allowing extensive customizations to the application. Before deploying eIndex SPV you can customize the components and processing capabilities of the system to suit your organization's processing requirements. You can configure the object structure, matching and standardization rules, survivorship rules, queries, Enterprise Data Manager (EDM) appearance, and field validation rules.
- **Identification** - eIndex SPV employs configurable probabilistic matching technology, using a matching algorithm to formulate an effective statistical measure of how closely records match. Using a state-of-the-art algorithm in real-time mode and establishing a common method of locating patient records, eIndex SPV consistently and precisely identifies patients within an enterprise.
- **Matching Algorithm** - The matching algorithm and logic used by eIndex SPV is highly configurable. eIndex SPV uses the Sun SeeBeyond Match Engine (SBME) for standardization and matching, but it can also be implemented with other match engines that provide a compatible API. The matching and standardization logic of the match engine is customizable.
- **Unique Identifier** - Patients from various systems are cross-referenced using an enterprise-wide unique identifier (EUID) that eIndex SPV assigns to each patient record. eIndex SPV uses the EUID to cross-reference the local IDs assigned to each patient by the various computer systems throughout the enterprise.

7.2 eIndex SPV and Java CAPS

eIndex SPV is a part of the Java Composite Application Platform Suite. eIndex SPV is a healthcare-oriented, enterprise-wide master patient index that creates a single view of patient information by maintaining the most current information about the patients who participate throughout your organization and by linking information from different locations and computer systems. eIndex SPV provides accurate identification of patients throughout your healthcare enterprise and cross-references a patient's local identifiers using an enterprise-wide unique identification number (EUID) assigned by eIndex SPV. eIndex SPV also ensures accurate patient data by identifying potential duplicate records and providing the ability to merge or resolve duplicate records. All patient information is centralized in one shared index. Maintaining a centralized database for multiple systems enables eIndex SPV to integrate data throughout the enterprise while allowing local systems to continue operating independently.

7.2.1 Java CAPS Products

eIndex SPV is tightly integrated with Java CAPS and exists within a Project of the Java CAPS environment. The configuration tools of eIndex SPV (the editors and Custom Plug-in module) run within Enterprise Designer. In addition, eIndex SPV is integrated with the suite in the following ways.

- Scripts for creating and customizing the eIndex SPV database are stored in, and can be run from, **Enterprise Designer**.
- eIndex SPV configuration files, connectivity components, and deployment profiles are stored in the **Sun SeeBeyond Repository**.
- The eIndex Manager Service, which coordinates the activities of a deployed eIndex SPV application, runs within the **Sun SeeBeyond Integration Server** or the **Sun Java System Application Server**.
- Messages sent to eIndex SPV can also be processed through **eInsight** for enhanced business management functions.
- Information in the eIndex SPV database can be accessed via web pages created by **eVision**, as well as through the eIndex SPV database maintenance and monitoring system.
- Web-based process monitoring is provided by the **Enterprise Manager Monitor**.

7.2.2. Java CAPS Services

Java CAPS has a wide range of functions that it shares with all of the Java CAPS products. eIndex SPV is able to leverage many platform level services, including the following.

- **Resource Management** - The suite uses a distributed and open architecture that enables components to access system resources (memory and processing power) as needed and in conjunction with other components.
- **Security** - The suite provides a security module for eIndex SPV to fulfill security needs such as user authentication and authorization of eIndex SPV functions.
- **Repository storage and access** - The setup, component, and configuration information for the elements of a Project, including database scripts, application files, runtime files, and related eIndex SPV components, are stored in the Repository.
- **Deployment abilities** - The suite provides deployment profiles that contain all the information necessary to enable an eIndex SPV Project and deploy eIndex SPV to the server (enables access to the database).
- **Monitoring** - The Enterprise Manager Monitor lends web-based monitoring abilities to eIndex SPV, allowing you to observe and correct message transfer and routing activities.
- **Connectivity Mapping** - The Connectivity Map maintains the relationships between eIndex SPV and other system components.
- **Version Control** - Version Control allows you to maintain multiple versions of the same components through a check in and check out process.
- **Impact Analysis** - Impact Analysis allows you to view how changes to one component will impact other components of a Project or all Projects in the Repository.

7.3 eIndex SPV Architecture

The components of eIndex SPV are designed to work within Enterprise Designer to configure the eIndex SPV application and to define connectivity between external systems and eIndex SPV. eIndex SPV consists of several components, each acting independently of the others, but working together to provide accurate data maintenance and identification. These components interact with other Java CAPS products, such as eGate Integrator and eWays, and through them with the external data processing products and systems used by an organization.

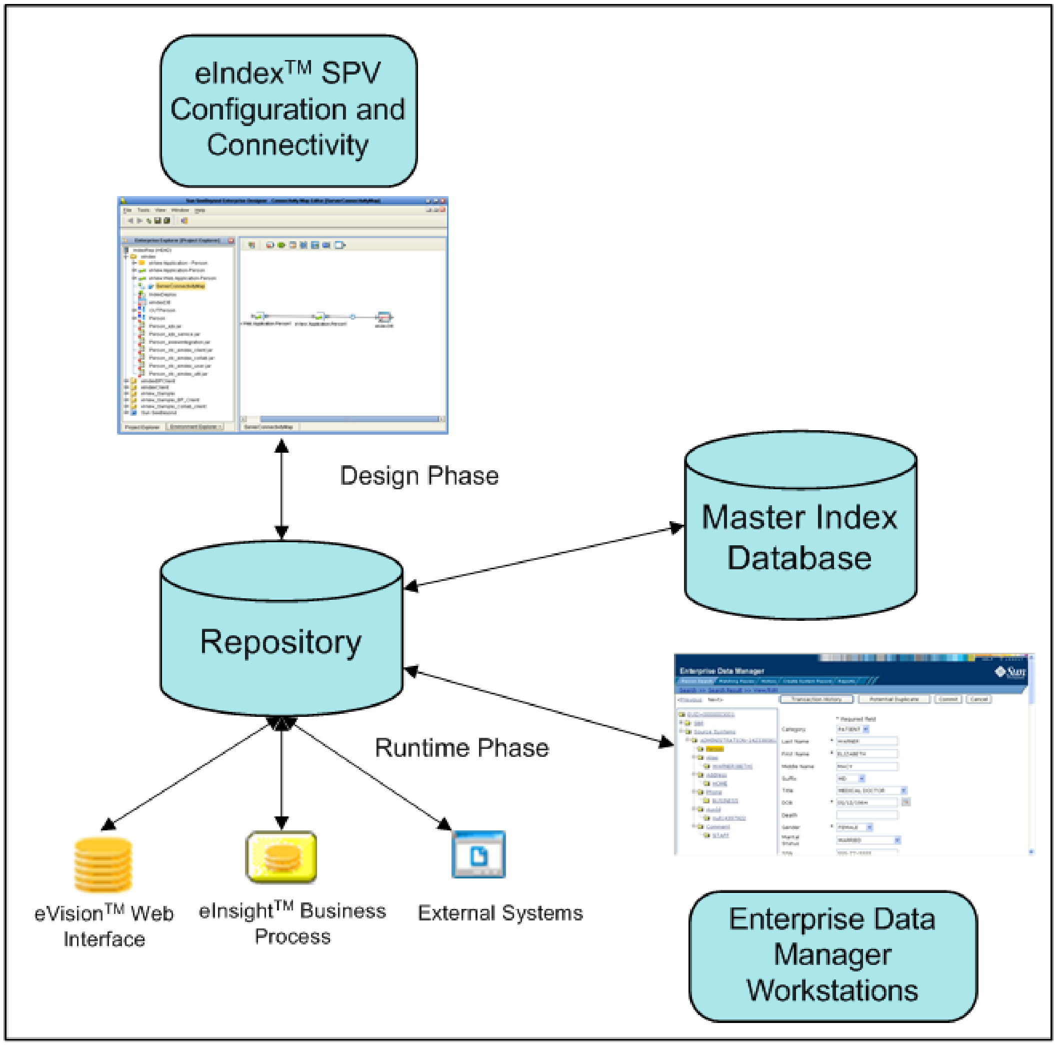
Java CAPS provides the flexible architecture required to enable the seamless sharing of customer information. As data is transferred from one local system to another using eGate, eIndex SPV consults the database and retrieves the appropriate local identifier. This is a transparent process—each computer system within this network can continue using its own local identifiers.

eIndex SPV consists of the following components:

- The Enterprise Data Manager
- An eIndex SPV database
- An eIndex SPV Project
- An eIndex SPV Environment
- Custom Java API methods

In addition, the eIndex SPV system includes one or more client Projects that can use the eIndex SPV custom Java methods in their Collaborations and Business Processes. In Figure 18, you can see how the different eIndex SPV components interact.

Figure 18 eIndex SPV Architecture


eIndex™ SPV Configuration and Connectivity
Design Phase
Master Index Database
Repository
Runtime Phase
eVision™ Web Interface
eInsight™ Business Process
External Systems
Enterprise Data Manager Workstations

7.4 Process Overview

eIndex SPV is implemented within a Project in Enterprise Designer. Several tools in Enterprise Designer help you configure eIndex SPV, including the application generator and eIndex SPV editors. The Projects associated with eIndex SPV also include several standard eGate components, such as OTDs, Collaborations, queues, topics, a deployment profile, and so on. You can use the standard Enterprise Designer editors to create these components.

7.4.1 Analysis and Design Phase Overview

The process of configuring eIndex SPV begins with a thorough analysis of the data you plan to store in the eIndex SPV database and to share among the systems connected to eIndex SPV. The results of this analysis define the structure of the information stored in the eIndex SPV database and provide information to help you customize the processing and matching logic.

From this analysis you can customize the object structure, matching and standardization logic, any required custom processing, and the connectivity components. The analysis results can also be used to design the physical components of the system, such as the Logical Hosts, networking components, and database.

7.4.2 Development Phase Overview

The development phase consists of standard tasks for configuring eIndex SPV and advanced tasks for further customizing the application.

Standard Development Tasks

The following steps outline the basic procedure for configuring eIndex SPV.

1. Customize the configuration files and regenerate the application (this updates custom components, such as OTDs and database scripts).
2. Customize the database scripts, create the database, and load any startup data.
3. If necessary, create and define connectivity components, and then create a Connectivity Map to configure the relationship between these components.
4. Define the Environment and configure the application server and database connections.
5. Create the deployment profile and enable it to complete the deployment process. The indexing application is now deployed to the application server.
6. Define EDM security using the Enterprise Manager.

Advanced Development Tasks

You can perform additional tasks during the development phase to customize your indexing application further.

- **Custom Plug-ins** - Create Java classes to perform custom processing during the matching process and once the matching process is complete (such as performing additional operations before finalizing a transaction or validating certain field values).
- **Database Distribution** - Before running the predefined scripts against the database, create additional tablespaces to distribute the tables of the master index.
- **Match Engine Configuration** - Customize how weighting is performed by modifying the match engine configuration files. For the SBME, these files are included in the eIndex SPV Project.
- **eInsight Integration** - When you regenerate the eIndex SPV Project, a set of operations are updated to be specifically tailored to the object structure you defined. Use these operations to integrate the indexing application with an eInsight Business Process.

7.4.3 Runtime Phase Overview

Once all of the analysis, design, and development tasks are complete and the system is running, you can perform any of the following maintenance tasks.

- Transform and route data between external systems and eIndex SPV (where the matching process occurs)
- Monitor and manage activities and the overall Project with the Enterprise Manager
- Monitor and maintain the indexed records in the eIndex SPV database using the EDM

eIndex Manager Service

The eIndex Manager Service provides a session bean to all components of eIndex SPV, such as the EDM, Query Builder, Update Manager, and so on. During the runtime phase, the eIndex Manager Service does the following.

- Manages connectivity to the eIndex SPV database
- Specifies the query to use for the match process and the system parameters that control the match process
- Coordinates the activities of the various components of eIndex SPV, including queries, updates, object persistence, system parameters, and so on

eIndex SPV Database

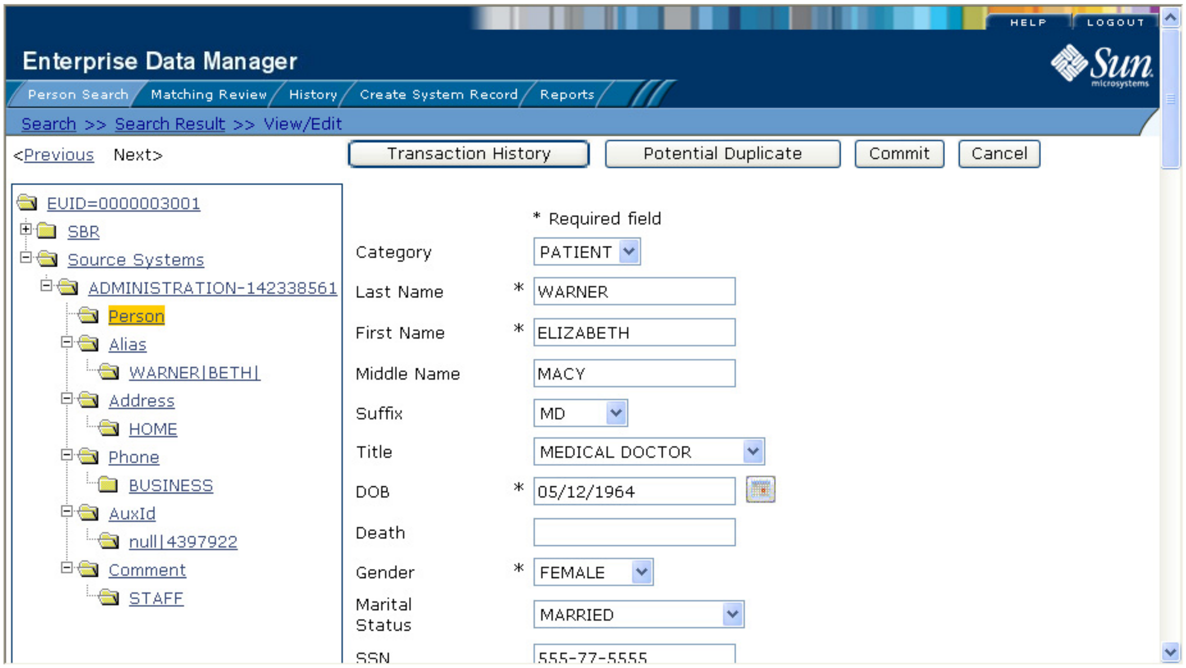
The components of eIndex SPV connect to the database to provide the following features.

- **Persistence** - The Object Persistence service writes instance data to database tables to ensure that data is able to persist in the system.
- **Recoverability** - The eIndex SPV database allows you to recover data from the last state of consistency.
- **Transaction History** - The database stores a description of the changes that occur for each transaction. This allows you to view a complete history of changes to each record in the database.

Database Monitoring and Maintenance

The EDM is a web-based interface that allows you to monitor and maintain the data in your eIndex SPV database. The appearance and search capabilities of the EDM are highly configurable to allow you to view and search for information in the way that best suits your business needs. A sample View/Edit page from the EDM is displayed in Figure 19.

Figure 19 Enterprise Data Manager



* Required field

Category: PATIENT
Last Name *: WARNER
First Name *: ELIZABETH
Middle Name: MACY
Suffix: MD
Title: MEDICAL DOCTOR
DOB *: 05/12/1964
Death:
Gender *: FEMALE
Marital Status: MARRIED
SSN: 555-77-5555

The EDM allows you to perform these primary functions to monitor and maintain the data in a master index database.

- **Transaction History** - You can view a complete history of each patient in the database for both the local system records and the single best record.
- **Data Maintenance** - You can add new records; view, update, deactivate, or reactivate existing records; and compare records for similarities and differences.
- **Search** - You can perform searches against the database for a specific patient or a set of patient records. For certain searches, the results are assigned a matching weight indicating the probability of a match.
- **Potential Duplicate Detection and Handling** - Using matching algorithm logic, eIndex SPV identifies potential duplicate records and provides the functionality to correct the duplication.
- **Merge and Unmerge** - You can merge records you find to be actual duplicates of one another at either the EUID or system record level. Merges made in error can easily be unmerged.

Chapter 8

eTL Integrator

Extraction, Transform, and Load (ETL) is a data integration methodology that extracts data from data sources, transforms and cleanses the data, then loads the data in a uniform format into one or more target data sources.

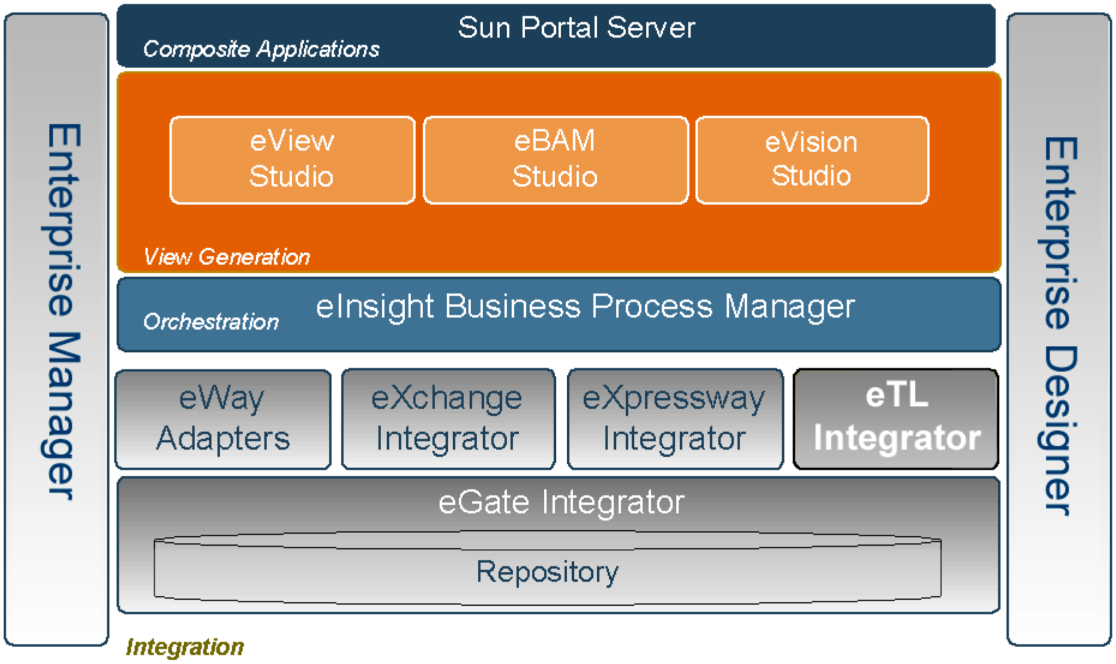
eTL Integrator provides high-volume extraction and loading of tabular data sets for Java CAPS Projects, or as a standalone product. You can use eTL Integrator to acquire a temporary subset of data for reports or other purposes, or acquire a more permanent data set for the population of a data mart or data warehouse. You can also use eTL for database type conversions or to migrate data from one database or platform to another.

eTL Integrator applies the following ETL methodology:

1. *Extraction:* the input data is extracted from data sources. Using eTL Integrator, the data can be filtered and joined from multiple, heterogeneous sources, which results in a desired subset of data suitable for transformation.
2. *Transformation*: eTL Integrator applies the operators specified for the process to transforms and cleanse the data to the desired state.
3. *Load*: The transformed data is loaded into one or multiple databases or data warehouses.

Figure 20 shows the integration between eTL Integrator and other Java CAPS products.

Figure 20 eTL Integrator and Other Java CAPS Products



Chapter 9

eBAM Studio

This chapter describes the business activity monitoring capabilities provided by Sun SeeBeyond eBAM™ Studio, and how eBAM interacts with other Java CAPS components.

What's in This Chapter

9.1 eBAM Studio

Business Activity Monitoring (BAM) involves the collection, aggregation, and presentation of business activity data according to specified Key Performance Indicators (KPIs). eBAM Studio provides the tools for generating custom, cross-application, digital dashboards for defining and monitoring KPIs that summarize the aggregated business data collected through the eInsight and eGate application layers. The eBAM Web interface allows the business analyst to transform data that has been collected over time into meaningful, rich visual presentations.

KPIs provide a context for business processes by turning raw data into useful information, allowing the business analyst to focus on monitoring and analyzing measurable enterprise-wide operations and processes. eBAM Studio provides the business analyst with different views of performance data, enabling the identification of time-critical business trends.

eBAM Studio renders real-time and historical data in familiar visual formats, such as pie and bar charts, for display in digital dashboards. These recognizable, easy-to-read contexts enable the business analyst to quickly translate information into action. Leveraging the Java Composite Application Platform Suite (Java CAPS) and the Web, eBAM provides visibility to key business information, recasts it in a graphical format, and makes it available across the enterprise.

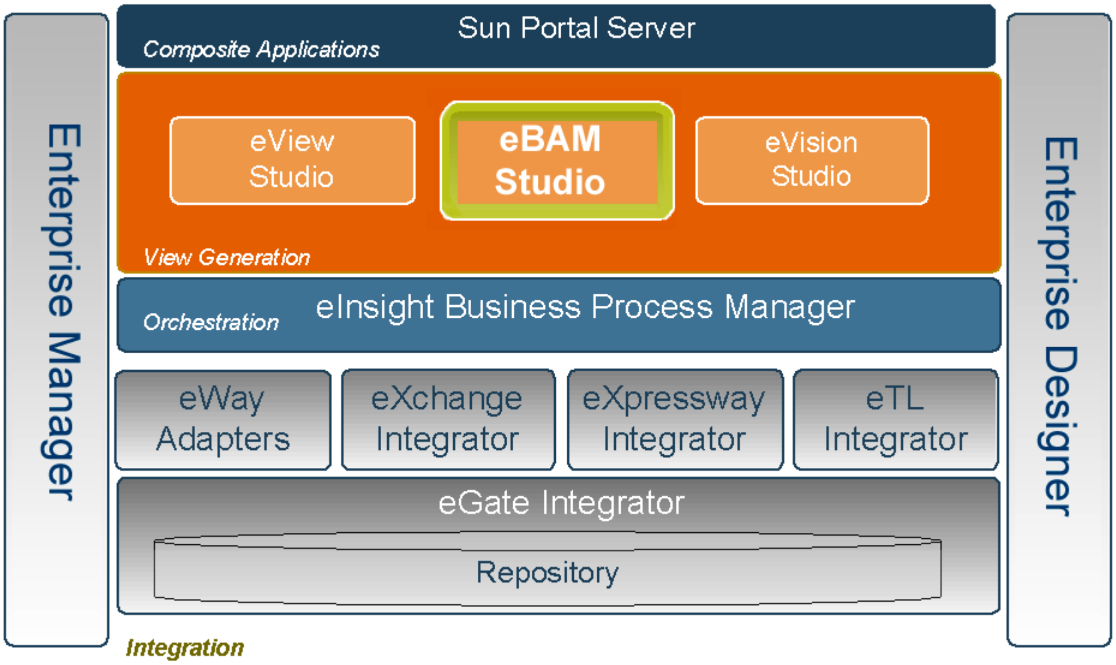
eBAM Studio leverages the integration suite platform services; for example:

- KPIs and other run-time conditions are stored in the Repository.
- Design-time provided by Enterprise Designer; run-time resource management provided by the Integration Server / Application Server.

- Enterprise Designer is used to design the data collection, graphical display, and notification processes.
- The data filtered by eBAM Studio can be queried by eInsight or eVision and then mapped to other components.
- Monitor provides the framework for displaying the charts that eBAM constructs.
- Web applications created by eBAM Studio can be viewed either in a browser or as a portlet in a JSR 168-compliant portal server.

How eBAM Studio is integrated with the Java Composite Application Platform Suite is shown in the following figure.

Figure 21 eBAM Studio Integration with the Integration Suite



9.1.1 Features and Functions

Users create eBAM Services that contain a key element called a Data Definition. The Data Definition specifies the following:

- The source of the raw data stream to be captured and stored by the eBAM engine.
- An identifying label and a datatype for each of the data elements collected.
- For Data Definitions that use data retention: The length of time to retain data before flagging it for removal, and the frequency at which expired data is removed.

Based on user-defined threshold conditions, eBAM Studio checks for notification conditions at specified intervals. Depending on the business processes that have been set up, eBAM can deliver notifications to assigned destinations, execute stored queries, update data, and/or deliver user-defined views of the data for visual presentation in chart format.

eBAM Studio delivers a real-time view of business activities (for example, monitoring service-level agreements for enforcement), allowing the business analyst to identify problems and trends and resolve them proactively. Digital dashboards can be used as a standalone interface, or can be securely accessed from within a portal.

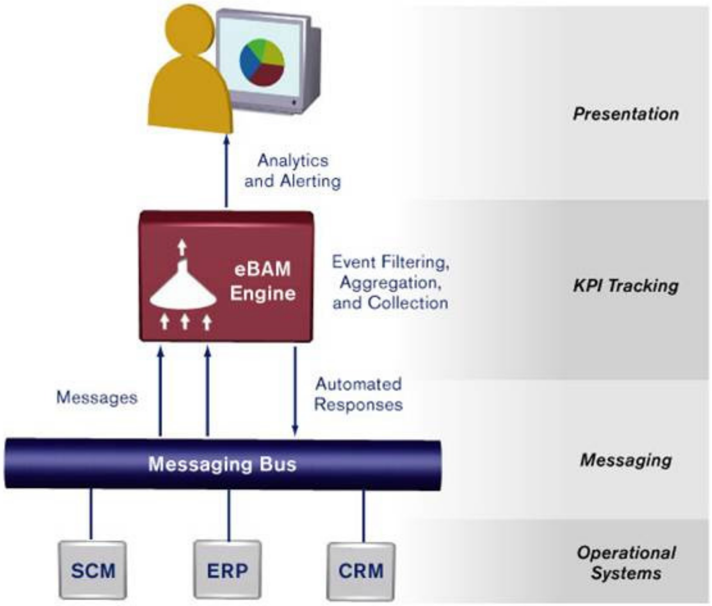
The business analyst can define the rules for monitoring business activities; eBAM provides the run-time environment for the ongoing execution of the rules, delivering notifications and updates to users via e-mail or the Web.

9.2 Using eBAM Studio

Creating an eBAM application begins when the business analyst specifies the enterprise systems and data to be monitored. The business analyst then defines the polling intervals and thresholds that will trigger automated system responses.

When deployed and activated, the eBAM application listens for specified events. When messages are collected, the eBAM application aggregates and filters the data items of interest, constructs the KPIs, measures them against the business rules and alert thresholds, and delivers the results. See Figure 22.

Figure 22 eBAM Information Delivery



9.2.1 Queries

The eBAM graphical palette is used to construct queries of mid-stream data traffic flowing through an eBAM application. A query is a stored procedure whose results are made available via a Web service; the service is used as a Business Process activity within eBAM (or eInsight, or in an eVision Page Flow), often using run-time input. Queries can reflect values of the data fields that match the defined conditions, as well computed KPI results. Query results can be further manipulated by and mapped to other integration suite components.

9.2.2 Charts

eBAM provides a wide variety of charts—pie charts bar charts, dial-type meters—to display KPI results in a clear, appealing, and understandable graphical context, and to provide real-time, persistent graphical monitoring of business processes over the Web.

9.2.3 Alerts

Notifications are triggered according to preset threshold limits set by the designer. eBAM generates an alert when a KPI falls below or exceeds a specified threshold. The alert can be used to send e-mail messages and/or to launch other processes that can be designed to perform an array of notification actions.

Glossary

BI

Business integration (also Business Intelligence).

CAPS

Composite Application Platform Suite.

Collaboration

A logical operation performed between some combination of message destinations and external applications. The operation is defined by a Collaboration Defintion, which can be encoded in either Java or XSLT.

Also see **"Service"** and **"Collaboration Definition"**.

Collaboration Definition

The encoding of business rules, in Java or XSLT format. Typically, the encoding consists of operations on OTDs (see **"OTD"**). Several Collaborations can have the same Collaboration Definition.

Connection

Consists of the configuration information that enables an eWay to connect to an external system.

Connectivity Map

Contains business logic and routing information about the data transmission. A Connectivity Map usually includes one or more Collaborations, topics, queues, and eWays. A Connectivity Map is created under a Project. A Project may have multiple Connectivity Maps.

Constants

A name or value pair that is visible across a Project.

CRM

Customer Relations Management

Data Cleansing

Data must be cleansed of errors in structure and content before it is useful in data warehousing and integration; this means transforming data for accurate and effective use in a database or data management system by cleansing "dirty" or redundant data.

Data Dictionary

Defines the organization of a database and lists all files in the database, the number of records in each file, and the names and types of each field. The data dictionary is often hidden from end users. Although the dictionary doesn't contain actual data, it does contain essential information for managing the database.

Data Integrity

Refers to the accuracy and validity of data. Data integrity can be compromised in many ways, including human error through data entry, or through faulty logic in programming. Computer viruses, software bugs and many other factors can also compromise data integrity.

Data Mapping

In relational databases (RDBMSs) data mapping is the relationship and data flow between source and target objects. Mapping involves structuring the relationship between source and target objects.

Data Mart

A smaller, focused, database designed to help managers make business decisions. (A data warehouse is a larger, enterprise, database(s).)

Data Mining

Used to synthesize or isolate unique data patterns to predict future behaviors or to filter data to select patterns that help discover previously unknown relationships among data. Commonly used by marketers who acquire and distill consumer information.

Data Transformation

Data transformation is necessary after extracting data from legacy data formats, or any format that requires cleansing. Data is transformed for efficient use for Business-to-Business Enterprise Data Integration.

Data Warehouse

A copy or view of enterprise transaction data (sometimes non-transaction data) that is used for reporting. The data is often summarized and always structured for queries and analysis.

Deployment Profile

Contains the information about how the Project components will be deployed in an Environment. A Project can have multiple Deployment Profiles, but only one Deployment Profile can be activated for a Project in any one Environment.

Derived Collaboration

Collaboration that inherits operations from another, according to standard object-oriented practice.

Dimension Table

Dimension tables describe the business entities of an enterprise; also called lookup or reference tables.

Dirty Data

Dirty data contains, but is not limited to, incorrect data including spelling errors, punctuation errors, incorrect data referencing, incomplete, inconsistent, outdated, and redundant data.

Drill Down

To move from summary to more detailed data by "drilling down" to get it. In database terminology this might mean starting with a general category and drilling down to a specific field in a record.

eGate System

See **"Project"**.

Environment

A collection of physical resources and their configurations that are used to host Project components. An Environment contains logical hosts and external systems.

EPR

Enterprise Resource Management

ETL

Extract, Transform, Load. Extract is the process of reading data from a source database and extracting the desired subset of data. Transform is the process of converting the extracted data from its previous form into the desired form. Load is the process of writing the data into a larger database.

eView Manager Service

A component of an eView master index that provides an interface to all components and includes the primary functions of the master index.

eWay

A link between a Collaboration and an external connection including the message server connection (topic or queue) or external application.

External Application

A logical representation in an eGate Project of an external application.

External System

A representation in an eGate Project of an external application system.

Extraction

Data are extracted from a source using software tools. This first step in ETL initially "gets" the data.

Fact Table

A fact table typically contains two types of columns: those containing facts and those that contain foreign keys to dimension tables. Fact tables contain detail facts and/or summary facts.

Integration Server

J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays). Provides transaction services, persistence, and external connectivity.

JMS IQ Manager

JMS-compliant, guaranteed delivery store, forwarding, and queueing service.

Join

Matches records, which are joined by a common field, in two tables in a relational database. Often part of a Select query.

Link

The JMS Connection between a Collaboration and a topic or queue in a JMS-compliant message server.

Linked Message Destination

A reference to a Message Destination defined in another Connectivity Map.

Logical Host

An instance of the eGate runtime Environment that is installed on a machine. A Logical Host contains the software and other installed components that are required at runtime, such as application and message servers.

Management Agent

Uses J2EE technology to manage and monitor an eGate 5.0 deployment that may contain other application servers in addition to the SeeBeyond Integration Server. Defines management interfaces and services designed for distributed environments, focusing on providing functionality for managing networks, systems, and applications.

Master index

A database application that stores and cross-references information on specific objects in a business organization, regardless of the computer system from which the information originates. Also called *enterprise-wide master index*.

Matching Service

A component of an eView master index that contains the logic for the matching process.

Message Destination

A general term for a topic or queue. Two or more Projects can share a message destination that has the same name and is deployed on the same message server. A single Project may also have a single message destination referenced in multiple Connectivity Maps.

Metadata

"Data about data." Metadata describes "how," "when," and "who" about structure and format, of a particular set of data. ETL tools are used to generate and maintain a central metadata repository.

Non-normalized Data

Non-normalized data cannot be cross-referenced accurately, if at all, and causes manageability issues. Non-normalized data may be converted to normalized data.

Normalized Data

Normalization is a common database design process used to remove redundant or incorrect organization and data. The design and normalization of the database will create a maintainable data set that can be cross-referenced.

Normalized data is not only easier to analyze but also easier to expand. Normalization involves removing redundancy and correcting incorrect data structure and organization.

OLAP

Online analytical processing.

OTD

An acronym for Object Type Definition. OTDs contain the data structure and rules that define an object. An OTD is used in Java Collaboration Definitions for creating data transformations and interfacing with external systems.

Project

Contains a collection of logical components, configurations, and files that are used to solve business problems. A Project organizes the files and packages and maintains the settings that comprise an eGate system in SeeBeyond's Enterprise Designer.

Query

A request for information from a database. There are three query methods:

Choose – With this easy-to-use method, the database system presents a list of parameters from which you can choose. This method is not as flexible as other methods.

Query by example (QBE) – With this method, the system lets you specify fields and values to define a query.

Query language – With this method, you have the flexibility and power to make requests for information in the form of a stylized query using a query language. This is the most complex and powerful method.

Query Builder

A component of an eView master index that defines how queries are processed.

Queue

A JMS queue is a shareable object that conforms to the *point-to-point* (p2p, or PTP) messaging domain, where one sender delivers a message to exactly one receiver. When the SeeBeyond JMS IQ Manager sends a message to a queue, it ensures it is received once and only once, even though there may be many receivers "listening" to the queue. This is equivalent to the subscriber pooling in other queue implementations. You can reference a queue that exists in another Connectivity Map or Project.

Raw Data

Data that has not been turned into "information," through processing. Although factual and "real," raw data is unorganized.

Relational Database (RDBMS)

Short for Relational Database Management System, most often referred to as RDBMS. Data is stored in related tables. Relational databases can be viewed in many different ways.

In this system a single database can be spread across several tables. (RDBMS differs from flat-file databases where each database is self-contained as a single file or table.)

Repository

Stores and manages the setup, component, and configuration information for eGate Projects. The Repository also provides monitoring services for Projects, which include version control and impact analysis.

Schema Runtime Environment

An add-on in eGate 5.0 that provides the upgrade path for e*Gate 4.x users to upgrade to eGate 5.0. Also known as the SRE.

Service

Contains the information about executing a set of business rules. These business rules can be defined in a Java Collaboration Definition, XSLT Collaboration Definition, Business Process, eTL Definition, or other service. A Service also contains binding information for connecting to JMS Topics, Queues, eWays, and other services.

Single Best Record

Also known as the SBR, this is the best representation of an entity's information in an eView master index. The SBR is populated with information from all source systems based on the survivor strategies defined for each field.

Staging Data

Data that is to be processed before entering the warehouse.

Subproject

An independent Project that is included as part of another Project and listed on the Enterprise Explorer tree beneath the main Project icon.

Table

Refers to data arranged in rows and columns, like a spreadsheet. In relational database management systems, all information is stored in tables.

Topic

A JMS topic is a shareable object that conforms to the *publish-and-subscribe* (pub/sub) messaging domain, where one publisher broadcasts messages to potentially many subscribers. When the SeeBeyond JMS IQ Manager publishes a message on a topic, it ensures that all subscribers receive the message.

Transformation

Data that are extracted from databases are transformed into a desired form, using various tools that cleanse, merge, purge, aggregate, calculate, audit, remove redundancy, standardize, etc.

XSLT

An acronym for Extensible Stylesheet Language Transformations. A file format used in eGate to generate Collaboration Definitions.

e*Gate 4.x Terms in eGate 5.0

Table 2 provides definitions for the terms that are new with eGate release 5.0, as well as equivalent terms from eGate release 4.x.

Table 2 eGate 5.0 Terms

5.0 Term	4.x Equivalent Term
Collaboration	Collaboration
Collaboration Definition	Collaboration Definition
Connection	eWay Connection
Connectivity Map	Closest: Network View of an entire Schema
Deploy	Run the Control Broker
Deployment	<none>
Deployment Profile	Closest: Schema
Enterprise Designer	Enterprise Manager
Enterprise Manager	Enterprise Monitor
Environment	Schema (except only includes physical information, not business logic)
eWay	eWay Connection eWay
eWay Configuration	eWay Connection Configuration
External Application	eWay Connection
External System	eWay Connection
JMS Connection	eWay Connection
Integration Server	<none>
Link	JMS eWay Connection
Linked Message Destination	<none>
Logical Host	Participating Host
Message Destination	Topic or queue
Message Server	JMS IQ Manager
Object Type Definition (OTD)	Event Type Definition (ETD)
Process Manager	Control Broker
Project	Schema (except not including physical layer)
Queue	JMS queue
Repository	Registry
Subproject	Schema

Table 2 eGate 5.0 Terms (Continued)

5.0 Term	4.x Equivalent Term
Topic	JMS topic
XSLT	<none>

Index