

Conceptual Overview

iPlanet™ Integration Server

Version 3.0

August 2001

Copyright (c) 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Sun, Sun Microsystems, the Sun logo, Java, iPlanet and the iPlanet logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

This product includes software developed by Apache Software Foundation (<http://www.apache.org/>). Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

Federal Acquisitions: Commercial Software – Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright (c) 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats - Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, Java, iPlanet et le logo iPlanet sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Ce produit inclut des logiciels développés par Apache Software Foundation (<http://www.apache.org/>). Copyright (c) 1999 The Apache Software Foundation. Tous droits réservés.

Acquisitions Fédérales: progiciel – Les organisations gouvernementales sont sujettes aux conditions et termes standards d'utilisation.

LA DOCUMENTATION EST FOURNIE “EN L'ÉTAT” ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Contents

List of Figures	5
iIS Conceptual Overview	7
About iPlanet Integration Server	8
iIS Architecture	9
iIS Backbone	11
Application Proxies	11
Adapters	11
XML Messages	12
XML/XSLT Workshops	12
XSL Stylesheets	13
XSL Processor	13
Backbone Management Utility	13
iIS Process System	13
Process Engine	14
Process Development Workshops	14
Process Management Tools	15
Process Management Utility	16
Process Client APIs	16
Professional and Enterprise Editions	17
iIS Professional Edition	17
iIS Enterprise Edition	18
iPlanet UDS Environment	20
About iPlanet UDS	21
UDS System Management Services	21
Nodes in a UDS System	21
Configuring an iIS System	24
Documentation Roadmap	25
iIS Glossary	27
Index	33

List of Figures

Figure 1	iPlanet Integration Server	10
Figure 2	XML/XSLT Workshops	12
Figure 3	Process Development Workshops	15
Figure 4	Process Management Tools	16
Figure 5	iPlanet Integration Server, Professional Edition	17
Figure 6	iPlanet Integration Server, Enterprise Edition	19
Figure 7	Nodes in an iPlanet UDS Environment	22

iIS Conceptual Overview

This document provides a conceptual overview of iPlanet Integration Server, including a discussion of the components and development tools provided with your distribution.

This document contains the following sections:

- *“About iPlanet Integration Server”*
- *“iIS Architecture”*
- *“Professional and Enterprise Editions”*
- *“iPlanet UDS Environment”*
- *“Documentation Roadmap”*
- *“iIS Glossary”*

About iPlanet Integration Server

iPlanet Integration Server (iIS) is a suite of business integration tools for integrating and coordinating heterogeneous applications. The tools and software components provided with iIS let you integrate newly developed applications, legacy applications, and off-the-shelf packages (including Web browsers) into business processes that are managed and controlled by a process engine.

iPlanet Integration Server includes the following features:

- Point-to-point integration between applications

Integration is accomplished using XML messaging transported over HTTP or by using implementations of the Java Message Service (JMS). iIS adapters are available to integrate many package applications that do not have a native XML interface. The iIS adapter toolkit enables integration of any custom application.

- Business process management, using the iIS process engine

Application proxies interface with the iIS process engine to provide end-to-end control and automation of all the processes in an enterprise's business. Business process integration can combine automated systems with human interactivity.

- Message brokering and XSL data transformation, using the iIS backbone application proxies

XML messages are transformed according to XSL rules registered with the iIS backbone. Messages are then routed through the process engine and delivered to partner applications. XSL provides a standards-based way to share data across applications and domains.

- Interactive tools for process development and XML/XSLT authoring

The process development workshops allow you customize the workflow of activities in a business process, including defining user profiles, user validations, and sharing of data between activities in a process. The XML/XLST workshops include an interactive debugger.

- Robust runtime environment with failover capabilities

Interactive process management tools allow you to monitor and manage process engine performance. The process engine can be configured for failover and contains tools for the backup and restoration of process data.

- Support for multiple platforms

iIS can be deployed on multiple platforms with functionality distributed across different platforms in your environment. This allows you to deploy process management, process runtime, database access, and application proxies on the platform most suited for your system.

- Support for multiple styles of messaging

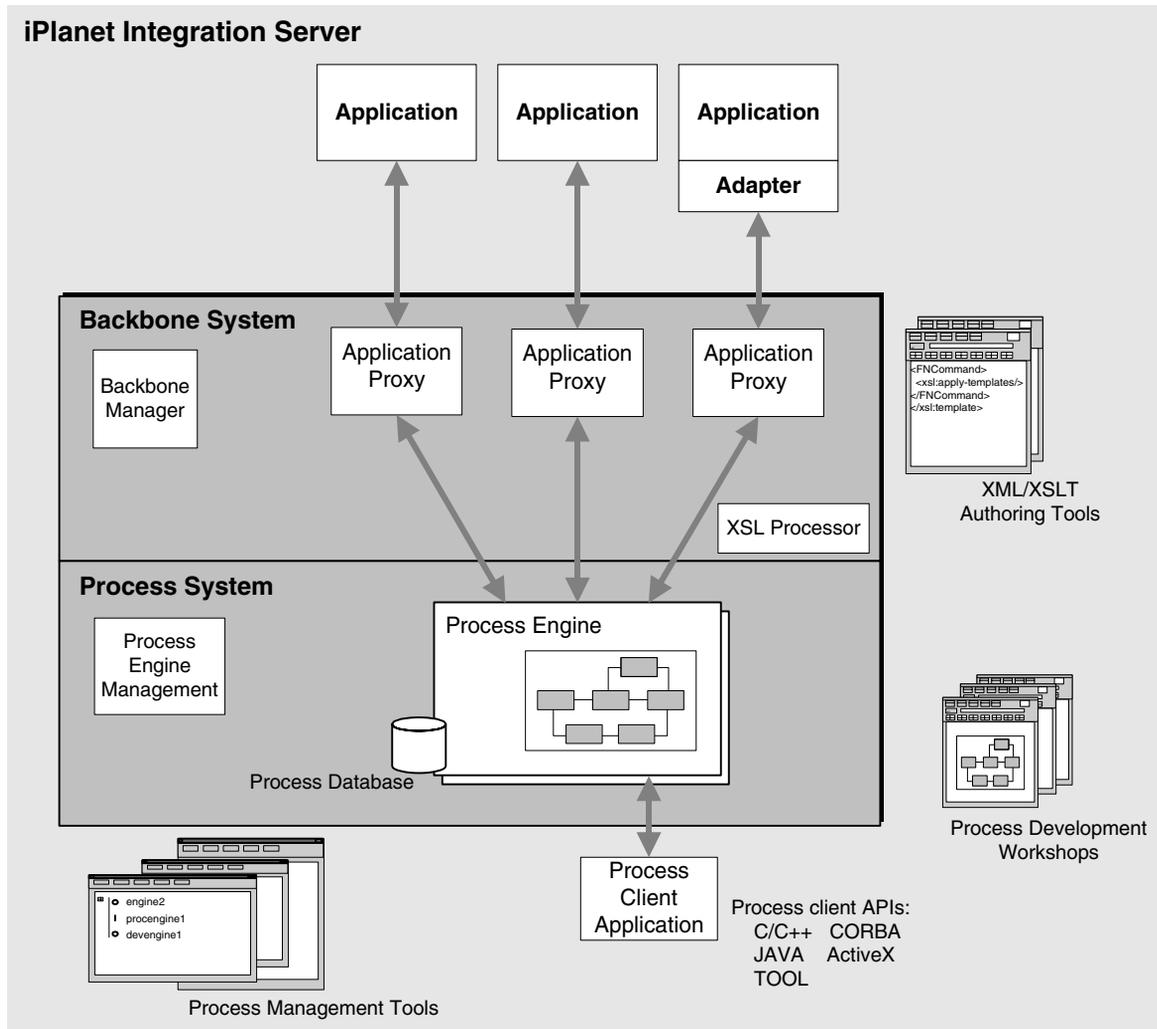
iIS supports XML messaging over HTTP and SSL, as well as messaging using implementations of JMS. iIS application proxies can also access application services made available with the Simple Object Access Protocol (SOAP). Different styles of messaging can be combined in your deployment of an iIS system.

iIS Architecture

iIS contains two systems—a backbone system and a process system. The *backbone system*, commonly referred to as the *iIS Backbone*, contains configurable application proxies that use XML-based messaging to communicate with one or more partner applications. Application proxies can also access applications that export services using SOAP. The *process system* contains an iIS process engine that manages and controls a business process, coordinating activities in the process with application proxies and process client applications.

Figure 1 shows the relationship of the backbone and process systems of iIS. Later sections discuss the components of each system, and how they can be used to integrate enterprise business applications into a business process.

Figure 1 iPlanet Integration Server



iIS Backbone

The iIS backbone system provides software tools and components that enable XML messaging between enterprise applications and an iIS process engine. iIS customers primarily use a backbone system to:

- Provide an XML interface that allows applications to exchange and translate XML messages. This allows different types of applications to communicate with each other or to communicate with the iIS process engine.
- Enable applications to participate in a managed business process.

Application Proxies

The heart of an iIS Backbone is a set of *application proxies* that perform message brokering and data transformation on behalf of applications. For business process support, proxies interact with an iIS process engine on behalf of applications that participate in a common business process. The main purpose of these interactions is to communicate the initiation and completion of work activities according to a process defined for the process engine.

Application proxies are configurable, and can act as a client, server, or client/server with respect to an application. A proxy can also represent more than one application, if appropriate.

Because proxies are configured (not coded), you can quickly specify and modify their function and behavior. Configuration information includes management of incoming and outgoing XML messages and data as well as maintaining sessions with its partner applications.

Adapters

Adapters provide an interface to packages or custom applications that lack a native XML interface into an iIS Backbone. iIS provides an adapter toolkit that allows you to create adapters according to your needs.

An iIS adapter serves the following functions:

- transforms XML documents sent by the proxy into requests invoked using the application's native API
- interprets the responses to these requests and generates appropriate application documents to be sent to the proxy
- responds to messages or application callback requests, generates appropriate documents, and forwards these to the proxy

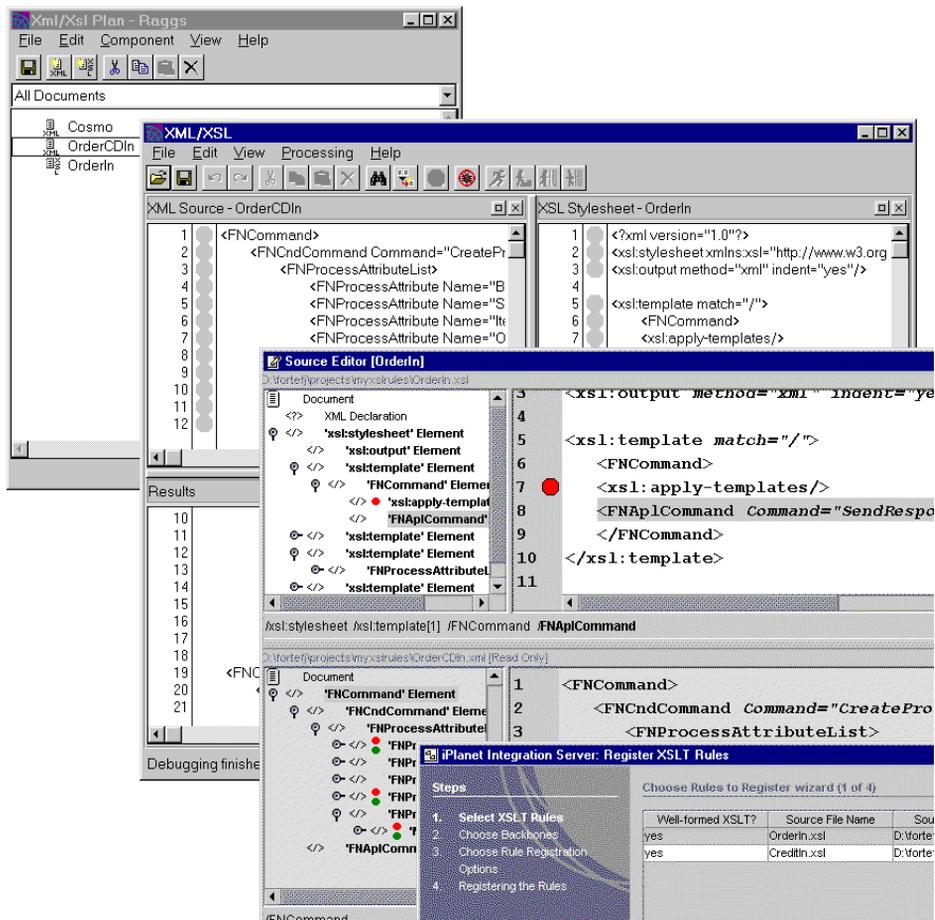
XML Messages

XML messages between an iIS proxy and its partner applications can be transported over HTTP or by using implementations of the Java Message Service. Secure transmission over HTTP is available using SSL. Additionally, application services that are made available through SOAP can be accessed by application proxies.

XML/XSLT Workshops

iIS provides XML/XSLT Workshops that facilitate the development, testing, debugging, storage, and management of XML documents and XSL stylesheets that are used for message transformation between applications.

Figure 2 XML/XSLT Workshops



XSL Stylesheets

XSL stylesheets (not pictured in [Figure 1](#)) contain XSL rules that are assigned to a proxy for decoding and encoding XML documents exchanged in messages with partner applications. By writing XSLT rules and adding them to the stylesheet base, you can define the function and behavior of a proxy.

XSL Processor

The XSL processor transforms XML messages, according to rules in the XSL stylesheet base, into documents that can be used by a proxy to coordinate a business process with the process engine or to communicate with partner applications.

Backbone Management Utility

iIS contains the Fusion Script utility for configuring, starting, and managing an iIS backbone system and its proxies.

iIS Process System

An iIS process system, pictured in [Figure 1 on page 10](#), supports the execution and management of business processes, coordinating the activities in the process with external applications. The heart of this system is the iIS process engine, which coordinates the work of different resources and applications that participate in the processes and that can be distributed across different platforms and systems.

iIS customers use the iIS process system to:

- develop process logic using the process development workshops
- manage sessions and processes, and the engine itself, using the Process Engine Console and other tools
- build *process client applications* that make direct API calls to the process engine using the process client APIs (C/C++, CORBA/IIOP, JavaBeans, ActiveX, and TOOL)

Process Engine

The iIS process engine executes and manages process definitions created with the Process Development Workshops. The process engine interacts with application proxies in a backbone system to coordinate the activities of a process with external applications. The process engine also interacts directly with process client applications (applications created using a process client API).

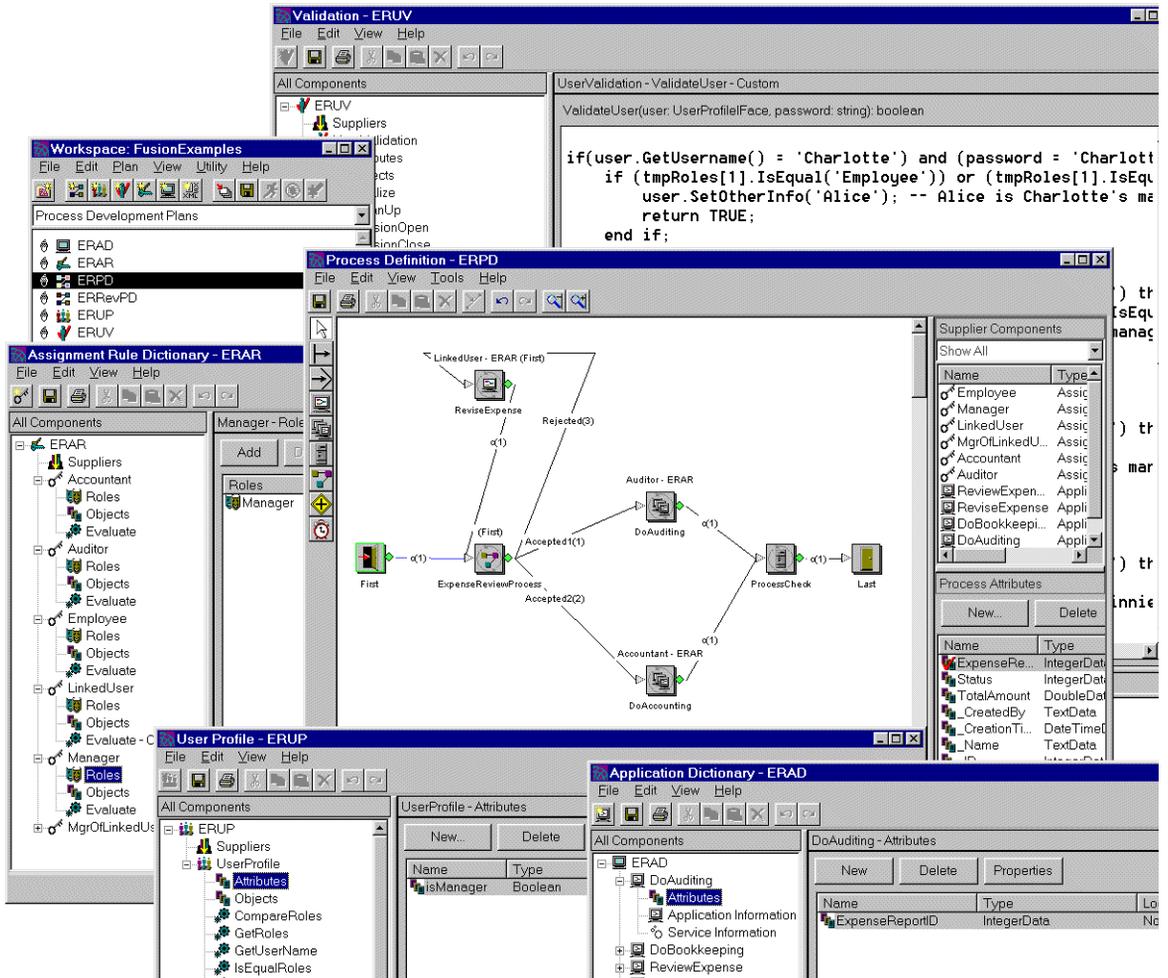
A process engine maintains an engine database containing state information on processes that it executes. The engine can be configured for failover and recovery, using information from the engine database as well as logging and history information to restore the state of the engine.

Process Development Workshops

The Process Development Workshops allow you to interactively define and edit business processes, which you then register with a process engine. Tasks you perform as you define the process logic for an enterprise system include:

- Defining work activities for the process
- Specifying process flow by routing work between activities as the activities are completed (or aborted)
- Creating and setting timers to automatically trigger activities and events
- Specifying user profiles and validation rules
- Incorporating assignment rules for activities defined in the process logic
- Defining process attributes that control the state of the process across the system

Figure 3 Process Development Workshops



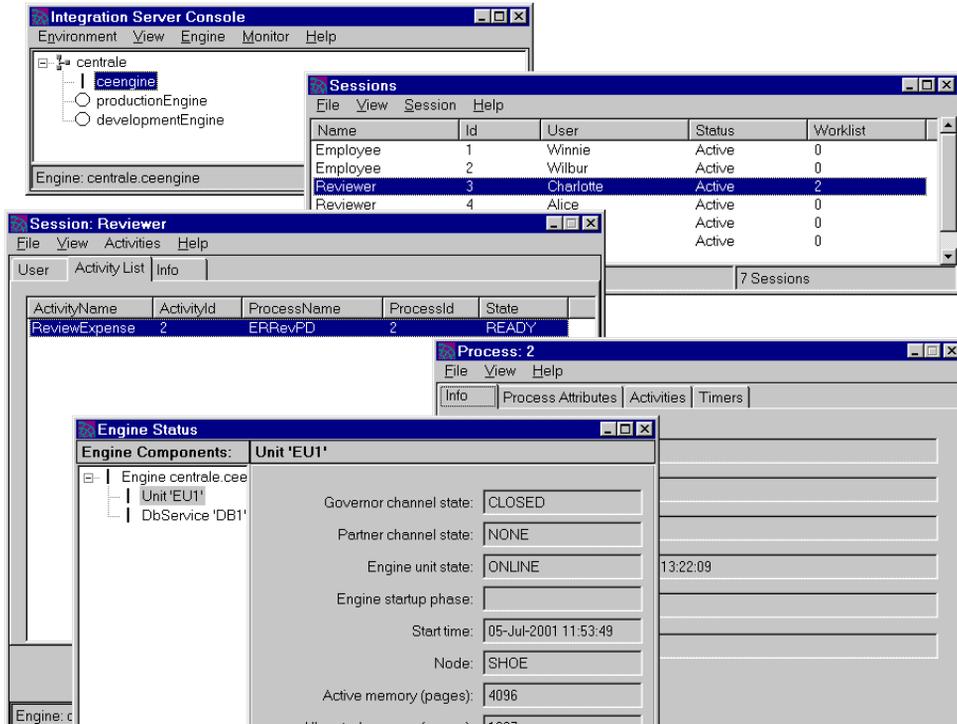
Process Management Tools

iIS provides a suite of tools for distributed system management of a process system, including configuring, starting, and shutting down process engines. Highlights of system management include:

- registering and executing processes
- monitoring and managing process execution
- maintaining application sessions with the process engine

- handling failover and recovery
- maintaining history logs of process execution

Figure 4 Process Management Tools



Process Management Utility

Most process management tasks can be handled interactively, but iIS also provides a scripting utility (Conductor Script) that can be used to automate all management tasks.

Process Client APIs

Programming APIs that allow you to build process client applications that directly interface to a process engine. APIs are available to build C++, CORBA/IIOP, ActiveX, and TOOL process client applications.

Professional and Enterprise Editions

iIS is available in either the Professional Edition or Enterprise Edition. *iIS Professional Edition* consists of the backbone system and related development tools. *iIS Enterprise Edition* contains a complete backbone system plus the process system and additional development and process management tools.

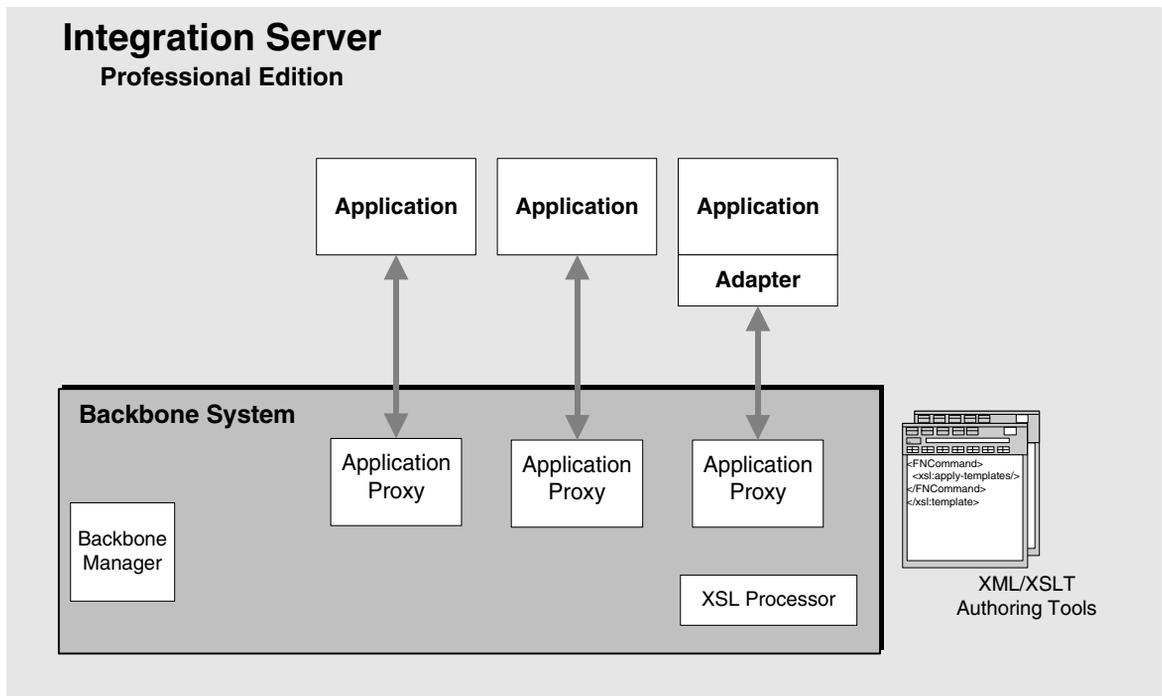
iIS Professional Edition

iIS Professional Edition enables XML messaging between enterprise applications. Messages containing XML documents can be sent over HTTP and SSL protocols or by using an implementation of the Java Message Service.

iIS Professional does not include a process system, and thus cannot access the business process automation features available with the process engine and the process development workshops.

Figure 5 shows the architecture and components of the iIS Professional Edition.

Figure 5 iPlanet Integration Server, Professional Edition



The iIS Professional Edition contains the following components:

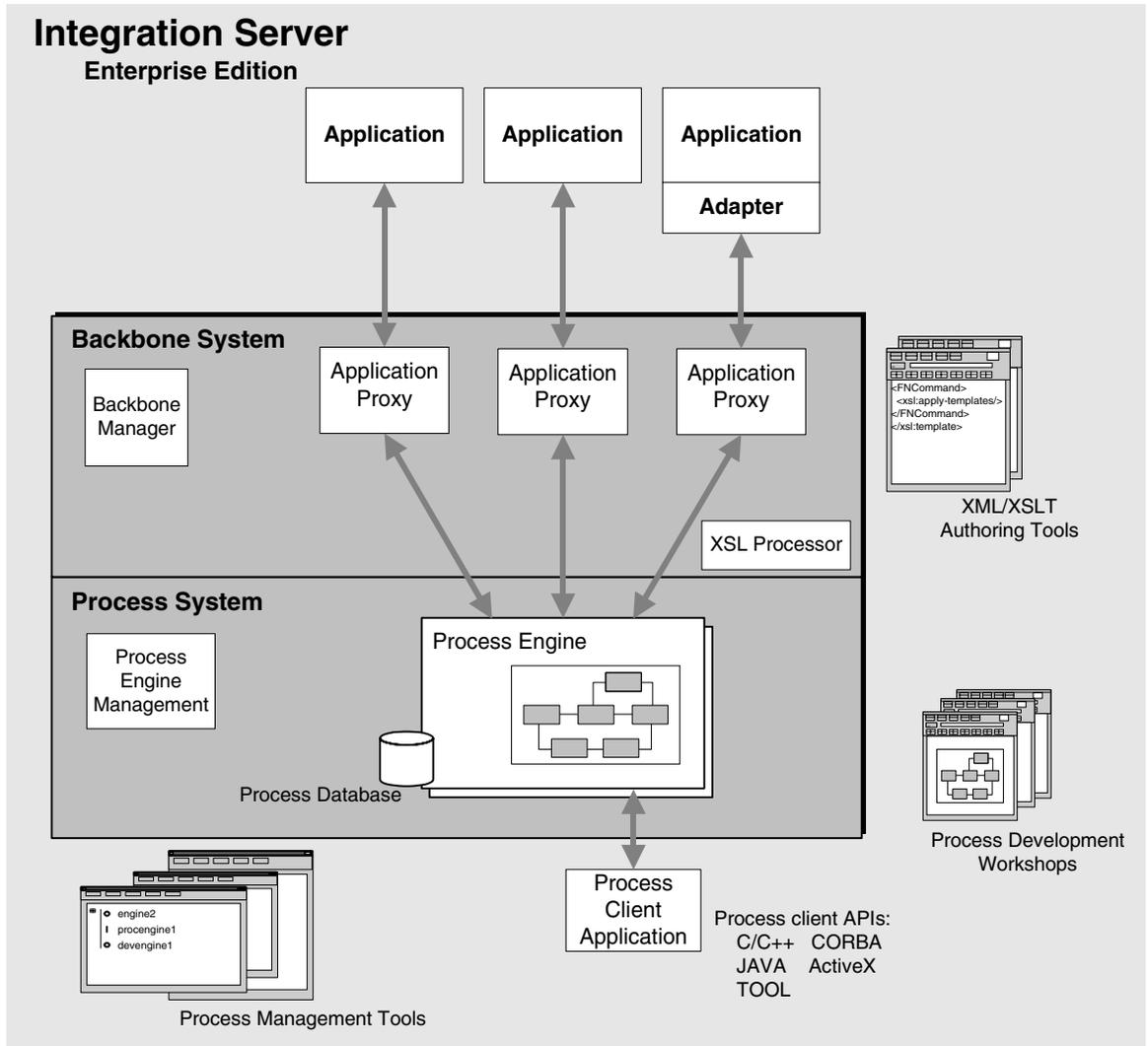
- Application proxies
- XSL Stylesheet Base (not pictured in [Figure 5](#))
- XML/XSLT Workshops
- XSL Processor
- Backbone manager

iIS Enterprise Edition

iIS Enterprise Edition includes all the features of iIS Professional Edition plus a complete process system for defining and managing the automation of business processes. Application proxies in iIS Enterprise Edition can be configured to access application services that are available through SOAP. (Access to SOAP services is not available with iIS Professional Edition.)

[Figure 6](#) shows the architecture and components of the iIS Enterprise Edition.

Figure 6 iPlanet Integration Server, Enterprise Edition



iIS Enterprise Edition includes a complete backbone system, as described in “[iIS Professional Edition](#)” on page 17, plus the following components of a process system:

- Process Development Workshops
- Process Engine
- System Management Tools
- Process Client APIs

iPlanet UDS Environment

iPlanet Integration Server is not a standalone system, but is installed and run from within an iPlanet UDS environment. iPlanet UDS provides a robust distributed application development environment that allows you to create, manage, and run iIS across various nodes in the UDS environment. The UDS central repository provides the base classes for iIS and supports team development of iIS process definitions and client applications.

Before installing and configuring an iIS system, you must first install and configure iPlanet UDS.

About iPlanet UDS

iPlanet UDS is a software environment for developing, deploying, and managing distributed applications on multiple platforms. It supports the full life cycle of a UDS application from development to the management of the application as it runs on a number of platforms. iIS is itself a UDS application that runs within a UDS environment.

An iPlanet UDS environment includes software that operates at a number of different levels, including a runtime environment for UDS applications, the development and management of UDS applications, and management of a UDS system.

UDS System Management Services

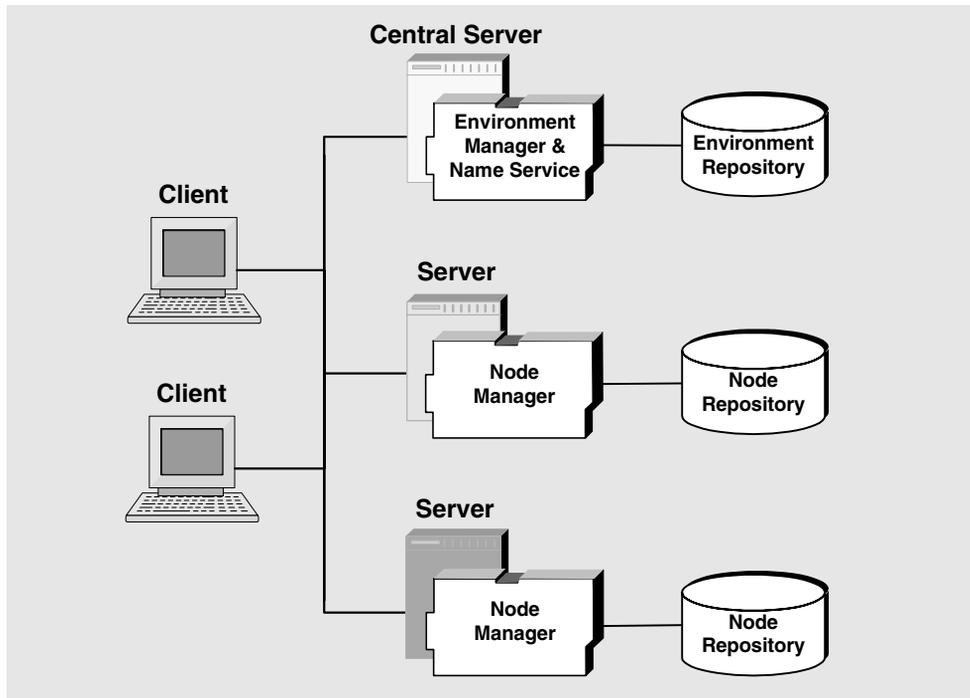
A UDS environment contains the following system management services, which can be installed on various nodes in your system:

- Environment Manager
Governs the UDS environment, supporting communication among the nodes in the environment.
- Node Manager
Each node in a UDS environment runs a node manager process to communicate environment-specific information to the Environment Manager.
- Repository Service
The repository service stores development projects and their components.

Nodes in a UDS System

During installation of UDS, you specify the role of a node within a UDS system. Later, during iIS installation, you can distribute iIS functions across the different nodes in your UDS environment. **Figure 7** shows the relationship of nodes in a UDS environment.

Figure 7 Nodes in an iPlanet UDS Environment



UDS supports the following types of nodes:

- Central Server Node

Hosts the Environment Manager process, and typically the repository server process. Each UDS environment must contain a single central server node. All other nodes in a UDS system rely upon the identity of the central server node.

A central server node can participate fully in development, distribution, and deployment of iIS applications. However, you may want to designate some of these functions to other nodes in your environment.

- Server Node

A server node can participate fully in development, distribution, and deployment of iIS applications. However, it contains a subset of the software for a central server node. A server node relies on the environment manager of the central server node to communicate with other nodes in the system.

- Development Node

Can be used for development of iIS applications. Any node, with the exception of a runtime client, can be used for development. During iIS installation, installing the process development workshops on a node makes that node a development node.

- Runtime Client Node

A client node that does not contain any project development components, and thus cannot develop iIS applications. A runtime client node can run iIS applications (exclusive of the development workshops).

Each node in a UDS system contains a UDS runtime in addition to other components specific to that node. However, a runtime client contains only the UDS runtime.

Configuring an iIS System

When you install iIS, you can install the complete system on a central server node. However, you typically distribute components of a system across various nodes in your UDS environment.

Table 1 Distribution of iIS Components in a UDS system

iIS Component	Node	Description
Process Engine	Central Server Server	A node hosting the process engine. Typically, this node also hosts the engine database. An iIS system can host multiple engines across different nodes.
Backbone	Central Server Server Runtime	An iIS backbone must be installed on top of a UDS runtime, which is available from any UDS node.
Process Development	Central Server Server	Process Development Workshops for creating and editing process definitions.
Repository Service	Central Server Server	There can be only a single central repository, which is typically hosted by the central server. However, a server node can also host the central repository.
Process system management tools	Central Server Server	Interactive and scripting tools that manage a process engine. System management tools can be distributed across various nodes in a system.
Process client development	Central Server Server	Development of process clients can be distributed across various nodes in your system.
Process client deployment	Central Server Server Runtime	Process clients can be deployed on runtime only nodes.

Documentation Roadmap

As described earlier in this document, an iIS development system contains a number of components, some for development, some for runtime execution, and some for both. An iIS project team consists of individuals who work with different components of the system.

Table 2 describes very briefly the roles of different individuals in the development and deployment of an iIS system, with pointers to the relevant documentation for each role. Often, individuals on an iIS project team have overlapping roles. Refer to the *iIS Process Development Guide* for a more complete discussion of the roles of individuals in an iIS system.

The documentation listed in **Table 2** is a starting point to for each role. In most cases, a thorough reading of additional relevant material is recommended. Refer to the Preface in any iIS manual for information on installing and using iIS documentation.

Table 2 Documentation Roadmap for Members of an iIS Development Team

Role	Description	Documentation Starting Point
All team members	—	<ul style="list-style-type: none"> <i>iIS Conceptual Overview</i> (This document)
Application system designer	Designs the overall enterprise application. Coordinates the work of developers and integrators.	<ul style="list-style-type: none"> <i>iIS Process Development Guide</i> Chapter 1: Fundamentals <i>iIS Process System Guide</i> Chapter 1: Introduction: iIS Process Management
Process developer	Creates process definitions, following guidelines provided by the application system designer.	<ul style="list-style-type: none"> <i>iIS Process Development Guide</i> <i>iIS Process System Guide</i> Chapter 1: Introduction: iIS Process Management iIS Online Help for Process Development Workshops

Table 2 Documentation Roadmap for Members of an iIS Development Team *(Continued)*

Role	Description	Documentation Starting Point
Application integrators	Integrates packages and legacy applications for use with an iIS Backbone, including development of XSL stylesheets and integration of adapters, if needed.	<ul style="list-style-type: none"> • <i>iIS Backbone System Guide</i> • <i>iIS Backbone Integration Guide</i> • <i>iIS Process Development Guide</i> Chapter 1: Fundamentals • <i>iIS Process System Guide</i> Chapter 1: Introduction: iIS Process Management • <i>iIS Process Client Programming Guide</i> Chapter 1: Building an iIS Process Client Application • iIS Online Help (topics for iIS Backbone)
Application developer	Creates iIS process client applications, following guidelines provided by the application system designer.	<ul style="list-style-type: none"> • <i>iIS Process Client Programming Guide</i> Chapter 1: Building an iIS Process Client Application Subsequent chapters, as needed • iIS Online Help (Process client API reference)
System manager (Backbone)	Installs and manages an iIS Backbone.	<ul style="list-style-type: none"> • <i>iIS Installation Guide</i> • <i>iIS Backbone System Guide</i> • <i>iIS Process System Guide</i> Chapter 1: Introduction: iIS Process Management • <i>iIS Backbone Integration Guide</i> • Relevant iPlanet UDS documentation for installing and managing a UDS system
System manager (Process Engine)	Installs and manages an iIS process system.	<ul style="list-style-type: none"> • <i>iIS Installation Guide</i> • <i>iIS Process System Guide</i> • <i>iIS Process Development Guide</i> Chapter 1: Fundamentals • <i>iIS Process Client Programming Guide</i> Chapter 1: Building an iIS Process Client Application • iIS Online Help for Process System Management • Relevant iPlanet UDS documentation for installing and managing a UDS system

iIS Glossary

This glossary provides information about terms and concepts you might encounter while using the iPlanet Integration Server (iIS) product and documentation.

Terms in *italics* within definitions are related terms that also are defined in the glossary.

adapter An optional iIS component that provides an XML/HTTP interface for a participating *application* that is not *XML-enabled*. The adapter converts arriving XML documents into the application's native API; it also converts the application's API calls into XML documents understood by the *proxy*.

application A participant in an *iIS enterprise application*. Two kinds of applications can participate in the business process: a *process client* application, which establishes a direct session with a *process engine* and an *XML-enabled application*, which communicates with the engine through a *proxy*.

application document The XML *proxy document* sent by an *application* to its *proxy* or from the proxy to the application.

An application document can contain any *XML* that the application understands. XSL *stylesheets* that you create transform this XML into a version that the proxy understands.

application proxy See *proxy*.

authentication document A *proxy document* used for *session authentication*. Either a *proxy* or an *application* can transmit an authentication document in response to a request for user authentication.

backbone A set of related XML-based *proxies* associated with a *backbone manager*, XSL *stylesheet base*, and optional *adapters*. The backbone enables an associated *process engine* to command and respond to *applications* in fulfillment of a business process, or can be used solely for message brokering and data transformation.

backbone manager The component of the *backbone* responsible for starting and stopping *proxies* and controlling dynamic changes to proxy configurations.

command document A *proxy document* that a proxy generates to send commands to the *process engine* (for example, to start an activity) or to send a message to an application (for example, to inform the application that an activity has been completed).

command processor The component of the *proxy* that interprets *command documents* and generates messages regarding the commands to be performed by the *process engine* and the application.

document element The topmost visible element in the hierarchy of an XML document. The document is the only child of the *root element*. An XML document can have only one document element.

Enterprise Edition See *iIS Enterprise Edition*.

FNscript The script utility for configuring, monitoring and managing *backbone* components. Also referred to as Fusion Script.

Fusion See *iPlanet Integration Server*.

HTTP Hypertext Transport Protocol, one of the communications protocols that *iPlanet Integration Server* uses to transfer messages among *applications*, *proxies*, and the *process engine*.

iPlanet Integration Server A business process integration suite that encompasses a *process management system* and an XML *backbone*.

iIS has two licensed versions, *iIS Enterprise Edition* and *iIS Professional Edition*. Prior to Version 3.0, iIS was marketed as *Fusion*.

iIS backbone See *backbone*.

iIS enterprise application A complete *iPlanet Integration Server* solution that provides process-driven application integration across the enterprise.

iIS Enterprise Edition A licensed version of *iPlanet Integration Server* that contains the complete functionality of the product. Compare to *iIS Professional Edition*.

iIS process engine See *process engine*.

iIS Professional Edition A licensed version of *iPlanet Integration Server* that contains a subset of the components available with *iIS Enterprise Edition*.

iIS Professional Edition provides a *backbone* and XML/XSL authoring tools. The backbone is used solely for message brokering and data transformation, and does not communicate with a *process engine*.

iIS process management system See *process management system*.

inbound Refers to any communication that flows from an *application* to a *proxy*, for example an inbound session.

An inbound *stylesheet* transforms the application's XML into XML that the proxy understands.

independent proxy An *independent proxy* is used only for message brokering and data transformation between integrated applications; it does not interact with a *process engine*. See also *proxy*, *process-based proxy*.

interactive application An *application* that requires input or instructions from a human.

interface An interface defines a set of class elements, without providing the code that implements them. The code for the methods and event handlers in the interface is provided by the classes that implement the interface.

JMS Java Message Service, one of the protocols that *iPlanet Integration Server* uses to transfer messages among *applications*, *proxies*, and the *process engine*.

message broker The component of the proxy that is responsible for routing messages.

outbound Refers to any communication that flows from a *proxy* to an *application*, for example an outbound session.

An outbound *stylesheet* transforms the proxy's XML into XML that the application understands.

process client An *application* coded with a process client API or *process client interface* that makes calls directly to the *process engine*.

process client interface An interface or programming language API for constructing a *process client* application. The supplied interfaces are iPlanet UDS (TOOL), JavaBeans, ActiveX, C++, and IIOP.

process engine The *iPlanet Integration Server* component that drives and manages business processes.

process development workshops A suite of iPlanet UDS-based development tools used to create process definitions for use within a *process management system*.

process management system The business process management component of iIS, including the *process engine*, the *process development workshops*, and the *process client interfaces*.

process-based proxy A *process-based proxy* interoperates with a *process engine* on behalf of an integrated application that participates in a business process managed by the engine. See also *proxy, independent proxy*.

Professional Edition See *iIS Professional Edition*.

proxy A component of a *backbone* that routes and transforms XML documents. A proxy can be *process-based* or *independent* and can use HTTP or JMS for messaging.

proxy document An XML document that is processed by a *proxy*. Proxy documents specify the operation of a *backbone* by controlling message flow between the proxy and participating *applications*, as well as between the proxy and the *process engine*.

There are four types of proxy documents: *application documents, authentication documents, command documents, and state documents*.

robotic application An *application* that requires no human interaction to participate in the business process.

root element The abstract element implied as the topmost element of the hierarchy of an XML document. The root element is denoted by “/”.

The root element can have only one child element, which is called the *document element*.

service provider A *robotic application* or *interactive application* that performs an activity in the fulfillment of a business process. Service providers generally are associated with offered activities. A service provider acts as a server to its *proxy*.

service requestor A *robotic application* or *interactive application* that requests the creation of a business process. Service requestors generally are associated with the creation of iIS process instances. A service requestor acts as a client to its *proxy*.

session authentication The means by which an HTTP *proxy* or *application* ensures that a user requesting access is authorized to obtain such access.

HTTP session authentication can be implemented by simply sending the username and password to the proxy or application or by sending an *authentication document* containing additional user information.

SOAP Simple Object Access Protocol, one of the protocols that *iPlanet Integration Server* uses to transfer messages among *applications*, *proxies*, and the *process engine*.

state document A *proxy document* generated by the *process engine* to communicate changes of activity state to the proxy (for example, information that a specific activity has started).

stylesheet An *XML* document that provides rules for transforming a source *proxy document* (for example an *application document*) into a results proxy document (for example, a *command document*).

stylesheet base A set of XSL *stylesheets* registered with a specific *backbone*. You associate stylesheets from the stylesheet base with specific *proxies* on that backbone.

template Instructions within an XSL *stylesheet* that define how an item in an XML source document should be rendered in the XML results document.

transformation The process of applying a *stylesheet* to an XML source document to render it, generally in a different form, in an XML results document.

XML The Extensible Markup Language used in *proxy documents* that provides a standard communication medium between *proxies* and *applications*, and between proxies and the *process engine*.

XML/XSL NetBeans modules Graphical tools, available as NetBeans plug-ins, that provide for authoring, testing, and debugging XSL *stylesheets*.

The XML/XSL NetBeans modules include an XSLT Authoring Tool and an XML/XSL Editor and Debugger. These NetBeans modules provide a richer set of functionality than the tools available with iPlanet UDS.

XML/XSL plan A collection of XML documents and XSL *stylesheets* that you create in the *XML/XSL Plan Workshop* for testing and debugging.

XML/XSL Plan Workshop A graphical tool based in iPlanet UDS that *iPlanet Integration Server* provides for creating and managing XML documents and XSL *stylesheets*.

XML/XSL Workshop A graphical tool based in iPlanet UDS that *iPlanet Integration Server* provides for testing and debugging XSL *stylesheets*. Compare to the *XML/XSL NetBeans modules*.

XML-enabled application An *application* that can generate and understand *XML proxy documents* when communicating with its *proxy*.

If an application is not XML-enabled, it must use an *adapter* to translate between its native API and XML.

XPath The language you use within an XSL *stylesheet* to locate an item in the XML source document that you want to transform in the XML results document.

XSL The Extensible Stylesheet Language used to transform XML *proxy documents* into other XML proxy documents. *XSLT* is a subset of *XSL*.

XSL processor The component of a *proxy* that applies an *XSL stylesheet* to a source *proxy document* to transform it into a different type of proxy document.

XSLT Extensible Stylesheet Language Transformations, the subset of *XSL* that provides a means for transforming items from an XML source document to an XML results document, for example, to change an element into a different element.

Index

A

adapter
about 11

C

configuring 24

D

Documentation Roadmap 25

I

iIS
architecture 9
conceptual overview 7
documentation roadmap 25
Enterprise Edition 18
features 8
Professional Edition 17
UDS environment 21

iIS backbone system
overview 9
iIS process management system
overview 9
iIS system 24
UDS environment 21

P

proxies
overview 11

U

UDS environment
overview 21

