

Building Custom Applications with the Oracle Internet File System Content Management Software Development Kit

*An Oracle Technical White Paper
June, 2002*

Building Custom Applications with the Oracle Content Management Software Development Kit

INTRODUCTION

The Oracle Content Management SDK allows you to build applications for managing content across the enterprise. Most business applications need to manage all types of information -- not only structured data, but unstructured data like documents, messages, and web site content. For example, customer support systems need to not only maintain records of customer support requests along but also manage the production of technical documentation that is used to resolve customer's issues. Technical publishing applications are needed to manage the complex processes for developing and releasing product documentation. Sales automation systems need to manage not only customer account information and call logs, but also discussions held with the customer in the form of electronic message threads, Requests for Information received from the customer, and Bids submitted to the customer. On-line product catalogs need to include product images and collateral. To build these applications, you need a robust content management platform that stores all types of information, performs and scales to meet the demands of large-scale deployments, makes the information easy to access from a standard desktop environment, provides services for organizing, securing, processing, and managing the life-cycle of the information, provides frameworks for automating business processes, and integrates easily with other information systems.

Oracle 9i delivers the qualities you need in a content management platform with the Oracle Content Management SDK. You can now leverage the performance, scalability, and reliability of the Oracle 9i database to manage all types of information. With the Oracle Content Management SDK, Oracle 9i provides robust content management services (such as versioning, access control, and categorization), that you can leverage to implement application functionality for organizing, securing, and managing the life-cycle of information instead of coding the functionality from scratch. The Oracle Content Management SDK provides easy to use frameworks for parsing and rendering content stored in the database, automating business tasks on the server, and integrating with external information systems. The Oracle Content Management SDK makes information in the

database accessible from common desktop tools (for example., Microsoft Windows Explorer, Microsoft Office, and Netscape Communicator) from various operating systems (for example, Microsoft Windows, NT, UNIX, and **Mac OS X**) via standard protocols (for example, SMB, WebDAV, NFS, and AFP). Using its Java API, you can also build Java clients and web user interfaces that are tailored to your business.

In this paper, we will look at the how these elements are used to create applications, and the paradigm for generating the application components.

FILE SERVICES

With no customization, the Oracle Content Management SDK platform allows you to store and access any type of information in the database with common desktop tools on various operating systems via standard protocols. The protocol servers make storing your information in the database as easy as working with a typical file system. This functionality alone may be enough to meet your business needs as a first phase of deployment. For example, your Documentation organization could continue use their preferred technical publishing tools, like Adobe FrameMaker, to author documentation that is stored in the Oracle 9i database via SMB. The documentation can be organized into folder hierarchies that are shared by the entire organization just like a shared file server. After this first phase of deployment, you could start customizing the system to implement business-specific functionality, as discussed in the [Creating Custom Content Management Applications](#) section.

Store Any Content

The Oracle Content Management SDK allows you to store any type of information in a single repository – the Oracle 9i database. Now, the database can be used to store information comprised of structured data, like product information and project records, as well as unstructured content like word processing documents, spreadsheets, images, audio files, and video files. Once your information is stored in the database, you can leverage Oracle 9i's relational data management, indexing, retrieval, and administration capabilities. Structured and unstructured data components are stored in relational database tables. The data is indexed for quick retrieval using Oracle's **b-tree** indices for structured data, Oracle Text for text, and Oracle interMedia for audio, video, and image files. Administrators can use Oracle 9i's administration tools to secure, backup and restore information in a single place without having to worry about disrupting the referential integrity of the information.

The Oracle Content Management SDK allows you to manage information in an object-oriented manner using Java. Each piece of information, be it comprised of multiple structured data fields and/or LOBs, is represented as a single Java object. Users can save, open, edit and delete these objects just like files in a file system. Users can simply search for documents, without having to know how the

documents' content and meta-data is stored in the Oracle *9i* database. The Oracle Content Management SDK handles the complexity of mapping the Java objects to the relational data storage schema.

Users can share access to information stored in the database just as they would a shared file server. They can organize the repository into project work areas using folder hierarchies according to how each project works with the information. By storing the information in a central location, users can maintain a single point of truth. As multiple authors make modifications to the information, they can ensure that everyone is working off of the most recent iteration.

Access Through Standard Protocols Using Common Desktop Tools

The Oracle Content Management SDK feature of Oracle *9i* provides a set of standard protocol servers that allow users to access content stored in the database via common desktop tools on a variety of operating systems. The protocol servers include:

- SMB and NTFS for Microsoft Windows and NT desktops
- NFS for UNIX desktops
- AFP for **Mac OS X** desktops
- HTTP/WebDAV for web clients
- IMAP4 and SMTP for e-mail clients
- FTP for standard file transfer clients like WS_FTP Pro

No special client is needed to access files in the database. Any Windows, UNIX, or Apple compatible computer can access content in the database immediately after it is set up and connected to a network. Users can immediately begin taking advantage of the file and print services through familiar clients without learning a new interface or changing the way they work.

Administrators can also access information using the **Commandline** Utility Protocol (CUP). Besides performing ad hoc operations and queries against the database, CUP provides support for XML configuration files, enabling an administrator to upload XML definitions used to create objects in the database.

CREATING CUSTOM CONTENT MANAGEMENT APPLICATIONS

You can build end-to-end custom user interfaces using the Oracle Content Management SDK. For example, you may want to build a custom client that is tailored for processing claims, managing customer accounts or managing a supply chain. The Oracle Content Management SDK includes a Java API that you can use to build custom Java clients or web user interfaces. Via the Java API, these applications can leverage the content management functionality discussed in the [Content Management Features](#) section.

Since the Oracle Content Management API is 100% Java, you can use standard Internet technologies to build your custom applications, including Java servlets and Java Server Pages.

Servlets

A servlet is a Java application that runs on the Oracle 9i Application Server. Servlets handle requests from web browsers and respond by returning an HTML page. Essentially, they are Java programs with embedded HTML code. You configure the Oracle 9i Application Server to watch for a particular URL, and it will respond to the request by running your servlet. Your servlets can then call the Oracle Content Management SDK to access, manipulate, and return information from the database.

Java Server Pages

Java Server Pages (JSPs) are HTML pages with embedded Java code. JSPs are similar to servlets -- in fact, when a JSP is compiled, it is turned into a servlet. The HTML pages determine the layout of your web application. The embedded Java code calls Java beans that access and manipulate information in the database via the Oracle Content Management SDK.

CONTENT MANAGEMENT FEATURES

Once you've chosen a medium for creating a user interface, you can begin leveraging the Oracle Content Management SDK to implement content management functionality in your application. The Content Management SDK includes built-in functionality for:

- [Organizing information](#)
- [Parsing and rendering information](#)
- [Controlling access to information](#)
- [Searching for information](#)
- [Managing collaboration on information](#)
- [Automating business processes](#)
- [Integrating with other information systems](#)

Rather than building this functionality from scratch, you can accelerate application development by leveraging the capabilities of the Oracle Content Management SDK.

Organizing Information

The Oracle Content Management SDK allows you to organize information stored

in the database in three ways:

- [Subclassing](#)
- [Foldering](#)
- [Categorization](#)

Subclassing

Subclassing is a way for you to differentiate types of information that are stored in the database.

On install, the Oracle Content Management SDK is pre-configured to manage common types of information, including documents, messages, and folders. The Oracle Content Management SDK organizes different types of information into a class hierarchy. Each class inherits the attributes and behaviors of its superclass in the hierarchy, and introduces new attributes and behaviors that are specific to the class.

You can extend the class hierarchy to define your own types of information. For example, you might subclass the Document class to define special types of file, like *Books*, *Presentations*, and *Web Pages*. Or, you might subclass the Folder class to define special types of container, like *Project Records* and *Sales Kits*.

When you subclass, you can define both custom attributes and functionality for that type of information. For example, you could track extra attributes for Presentations, such as *Presenter*, *Presentation Date*, and *Forum*. You could automatically default the value of the Forum attribute and constrain the Presentation Date to a specific time period. Or, you may want to respond differently to changes to web pages than you would to changes in other files by sending a notification to the web site administrator when a page is edited or deleted.

Foldering

You can also organize all types of information into hierarchical folder structures. Folder hierarchies can be used to create personal or shared work areas. Documents, Presentations, Sales Kits, and web pages can be organized according to how to how users work with one another. Users can browse through the folder hierarchies to locate information in the database via standard tools, such as Microsoft Windows Explorer.

The Oracle Content Management SDK implements foldering using a referential relationship model. While each document, web page, and sales kit is stored once in the database, they can be referenced by multiple folders. This gives users the flexibility to organize the same set of information with different folder hierarchies that are tailored for different working groups and user roles. For example, if three people were collaborating on the *XYZ Sales Proposal*, one might store the file in a

folder named *XYZ*, the second in a folder named *Sales*, and the third in a folder named *Proposals*. All of the users would be looking at the same file, and any edits would be available to all other users immediately after the changes were saved. But all of them would be able to organize the file in a way that makes it easy for them to find it when they need it.

You can extend the referential relationship model to manage other types of relationships. For example, you could define a custom type of relationship for managing links between a compound document (for example, a book) and its components. The custom relationship could store meta-data that is pertinent to the documents only in the context of their relationship, such as component type (for example, chapter, section, appendix).

Categorization

You can also categorize different types of information according to how the information is used in your business. For example, you could define categories like *Marketing Collateral*, *Product Standards*, and *Project Information*, then associate folders, documents, web pages, and presentations with one or more category. You could also add extra meta-data to the information that is relevant to its category. For example, the category Project Information could apply the extra attributes *Project Name* and *Project Number*.

Parsing and Rendering Information

The Oracle Content Management SDK enables you to automatically process information stored in the database with parsers and renderers. You can automatically extract attribute values or burst compound documents by parsing a document's content. You can use renderers to convert documents into different formats and store the output as renditions, or use stylesheets to dynamically determine the layout of XML files when they are retrieved from the database.

Parsers

A great deal of critical data is embedded in the unstructured text of documents, making it difficult to mine and reuse. Parsers allow you to extract data elements so you can use them to index and search for your documents or assemble new documents.

For example, you can use a parser to extract the abstract from a PDF file, the title from an HTML file, or line items within an XML-based purchase order. These data elements then become attributes of the document, making them available for editing and searching as you would the Name, Description and other document attributes. You can also treat the information in a relational way to create queries and reports.

A parser can also burst a document into sections and store them as components of a compound document. Breaking a document down in

this manner allows you to reuse core information assets to assemble multiple compound documents. Each information asset can be stored and maintained in a single place, and be automatically updated in each compound document that includes it.

A custom parser receives the document content as an `InputStream` or `Reader`. Your parser is free to perform any processing you choose – extract information, change values, make calculations, perform validation, etc. When finished, the parser returns the data structures required to create the resulting documents in the repository.

Renderers

Once information has been stored as an object in the database, you can use the Oracle Content Management SDK to dynamically render the object in a variety of formats and layouts. For example, you might use a renderer to convert a word processing document to PDF format so reviewers can annotate the document. You could then convert the same document to HTML so it can be easily viewed with a Web browser or to WML so that it can be viewed from a mobile device.

A custom renderer receives an object from the database and outputs the converted content as an `InputStream` or a `Reader`. The renderer is free to process the object as required (for example, retrieving a stylesheet to manipulate the layout, filter out data, and personalize the content according to the user's profile).

The Oracle Content Management SDK provides customization frameworks that make it easy for your application to call custom parsers and renderers. The `Override` framework allows you to call a custom parser or renderer within the transaction of certain operations performed by the server – `insert`, `update`, `free`, `addRelationship`, and `removeRelationship`. For example, you may want to automatically extract attribute values from the content of documents when they are first inserted into the database. To do so, you can override the `insert` operation to call your custom parser after the document is inserted. If your parser throws an exception, the entire transaction can be aborted. The `Agents` framework allows you to call custom parsers and renderers when certain events occur in the repository, or to batch process information during a designated time period. For example, you can create an agent that renders batches of documents to HTML during off-peak hours, and stores the output as renditions. Subsequently, application users can opt to view the documents natively in their Web browser instead of opening the word processor used to create the documents. The `Agents` and `Overrides` frameworks are discussed more in the **Error! Bookmark not defined.** section.

Controlling Access to Information

The Oracle Content Management SDK provides a robust security model that you can use to control how information is accessed in the database. The security model provides features that make it easy to apply broad security levels to your information, but also gives you the power to manage complex access permissions on an object-by-object basis.

The Oracle Content Management SDK uses Access Control Lists (ACLs) to secure information. An ACL is a list of users and/or groups who have been granted or revoked access permissions. You can grant and revoke permissions like discover, read, modify attributes, modify content, and delete for fine-grained access control. If you want to manage access more broadly, you can create PermissionBundles that represent sets of permissions – such as consumer rights, author rights, and administrator rights -- that can be granted or revoked together. You can also define custom permissions that are specific to your business. For example, you could create a permission for viewing the HTML rendition of a document. Then, although a consumer may not be granted the permission to access the original document, he could be granted this permission to view a read-only format.

ACLs can be used to define the access permissions that are appropriate for different security levels of information. For example, you could create an ACL for company confidential information. The ACL can specify which employees, contractors, and partners have the permission to author, delete and view company confidential information. Then, the ACL can be applied to all documents and folders that are confidential to the company. In this way, you can control how company confidential information is accessed in a single place. Changes made to the ACL will become immediately effective on the information in that security level.

If your application requires more complex access control, you can use Composite ACLs. With Composite ACLs, you can apply multiple Access Control Lists to your information. For example, you may want to grant access to documents based on the folders in which they are placed. Each folder may have a different Access Control List applied to it. If a document is placed in two folders -- a Marketing folder whose ACL grants full access to members of the marketing organization, and a Sales folder whose ACL grants access to the sales organization -- both folder ACLs may be used to control access to the document. To accomplish this, your custom application would override the addRelationship operation on folders to automatically add the folder's ACL to a Composite ACL on the document.

Searching for Information

Now that you have stored all types of business information in the Oracle 9i database, you can leverage the Oracle Content Management SDK to provide a faster and more powerful search interface than feasible if it were stored across

disparate file servers.

The Oracle Content Management SDK makes your information searchable with a wide range of criteria, including:

- **Class** Users can search for information based on its class, and class attributes. For example, they could search for all Presentations presented after January 2002.
- **Category** Users can search for information in a certain category, based on attributes that pertain to that category. For example, they could search for all Marketing Collateral created for the Oracle 9i campaign.
- **Folder.** Rather than searching across the entire repository, users can limit their search to a specific folder, or branch of folders.
- **Content.** Users can search for documents based on words or phrases contained in their text. The Oracle Content Management SDK leverages the Oracle Text feature of Oracle 9i to index the content of documents of over 150 formats. Oracle Text makes it possible to refine content-based searches by performing proximity searches -- where one word occurs near another. You can also expand the search to include words that sound like, are spelled like, or are related to a search term in a thesaurus. With Oracle Text's linguistic analysis capabilities, you can also search for documents based on their subject matter, or *theme*, without having to know any of the words that they contain.
- **Relationships.** Users can search for information based on relationships to other information in the repository. For example, a user could search for all documents attached to a cover letters that were sent before March 2001.

The Oracle Content Management SDK allows you to construct complex searches by combining search criteria together with Boolean operators. Since both structured and unstructured data is stored and indexed in the database, complex searches are expressed in a single query statement that is optimized by the Oracle kernel to perform much faster than would be possible in a standard file system.

Managing the Life-Cycle of Information

By creating an environment where users can share and collaborate on information, it becomes critical for your application to manage the information's life-cycle. The Oracle Content Management SDK provides functionality that you can leverage to track and manage changes made to the information.

The Oracle Content Management SDK provides built-in locking mechanism that you can use to prevent authors from overwriting each other's work, or to archive information. For example, when an author wants to edit a file, your application can require that she lock the file so other authors wont edit it at the same time. When the author is done, she would then check-in her changes as a new version

and unlock the file. When the document has been finalized, your application could lock the document so that no more versions can be created.

The Oracle Content Management SDK also provides built-in versioning functionality. Your application can use versioning to store iterative copies of documents as they are updated in the database. All versions of the document are stored in a document *Family*. By default, the most recent version of the document is displayed when a user accesses the document through one of the supported protocols. You can build a custom interface that allows users to view a history of all versions created for a document, and retrieve previous copies.

The versioning and locking mechanisms can be used to manage the life-cycle of any type of information, not just documents. For example, you may want to version Sales Kits – a custom subclass of Folder. The Oracle Content Management SDK allows you to specify if a type of information is versionable when it is defined.

The Oracle Content Management SDK also allows you to archive document content to an external storage device. Administrators can configure the repository to automatically move the content from a BLOB field in the database to a file system path that is referenced by a BFILE field in the database. All of the document's meta-data and content indexes remain in the database for fast searching.

Automating Business Processes

As long as you are providing a centralized working environment for users to collaborate on information, why not facilitate their work by automating some tasks on the server. For example, your application could automatically send an e-mail alert to a document's owner when another user changes it. Or when an author completes a draft, your application could kick off a workflow that manages the review and approval cycle.

The Oracle Content Management SDK provides frameworks that make it easy for you to automate business processes on the server: Agents, Overrides, and Advanced Queues.

Agents

The Agents framework allows you to automatically perform tasks on the server after certain events occur or during specified periods of time.

For example, you can build an agent that automatically sends an e-mail to the owner of a document when the document is changed by another user. The Oracle Content Management SDK keeps track of all events that occur in the repository. You can configure the agent to listen for any update events that occur on documents. After the event occurs, the agent can then check who made the

change, obtain the document owner's mail address and send a message to the owner's inbox.

In other cases, you may want an agent to run during a specified period of time. For example, you could build an agent that automatically deletes documents after their expiration date has passed. Rather than running the agent all the time, it can be configured to run at a specific time when user activity is less intensive.

The Agents framework allows your users to manage agents just like any other server process. In a central administration tool, administrators can monitor and configure your custom agents alongside the servers and agents that are installed with the Oracle Content Management SDK. They can configure when the agents startup and shutdown. Agents can be run on a separate middle-tier machine, allowing them to perform the task with minimal impact on system users.

Overrides

The Oracle Content Management SDK allows you to customize how certain operations are performed on the server with the Override framework. For example, you could automatically validate the format of an attribute value when a new document is inserted into the database. Or, you might automatically set the ACL on a document based on the folder in which its been placed.

The Override framework allows you to customize how the server performs common operations, including Insert, Update, Free, AddRelationship, and RemoveRelationship. The Override framework consists of a set of methods that are called before and after the operation is performed by the server. You can override these methods to include code that implements the business task. When the server performs the operation, your custom overrides will be called synchronously, executing within the transaction that performs the operation. If an exception is thrown by either your override or the operation, both will be rolled back by the server.

Advanced Queues

Agents and overrides are convenient for automating simple business tasks on the server. However, you may want to automate complex work processes that require graphical monitoring tools and an engine that can execute complex routing logic. In these cases, the Oracle Content Management SDK allows you to use Oracle Advanced Queues to integrate your content management application with Oracle Workflow.

Oracle Workflow is a robust workflow engine that you can use to define, automate, and monitor complex work processes. Oracle Workflow provides a graphical workflow design tool for defining the steps and routing logic of the process. Workflows can include notification steps -- where users are advised to perform certain tasks manually -- and **procedural** steps -- where data processing tasks are performed on the server. It also includes an engine for automating the

workflows. Once a workflow has been initiated, the engine will execute each step, dispatching the appropriate notifications to users or calling server procedures to manipulate data. After each step is performed, the engine evaluates routing criteria to determine the next step it should execute in the workflow. At any point in time, users can check the progress of a workflow using Oracle Workflow's graphical monitoring tool.

Oracle Advanced Queues allows you to configure inbound and outbound queues for processing requests between your content management application and Oracle Workflow. For example, you could create an agent that submits a request to initiate a review and approval workflow when documents are updated in the repository. The agent would submit the request to a queue that is processed by Oracle Workflow. Oracle Workflow would respond by initiating the appropriate workflow. In the same fashion, the workflow process could dispatch a request to your content management application to manipulate information in the repository. For example, the workflow could request that a document be published for general consumption after it has been approved by changing its ACL. The workflow process would dispatch a request that would be received in an in-bound queue by a custom agent. The agent would then perform the appropriate tasks in the repository using the Oracle Content Management Java APIs.

Integrating with Other Information Systems

The Oracle Content Management SDK allows you to exchange data and manage transactions between your content management application and external information systems. In the same way that your custom application can integrate with Oracle Workflow, it can use Oracle Advanced Queues to send and receive calls with another information system. The Oracle Content Management SDK also provides built-in XML processing capabilities that allow you to exchange data with other information systems in a standard format. When you need to exchange information between systems, you can use a built-in XML renderer to export data to an XML file that the information system can process. Using a built-in XML parser, XML data imported from the information system can be automatically processed to create the appropriate objects in the repository.

For example, you may need to integrate your application with a Product Data Management (PDM) system. When new product lines are defined in the PDM system, you could automatically create a category for the product line to organize product documentation in the content management system. To do so, you would need to initiate a request to create the category and also pass the product line meta-data in a standard data format. You could use Oracle Advanced Queues to create an agent and inbound queue to process requests from the PDM system. You can use XML to define a common schema for product line information. Using the Oracle Content Management SDK's XML parser, the agent could parse XML files generated by the external information system to create the corresponding product line categories in the content management system.

SUMMARY

The Oracle Content Management SDK significantly reduces the implementation time and effort required to build content management applications. It allows you to leverage the strengths of the Oracle 9i database to store and manage all types of business information. The Oracle Content Management SDK makes information in the database as easy to access as a standard file system. It supports standard Internet technologies like Java, JSPs, and XML for building multi-tier applications designed to scale and perform to the demands of a large-scale deployment. The Oracle Content Management SDK provides built-in functionality that you can use to organize, control access to, and manage the life-cycle of information in the database – without having to reinvent the wheel. It provides frameworks that make it easy to process information in the repository, automate business processes, and integrate with external information systems.

The Oracle Content Management SDK gives you the flexibility to manage the development of your application in phases: first making information in the database universally accessible from common desktop tools, then automating data processing tasks on the server, and later building a fully custom Web user interface that is tailored to your business. Whether your application is simple or complex, the Oracle Content Management SDK provides the tools you need to build it.



Managing Data in the Age of the Internet:
Oracle Internet File System (iFS)
June, 2002
Author: Dennis Dawson

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2002 Oracle Corporation
All rights reserved.