IBM FileNet P8

Version 4.0

IBM

System Overview

IBM FileNet P8

Version 4.0

IBM

GC31-5482-03

System Overview

Table of Contents

Revision Log

Part Number	Date	Revision
GC31-5482-02	11/4/2008	Added Revision Log.
		Added architecture information for IBM FileNet Connector for SharePoint Document Libraries and IBM FileNet Connector for SharePoint Web Parts.
		Added description of and architecture information for Application Integration BCS.
		Added description of WebSphere® Business Monitor for monitoring active IBM FileNet BPM processes.

Introduction

IBM® FileNet® P8 offers enterprise-level scalability and flexibility to handle the most demanding content challenges, the most complex business processes, and integration to all your existing systems.

FileNet P8 is a reliable, scalable, and highly available enterprise platform that enables you to capture, store, manage, secure, and process information to increase operational efficiency and lower total cost of ownership. FileNet P8 enables you to streamline and automate business processes, access and manage all forms of content, and automate records management to help meet compliance needs.

This system overview provides a technical summary of the components that make up FileNet P8. The "Features" section of this document presents an overview of product features and capabilities. The "Architecture" section provides an architectural overview of the system components and includes information about enterprise deployment and application development.

Some expansion products described in this document might not be available at the time of the FileNet P8 4.x release. In these cases, a description of the expansion product's current release is included here in order to provide a complete picture of the FileNet P8 platform.

Who should read this document

The *IBM FileNet P8 System Overview* contains information that is helpful to:

- Anyone who wants a general understanding of the FileNet P8 architecture, services, components, development and analysis tools, and applications.

- Developers responsible for building site-specific FileNet P8 applications.

- IT personnel responsible for the rollout of a FileNet P8 environment who must understand how FileNet P8 integrates with business systems.

Accessing IBM FileNet P8 documentation

This document contains references to related IBM FileNet P8 documentation. To access these documents:

1. Navigate to the Product Documentation for FileNet P8 Platform Web page (http://www-1.ibm.com/support/docview.wss?rs=3247&uid=swg27010422).
2. Select the appropriate product link.

To locate FileNet ECM products not listed, use the standard IBM navigational aids available on the IBM Web site (www.ibm.com).

Features

The FileNet P8 family of products includes back-end services, development tools, and applications that address enterprise content and process management requirements. This section presents information about features and product capabilities. The Architecture section includes an architectural overview of the FileNet P8 platform components.

Content management

At the core of the platform are repository services for capturing, managing, and storing your business-related digital assets. Multiple repositories, called object stores, can be created and managed within a single system to serve your business requirements. Object stores can be configured to store content in a database, a file system, a fixed content device (such as an Image Services repository, Network Appliance SnapLock, or Centera), or a combination of these options. An object store is capable of storing a variety of business-related data—for example, an insurance claim, a customer loan account, or information about business partners. It can also store any type of structured or unstructured content such as XML documents, Office documents, Web pages, photos, voice data, images, process definitions, templates, and more.

The following diagram illustrates object stores that might be required by a financial institution. Explanatory information follows the diagram.
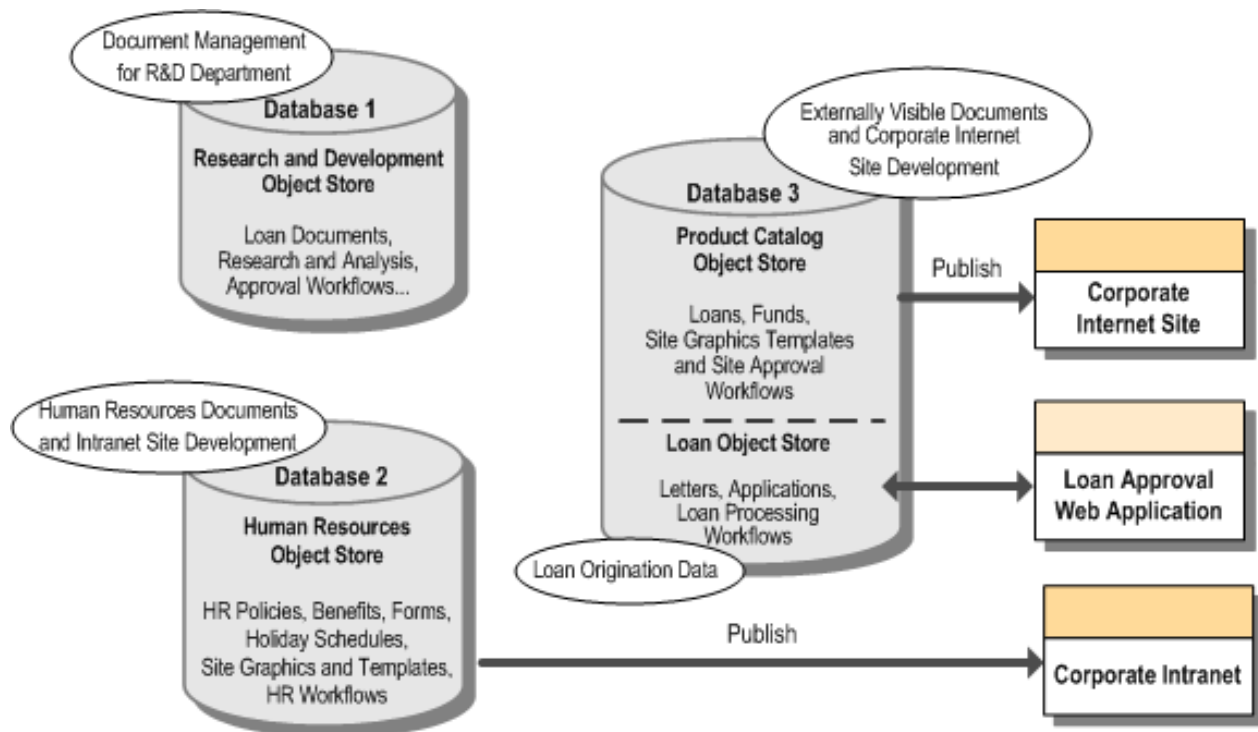
The Research and Development object store is for internal use and contains documents related to the development of new products and services. The Human Resources object store contains information for employees and is also used to manage the corporate intranet.

The Product Catalog object store contains information that is published to the company's Internet site, such as information on products and services, plus the graphics and templates used to publish the information to the corporate Internet site. The Loan object store contains data that is used for the loan approval process. The Product Catalog object store contains information about products.

Business object management

Traditionally, a document management system manages document content and the metadata (properties) that further describe the document. Content Engine provides document management capabilities plus the ability to manage other types of data. For example, as shown in the previous figure, object stores manage traditional office documents plus customer information and loan documents. In this context, the term "business object" refers to any object that is stored and managed in the system, and includes both structured and unstructured data. For example, the following diagram shows four objects: Loan, Loan Application, Customer, and Loan Policy.
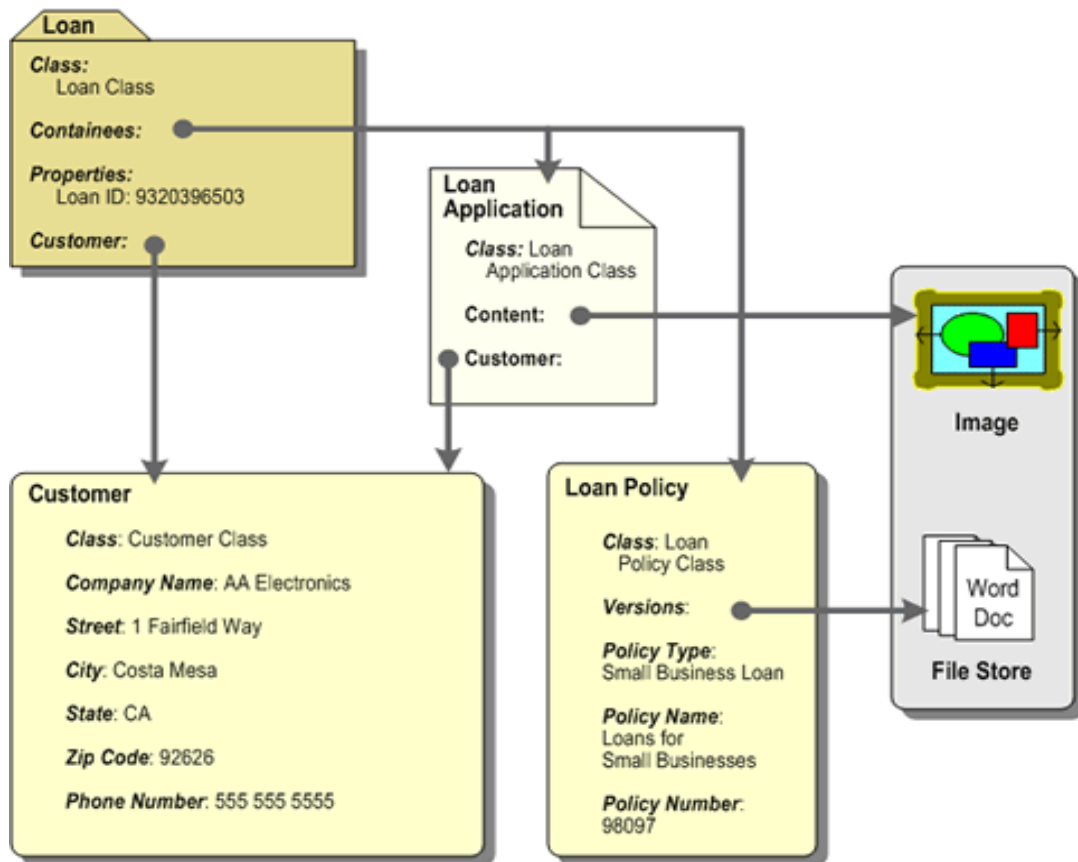
As illustrated in the diagram, an object might or might not have content. The Loan Application has a TIFF image as its content and the Loan Policy has a Word document as its content.

The Customer is an example of an object that has no content, but is simply a collection of properties that describes a customer. The Loan object is an example of a Folder object that can contain Loan-related documents that are visible through the "containees" property of the Folder object.

Leveraging Content Engine's ability to manage data stored in a non-proprietary relational database format enables rapid development.
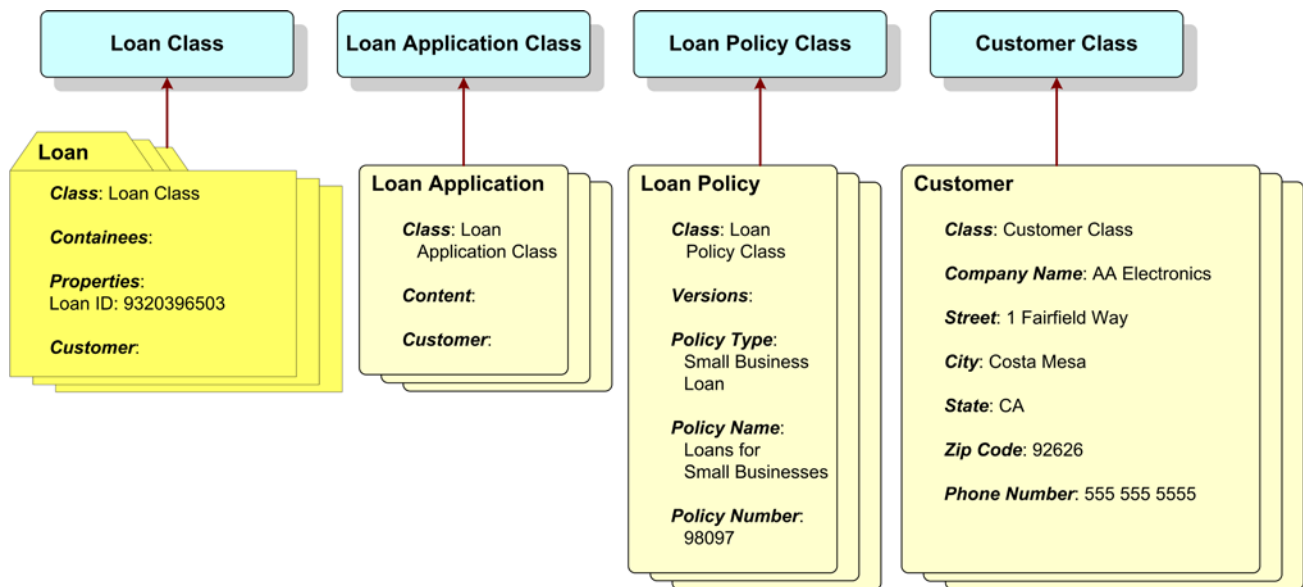
The system comes with a set of predefined classes, called base classes, including Document, Folder, Annotation, and CustomObject. These classes define the behavior for most business-related objects. Customers can use Enterprise Manager to create subclasses that define variations of these basic class types. For example, the Loan class in the example is a subclass of the Folder class, because it requires "container" behavior. That is, it is a folder that contains the various documents associated with a loan.

When subclassing the base classes, you can define:

- Custom properties that are used to provide business-related metadata—for example Loan ID and Customer.

- The default access rights for the object.

- The default security policies that can be applied to the object.

- Event subscriptions for performing custom actions that are executed when the object is added or updated.

- Where a document's content is stored, whether or not it can be versioned, its lifecycle policy, and any automatic classification templates.

The Loan class has been customized by adding a Loan ID and Customer property. The following image shows how this custom class appears in Enterprise Manager. Note that its location in the hierarchy is under "Other Classes."
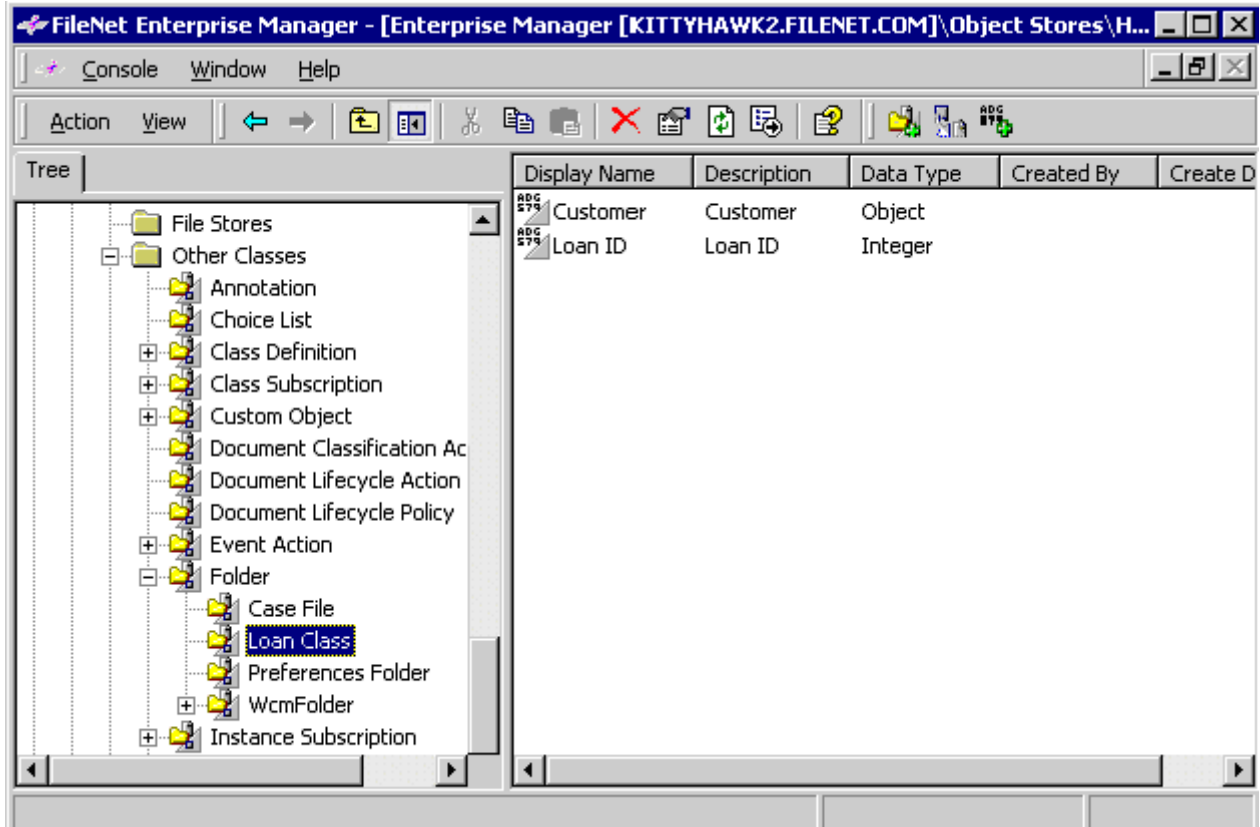
Content Engine enforces the behavior of the base Document, Folder, Annotation, and CustomObject classes as described in the following sections.

Document – Most users think of a document as a file they create with an application such as Word. The user stores the document in the document management system so that a history of changes to the document is maintained and the document can be easily found and edited. Users who design enterprise content management applications and those who manage them will need to understand how documents can be leveraged to support a variety of application needs. A document might be used to maintain a traditional electronic file as well as other types of data, such as an XML document or content that is managed in an external repository. Documents:

- Have system properties that the system manages automatically, such as Date Created.

- Can have custom properties for storing business-related metadata about the document.

- Are secured.

- Can have content that can be indexed for searching.

- Can point to content that is outside of the object store (external content).

- Can have no content (metadata only).

- Can be versioned to maintain a history of the content over time.

- Can be filed in folders.

- Can have a lifecycle.

- Can participate in business processes as workflow attachments.

- Can generate server events when they are created, modified, or deleted. These events are then used to customize behavior.

- Can be rendered to different formats, such as PDF and HTML.

- Can be published to a Web site.

- Can be annotated.

- Can be audited.

Folder – A folder is a container that is used to group other objects. Folders are the primary mechanism through which users access documents. Users typically think of folders as a place where documents are stored;, however, filing documents in multiple folders does not create extra copies of those documents, but rather creates a logical association between the folder and the document. Folders:

- Have system properties that the system manages automatically, such as Date Created.

- Can have custom properties for storing business-related metadata.

- Are secured.

- Are hierarchical, in the sense that a folder can have subfolders.

- Can contain documents and custom objects.

- Can generate server events when they are created, modified, or deleted. These events are then used to customize behavior.

- Can be annotated.

Annotation – An annotation object, illustrated at right, represents incidental information that can be attached to an object for the purpose of annotating or footnoting that object. You can associate annotations with custom objects, documents, and folders. Annotations:

- Are independently securable. Default security is provided by the class and by the annotated object. An annotation can optionally have a security policy assigned to it.

- Can have subclasses.

- Can have zero or more associated content elements, and the content need not have the same format as its annotated object.

- Are uniquely associated with a single document version, and thus are not versioned when a document version is updated.

- Can be modified and deleted independently of the annotated object.

- Can be searched for and retrieved with an ad hoc query.

- Can subscribe to server-side events that fire when an action (such as creating an annotation) occurs.

- Can participate in a link relationship.

- Can be audited.

CustomObject – A custom object is used to store and manage data that does not have content (and thus doesn't support versioning) or a lifecycle. For example, a customer might be represented in the object store as a custom object because there is no requirement for content. Custom objects:

- Have system properties that the system manages automatically, such as Date Created.

- Can have custom properties for storing business-related metadata.

- Are secured.

- Can participate in business processes as workflow attachments.

- Can generate server events when they are created, modified, or deleted. These events are then used to customize behavior.

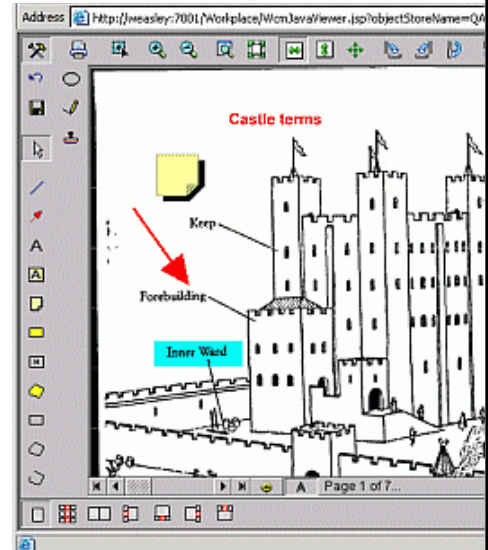The following diagram shows the classes for the Loan-related objects used as examples in previous diagrams. See later sections of this document for more information about events, property types, and other related features.

Folder Class

Subclass of

Loan Class

System Properties	Default Security
Folder Name	Loan Officers,
Date Created	Administrators
Created by	
Containees	
Parent	**Event Subscriptions**
Annotations	Launch Notification
. . .	workflow when a new
	document is file in or
Custom Properties	removed from the folder
Loan ID	
Customer	

Annotation Class

System Properties	Default Security
Annotated Object	Provided by
. . .	annotated object

Document Class ←——— *Subclass of* ———

Subclass of

Loan Application Class

System Properties	Default Security
Date Created	Lending
Created by	
Versions	**Event Subscriptions**
Folders filed in	Launch "New Loan Document"
Content Elements	workflow when new application
Annotations	added to the object store
. . .	
Custom Properties	**Versionable:** No
Customer	
Application Type	**Storage Location**
Status	File Store

Loan Policy Class

System Properties	Default Security
Date Created	Marketing
Created by	
Versions	**Event Subscriptions**
Folders filed in	Launch policy approval workflow
Annotations	when major version checked in
. . .	
Custom Properties	**Versionable:** Yes
Policy Type	
Policy Name	**Storage Location**
Policy Number	File Store

Custom Object

Subclass of

Customer Class

System Properties	Default Security
Date Created	Lending
Created by	
Annotations	
. . .	
Custom Properties	**Event Subscriptions**
Company Name	Update CRM system if
Street	properties are updated
City	
State	
Zip Code	
Phone Number	

There are many additional object types beyond Document, CustomObject, Folder, and Annotation. For example, each custom class that is created (such as the Customer class in the previous example) is managed as an object in the system. This means that system capabilities, such as custom properties and custom event actions, can also be applied to these objects.

Application development and business objects

From a development perspective, it is important to further refine the definition of what a business object is and how it manifests in software. Business objects can be categorized in the following ways:

Solution domain objects define the entities that make up a solution or application. Solution domain objects are understood by the business users, and often map to physical objects in the real world, but cannot map to objects in code.

Implementation objects are the objects that are manifested as code—for example, a Java™ class. The following points describe examples of implementation objects:

- **FileNet P8 objects** are implementation objects that are represented as a set of Java classes within FileNet P8 Java APIs. These objects include Document, Folder, WorkflowDefinition, WorkObject, ClassDescription, and more. These objects are used to manipulate data that is stored in the databases and file systems that are managed by the Content and Process Engines. FileNet P8 tools make it easy to customize the behavior of these objects in graphical user interfaces. For example, as defined earlier in this document, new types of documents that have custom properties and default security can be created without custom programming. In many cases, using the platform objects is sufficient for building applications.

- **Application objects** are implementation objects that solve specific business needs. These objects are often an aggregation of the FileNet P8 objects and are typically run in the classic "business" tier of a Web application as Java objects. These objects can also leverage core application server capabilities, such as JDBC, and interact with packaged applications or other components built in-house. Application developers typically create application objects by creating new Java classes or subclassing the Java classes delivered in the platform.

Properties

Content Engine object properties can be used to reference objects that reside in any object stores within the same FileNet P8 domain.

Properties have a type, which can be Binary, Boolean, String, DateTime, Integer, Float, ID, and Object. Object-valued properties are useful for defining relationships between objects. The Customer property shown in the loan example in the previous section illustrates how both the Loan and the Loan Application can point to the same Customer object. The properties that are associated with objects can have a single value or multiple values. For example, you might define a multi-valued property named Telephone Numbers in which multiple telephone numbers, such as home and cell phone numbers, are stored. You can then search for objects by creating a search expression that searches for more than one value for the same property.

Properties can also be configured to have default values that are set when a new object is created. In addition, the system can be configured to restrict the values for the property to a choice list. A choice list is a list of possible values that the user can select from when assigning a value to the property.

Events and subscriptions

Events provide a mechanism for initiating actions that are invoked when objects are created and modified in, and deleted from, an object store. For example, creating a document in an object store triggers a *create* event, which launches a workflow that approves the new document and posts the approved content to a Web site.

A subscription is the association of a particular event trigger with an event action. In the previous example, *create* is the event trigger and the *event action* is the workflow launch. Many different subscriptions might be associated with a particular event trigger. The following diagram illustrates how several events might be triggered when a new loan application document is created in an object store.
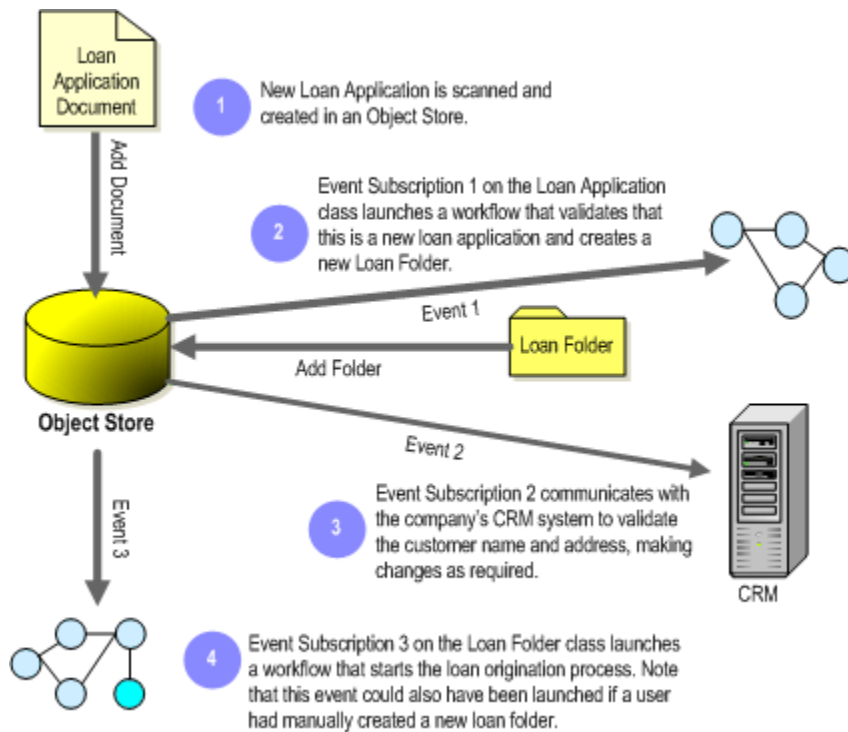
FileNet P8 comes with predefined event actions, including launching a workflow. In addition, events can be developed as a Java class.

Subscriptions can be associated with a class so that they apply to the class itself or to all instances or all objects of the class type. Or, subscriptions can be associated with individual objects. Event subscriptions can be run synchronously or asynchronously. When set to run synchronously, the object operation (for example, create or update) and the operations of the event actions are completed as a single transaction; failure in either results in rollback of both operations. For example, a synchronous event might be applied to a Claim folder class that returns an error if a document that doesn't belong to the Claim Document class is filed in the folder. When set to run asynchronously, the object operation and the event action operation run as separate transactions; in this case, the object operation can succeed independently of the event action operation.

Lifecycles

Administrators can use document lifecycles to define a sequential set of states that a document will go through over its lifetime, as well as the actions that are triggered when it transitions from one state to another. A user or application can "promote" or "demote" an object to move it forward and backward in its lifecycle. The following table illustrates a simple document lifecycle.

Stage	Event Actions	Lifecycle Action	Lifecycle State	Transactions
Author creates a new document. Lifecycle is automatically assigned when the document is added to an object store.	Create, Check In, Promote Lifecycle	Promote	Draft	Review workflow is launched. E-mail is sent to reviewers.
Document is reviewed and edited.	Check Out, Check In, Promote or Demote Lifecycle. Optional: Update.	Promote	Review	Approval workflow is launched. E-mail is sent to the authorized user.
Authorized user either rejects or accepts changes to the document.	Check Out, Check In, Promote Lifecycle	Demote or Promote	Review	Document is either demoted to review process (e-mail is sent to author and reviewers), or, if promoted, publish workflow is launched (e-mail is sent to authorized user).
Document is published.	Promote Lifecycle	Promote	Publish	E-mail sent to all users associated with this document.
Document is closed.	Update, Reset Lifecycle	Set exception mode	Close	Document is closed to further lifecycle changes.

In addition to the typical actions that are triggered as a document transitions from one state to another is the capability to apply a security template on state change so you can effect changes in security when the document changes state. On entry to the specified state, the security permissions associated with a configured security template are automatically applied to the document.

Search

Content Engine supports property and content-based searching. Key capabilities of search include the following:

- A single search can span multiple object stores in different databases.

- Workplace users can search for documents, folders, and custom objects. Searches can be designed to specify multiple folders, including a common folder name used in multiple object stores.

- Content-based searching using IBM FileNet P8 Content Search Engine supports extensive content search capabilities that account for misspelled words, typographical errors, phonetic searching, word stem searching, synonym expansion, and wildcard searching.

- Content searches return matches on properties and content.

- Search results can optionally be displayed in a "document summary" format, which includes a brief text summary extracted from result documents.

- Search results can be ranked by relevancy.

- Searching makes use of the Autonomy multilingual features.

- Bulk operations can be performed on search results in Enterprise Manager, where the operations can be scripted, or selected from a set of predefined operations such as delete, cancel checkout, file, unfile, and change security.

- Searches can be created and stored for easy execution of common queries.

- Search templates provide a simple user interface for entering search criteria.

- Shortcuts to searches can be saved so it is easy to find them later.

- Searches can be expanded to retrieve documents, folders, and custom objects in the same search specification.

The following image shows the Search Designer tool that is used to create stored searches and search templates.
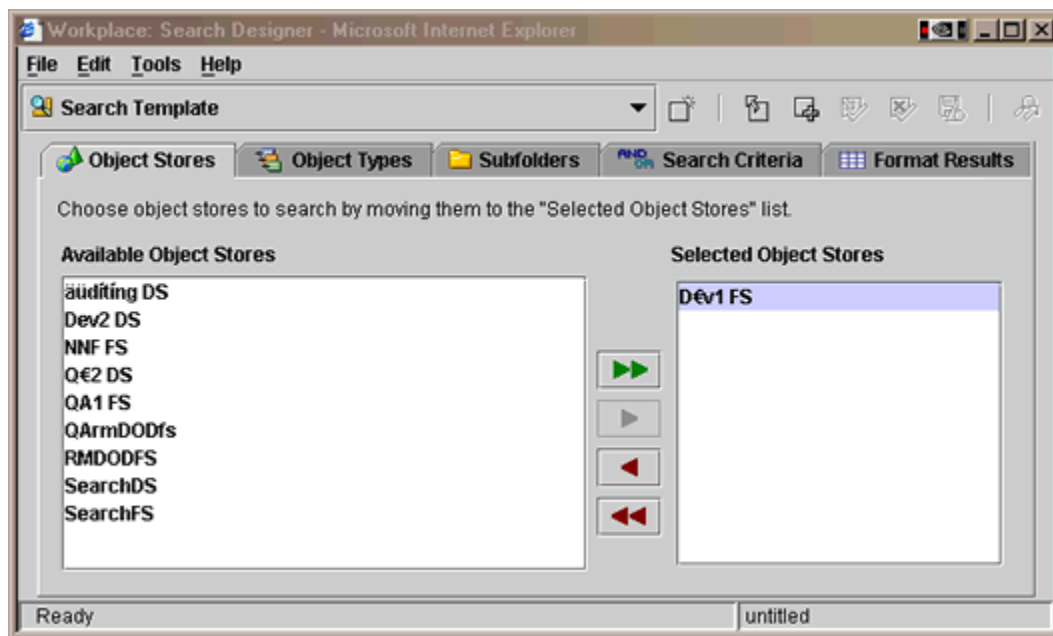
Versioning

You can create different versions of content to maintain a history of changes and to control which users can change the content at a given time. The set of versions for a single document is called a version series. Content Engine supports a two-level versioning scheme, in which a document version is either a major or minor version. Minor versions typically denote an "in-progress" document, whereas a major version typically denotes a completed document. In addition to version numbers, the system maintains a state property indicating the current state of each version of the document, as follows:

- **In Process –** A work in progress version. Only one version of a version series can be in this state.

- **Reservation –** A document currently checked out for modification. Only the latest version of a version series can be in this state.

- **Released –** A document released as a major version. Only one major version of a version series can be in this state.

- **Superseded –** A version superseded by another version. Many versions in the version series can be superseded.

You can configure the names associated with these states to more closely match their document management processes. The following illustration shows how two-level versioning works:

User Actions / Versions

NOTE: Bold text indicates property values that changed based on the user action. Bold lines on version indicate that the version was created as a result of the user action.

1 Bob adds a new document as a minor version.

Version: **0.1** Status: **In Process**

2 Bob checks out the document.

Version: 0.1 Status: In Process

Version: **0.2** Status: **Reservation**

3 Bob saves the reservation at the end of the day, but isn't ready to check in a new version yet.

Version: 0.1 Status: In Process

Version: 0.2 Status: Reservation

4 Bob makes more changes to the document and checks it in as a minor version. He requests that Susan make her changes to the document.

Version: 0.1 Status: **Superseded**

Version: 0.2 Status: **In Process**

5 Susan checks out the document and makes additional changes.

Version: 0.1 Status: Superseded

Version: 0.2 Status: In Process

Version: **0.3** Status: **Reservation**

6 Susan checks in the document as a minor version.

Version: 0.1 Status: Superseded

Version: 0.2 Status: **Superseded**

Version: 0.3 Status: **In Process**

7 Bob reviews his document and determines that it is ready, so he promotes it to a major version

Version: 0.1 Status: Superseded

Version: 0.2 Status: Superseded

Version: **1.0** Status: **Released**

Additional capabilities in the versioning model include the following:

- The system can be configured to apply security policies that in turn automatically apply different access rights for major and minor versions, making it easy to enforce a different viewing audience for in-progress documents.

- A document can be promoted from a minor to a major version without requiring the content to be versioned.

- A document can be demoted from a major version to a minor version, which is useful if the document has incorrectly been promoted to a major version.

- A document can be saved to the object store prior to being checked in. By doing so, users can avoid keeping content on their local systems, and multiple users can work on a checked-out document.

- Versioning can be enabled and disabled on a document class level for cases when multiple versions are not required.

- A version can be frozen so that the custom properties associated with it cannot be changed.

- Users or administrators can view all versions in their respective user interfaces.

- Versions can be deleted.

Classification

Classification is the process of assigning metadata to content, specifically the selection of a document class and property values. Classification can also be accomplished by filing objects into folders that define classification taxonomies. Classification can be performed:

- By a user.

- By an application that uses the FileNet P8 APIs.

- Automatically by using the content-based classification capability provided in the platform.

Entry templates, described in the next section, can simplify the classification process for the user by filing the document into pre-specified folders and by predefining the object's class and property values.

Automatic classification is a capability that classifies an incoming document by examining its content. FileNet P8 supports automatic classification of XML documents, though custom classification plug-ins can be created for any document format. For XML classification, administrators create mapping scripts that associate XML tags in the incoming document to properties, thus enabling the automatic classification of any XML document.

Entry templates

Users can easily add documents, folders, and custom objects to an object store using entry templates. Entry templates also make it easy to define approval workflows for these objects.

For example, a manufacturing company's template designer can create a set of entry templates for the various types of engineering specifications that are managed in the object store. The entry templates in this example predefine the document class, all property values except the document title, the access rights, and an approval workflow. When using the entry template to add a document to the object store, a user interacts with a wizard, which decreases the chance of invalid data entry by limiting the number of steps required and providing a more controlled entry process.

Entry templates are created in Workplace, typically by a relatively small set of users, and are used by a much larger number of users. Some of the key capabilities of entry templates and advantages to using them are as follows:

- Documents, folders, form data, and custom objects can be created with entry templates.

- Users who are not administrators can create entry templates. For example, a project manager can create the entry templates used by her project team.

- Entry templates support FileNet P8 records management by providing a simplified and customizable method of declaring a document as a record.

- Entry templates can specify the folder in which the object will be filed, can either prevent or allow the user to change the folder, and can restrict the user to selecting a particular folder or its subfolders.

- Default property values can be specified in an entry template, and the template designer can elect to show or hide each property to users of the template.

- Entry templates can specify whether a document should be added as a major or minor version and can specify whether a user can change this setting.

- Entry templates can be set up to automatically classify the document based on its content.

- Entry templates can specify the access rights for the object, and the designer can elect to hide or show the access rights for the user to modify.

- Entry templates can serve as placeholders for documents that a user provides while processing a workflow step. When the user clicks on the entry template in the Workplace step processor, the entry template wizard prompts the user to add a document and the new document replaces the entry template attachment.

- Entry templates support specifying a workflow that will be launched when the user creates an object with the template, whether the workflow is a simple approval workflow that is defined in the entry template wizard or any other workflow created using Process Designer. For workflows created using the entry template wizard, the user can choose from a workflow with three steps—review, approve, and publish—or one with multiple, sequential approval steps. The template designer can specify the participants in advance, or let the user do so when using the template.

The following image shows the user interface for defining an entry template for a document. It is currently at the step where properties are set for the template.
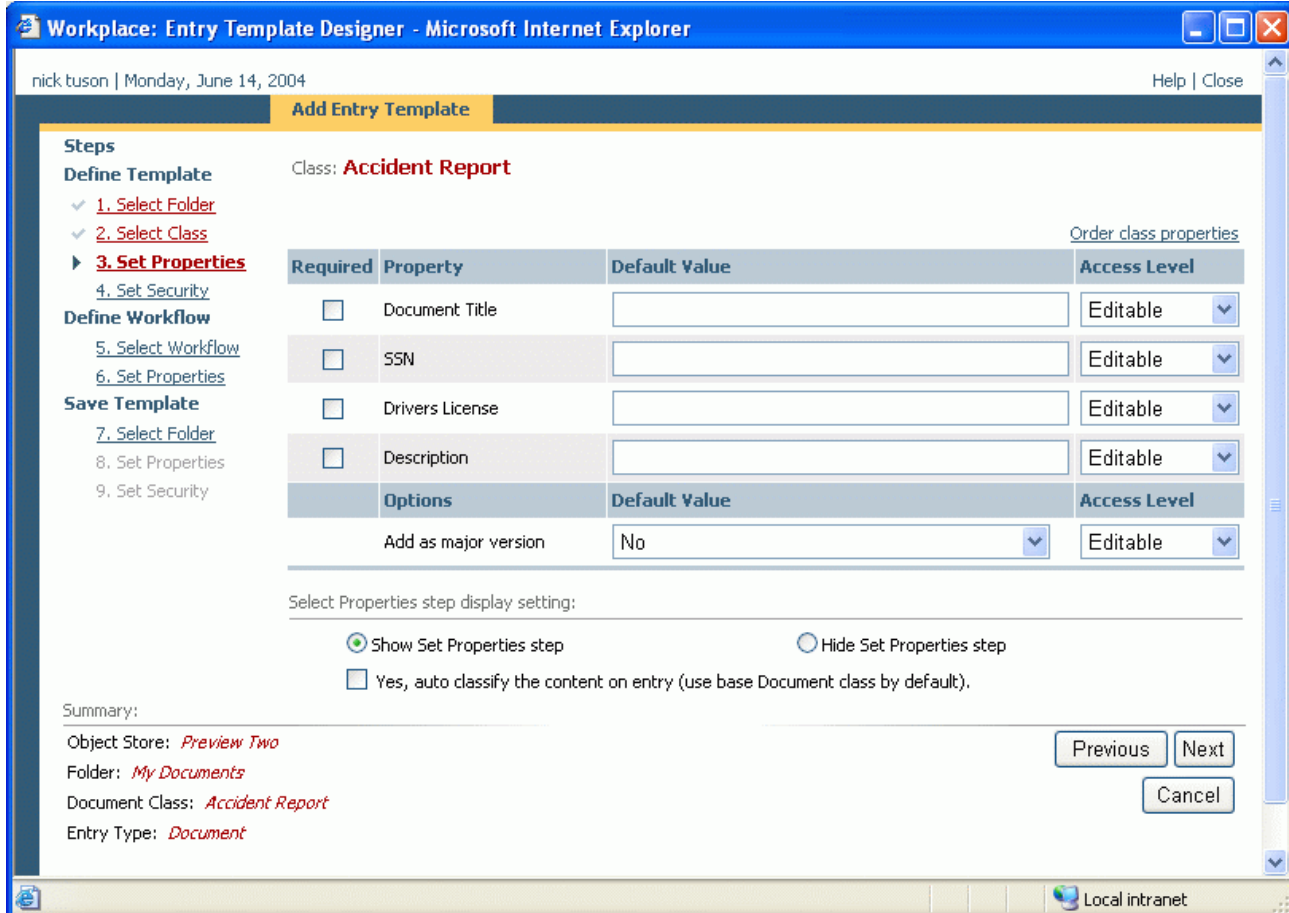
Renditions and publishing

Content Engine supports the translation of documents to PDF and HTML, making it easy to publish content in formats suitable for the Web and printing. The rendition model supports creating as many renditions of a particular document as desired, where each rendition can have its own set of properties and access rights. Having separate access rights makes it easy to define a different viewing audience for different renditions.

For example, employees in the Human Resources department maintain a benefits document. Over the year, the document might be versioned many times, but once a year, an HTML rendition of the document is created and made available to all employees.
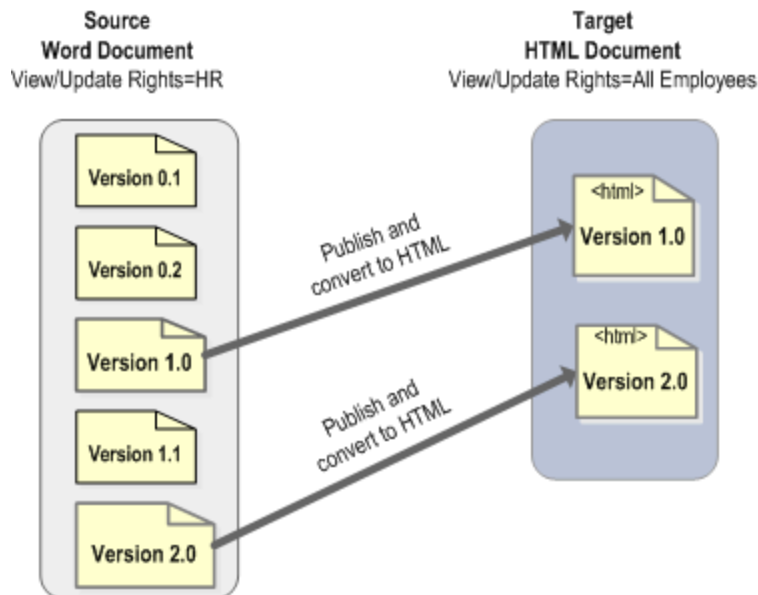

The publishing model supports the following capabilities:

- The relationship between the source document and its renditions is maintained by the system, and recreating the rendition can automatically replace or version the existing rendition, depending on the application requirements. In addition, the system can be configured to automatically delete a rendition if the target is deleted.

- Style templates enable administrators to specify custom behavior when creating a rendition. For example, you can use a style template to create a watermark on a PDF document or to specify a password in the PDF document to restrict the ability to open the document.

- Publish templates enable users to define characteristics of target renditions, including:

  - Where to file the rendition.

  - Access rights associated with the rendition.

  - New metadata values, or that the metadata values are the same as the source.

  - What happens when the document is re-published, such as whether or not a previous rendition is versioned or replaced.

  - An associated custom script (in the form of an event action) that is executed after the rendering has been completed. Such a script provides an easy mechanism to customize behavior.

Managing XML content

Content Engine manages XML content as documents, and therefore XML documents can take advantage of all of the repository services described previously—including metadata, versioning, events, and lifecycles. Additional functionality provides XML-specific capabilities, including automatic classification, schema validation, and content-based searching that filters based on an XML tag.

For example, as shown in the following diagram, an XML document containing information about a customer order is entered into an object store. Automatic XML classification is used to extract the customer ID and order number from the XML and to change the class to "Order." An auto-classification event subscription automatically launches a workflow that fulfills the order. Users of the system can easily find all orders for a particular product by searching for documents that belong to the "Order" class and where the "Product" tag contains the product.
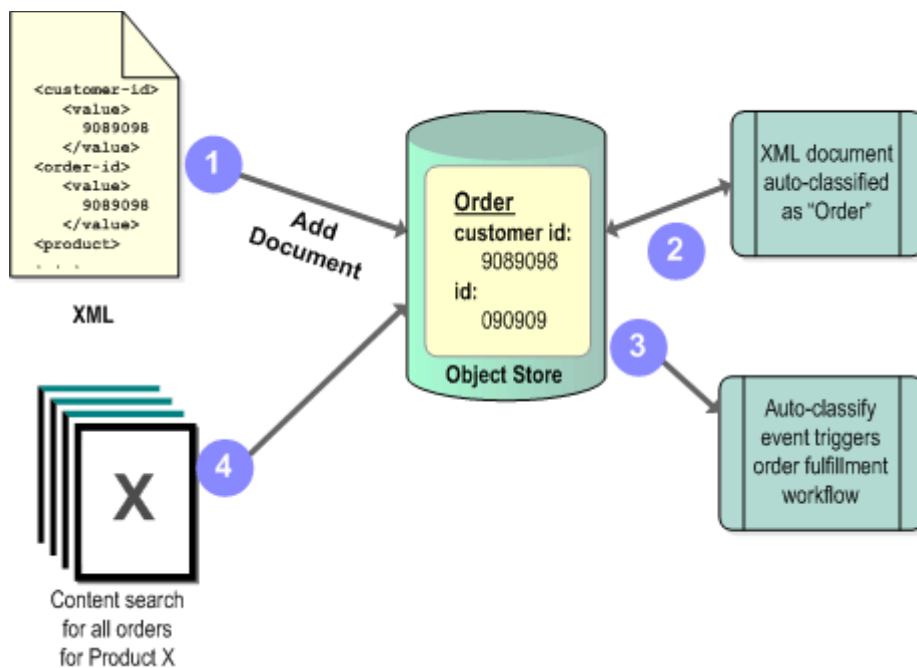
Content storage

Content Engine supports storing content in an NTFS file system, in the object store's database, or in a fixed content storage system.

Fixed content can generally be defined as document content that has been stored on a fixed content system for which the Content Engine provides connectivity and configuration in its fixed content device. Fixed content systems potentially provide extremely large storage capacity and generally provide retention capabilities.

FileNet P8 supports FileNet Image Services as a fixed content store, as well as storage systems such as Network Appliance SnapLock and EMC Centera fixed-content products. From the user or programmer perspective, the storage location is typically transparent and is defined based on document class configuration, storage policies, or storage areas.

Content caching

Both file and database content can be cached on Content Engine servers for quick retrieval of frequently used content.

A content cache is created and managed using Enterprise Manager. One content cache can exist for each server that has the Content Cache Service installed. An administrative utility is provided for monitoring the cache.

A distributed FileNet P8 system will have many clients reading content files from either a remote file server or a SQL Server database (most likely on a remote server). To reduce network traffic, content can be cached in the file system on the local server that hosts the Object Store Service.

The Object Store Service invokes content caching on the server. A list of files in the cache is maintained by the service using an in-memory database. The Content Cache Service manages the files in the cache, replicating files as needed. When the cache reaches the configured size, the service trims the cache, deleting the least-used files.

Import and export

Content Engine provides an import and export capability for moving objects from one object store to another. In Enterprise Manager, an administrator can select the objects to export. Dependencies between objects are detected so that all related objects can also be exported in the appropriate order for import. For example, if a folder is exported, the user can also choose to export all documents in the folder without selecting the documents individually. The folder will be exported first, followed by the documents. The export manifest is a list of all of the objects that the administrator has requested to export. The user can view and modify this list, and then perform the export operation, which copies the data to an XML file.

Content Engine XML import and export APIs include security information related to objects. The exported XML includes the security and owner information for the object. You can import the security information and owner information independently. The APIs don't circumvent any security associated with the FileNet P8 domain or the FileNet P8 object store; these permissions are still checked when trying to update security and owner through import.

Enterprise Manager includes a security mapping tool that you can use in conjunction with the XML security import/export. Using the tool, you can map security principals in one domain (development, for instance), contained in an exported XML file, to security principals in another domain (production) before import.

During the import process, you can run scripts to perform additional customization. For example, you might want to update the access rights associated with the new document as it is imported. Scripts can run for each object imported or at the beginning and end of the import process.

Content Engine can be configured to import automatically when a new object store is created. For example, an administrator might define a set of properties, such as Employee ID and Department, which must be consistent for all object stores in the enterprise. These can be defined in a single object store and then exported to an XML file. The administrator can then configure Content Engine to automatically create these property definitions every time a new object store is created.

Integration with external storage devices

Content Federation Services (CFS) enables you to integrate data in an external repository with FileNet P8 and access the documents as though they were stored in a FileNet P8 object store.

CFS creates a reference within an object store that points to the source document in an external repository. The document in the FileNet P8 object store is represented by metadata only; the document in the external repository contains all its original content. In addition, CFS provides a mechanism for mapping source document metadata to Content Engine document properties.

The process of creating a Content Engine document that references a document in an external repository and mapping the source document metadata to Content Engine document properties is called federation. No changes are made to either the content or the metadata at federation. The federation process is managed by sets of rules and data mappings that define what information to retrieve, when to federate documents, and how to map metadata information from the source document to document property values in Content Engine.

Once documents have been federated, they can be used with all the standard FileNet P8 functionality, including property searches, records management, and workflows. In addition, if the FileNet P8 object store is configured for full-text indexing, the content of text-based federated documents is indexed so that text-based searches can locate the documents.

An integral part of the records management process is ensuring that documents that have been declared as records are not deleted until the time specified by the records management disposition schedule. You can use IBM FileNet Records Manager with federated documents; once a document is declared a record, it is automatically locked down so that the content of the existing version can only be deleted by the Records Manager application.

Other CFS capabilities include failover, failure recovery, duplicate detection, federation of external repository updates, federation of documents with version series, and storage of general configuration information. CFS can be configured for high availability environments.

CFS supports Image Services, Content Services, and repositories acquired from other vendors. The following sections provide details for these options.

Integration with Image Services

Content Federation Services for Image Services (CFS-IS) natively integrates and federates content from Image Services repositories. CFS-IS enables Content Engine to use Image Services as another content storage device. Users of FileNet P8 applications have full access to content stored in existing Image Services repositories. Anything that is created through a FileNet P8 application or created programmatically by using the Content Engine APIs can be stored in the Image Services permanent storage infrastructure. Existing Image Services content is preserved and usable by Image Services applications, and is reusable by FileNet P8 applications, such as Workplace XT and Records Manager, without duplication and without change to existing applications. The location of document content is transparent to all applications.

The ability for Content Engine document indexes to directly connect to the repositories of Image Services systems provides a number of key features and benefits:

- CFS-IS enables direct access to existing Image Services documents from Workplace XT or any application built on the FileNet P8 platform.

- You can continue to use all existing applications that store documents into Image Services systems (for example, HPII, MRII, COLD II, DW SAP, Capture, and other applications).

- CFS-IS enables the use of Image Services to store new Content Engine documents.

- CFS-IS provides a robust mechanism for cataloging Image Services documents in Content Engine.

- You can migrate existing document properties from an Image Services catalog to a Content Engine catalog using the Image Services Catalog Export tool.

- The connections between Content Engine and Image Services systems tolerate network outages and system restarts on either side of the connection.

High volume imaging

You can leverage your existing Image Services infrastructure to capture content the way you've always done, but you can now make that content available to FileNet P8 applications. This content includes new documents being captured every day as well as the documents you have already stored in Image Services repositories. You can configure your system to catalog metadata in the Image Services catalog, the Content Engine catalog, or both. Content stored in an Image Services content store is available to all applications.

Image Services and Content Engine catalog synchronization

For those enterprises wishing to gradually migrate content from an Image Services system to a Content Engine system, dual cataloging of documents is an option. Image Services documents are cataloged in the Content Engine catalog, but can also be cataloged in the Image Services catalog, resulting in all content being accessible by both Image Services applications and any application built on the FileNet P8 platform. Both the Image Services and Content Engine catalogs are masters and are automatically synchronized by CFS-IS. If properties change in Image Services, they are automatically propagated to the Content Engine catalog. Note that synchronization is not bidirectional; that is, updates in the Content Engine catalog do not propagate back to the Image Services catalog. (For more information, refer to *IBM FileNet P8 Content Federation Services for Image Services Guidelines*.)

During the transition period from the Image Services to the Content Engine catalog, both FileNet P8 platform applications and Image Services applications can run concurrently, accessing the documents stored in Image Services. When a transition to the Content Engine catalog is complete, entries can be removed from the Image Services catalog, and Image Services applications can be replaced by FileNet P8 applications.

Image Services Catalog Export Tool

Existing documents, images, and other content that are already cataloged on the Image Services system can also be cataloged on a Content Engine system. The Image Services Catalog Export tool exports existing index data (document properties) from an Image Services catalog to a Content Engine catalog.

The tool includes an option to delete the index entries from the Image Services index database after the index data has been exported to the Content Engine catalog. Information about this tool, and instructions for its use, are in the *IBM FileNet P8 Content Federation Services for Image Services Guidelines*.

Integration with Content Services and other repositories

Content Federation Services for Content Services (CFS-CS) and Content Federation Services for IICE (CFS-IICE) access information in Content Services and repositories acquired from other vendors by using services provided by IBM's Information Integrator Content Edition (IICE) product. Before accessing the information, you must:

- Create the data maps that identify how metadata in the external repository maps to document properties in the FileNet P8 object store. In general, there is one data map per source repository. Multi-value properties are supported for metadata mapping.

- Define federation rules that identify which documents to federate and which folders will store the federated documents in the FileNet P8 object store.

You can have multiple rules defined so that you can tailor the mapping and storage information for different categories of source documents. Each federation rule is composed of three elements:

- The data map that defines how metadata in the source repository is to be mapped to Content Engine document properties

- The documents to be federated; this information is generated using search criteria

- The document class to associate with the federated document and the folder and object store in which to store the federated document

Each rule execution is known as a "run." For each run, the federation process executes the rule's criteria, extracts metadata from documents that meet the specified criteria, maps the properties from the external repository to metadata properties on Content Engine, and stores the metadata in the specified object store in Content Engine. Your organization can determine whether to federate documents on a daily or on an ad-hoc basis.

Optimizing federation

Each rule can cause millions of source documents to be federated at one time. Several options are provided in the CFS Federator Administration tool to minimize the load on both the source repository and the FileNet P8 system and to optimize the number of documents that can be processed in a designated time period. These options are referred to as chunking and sweeping. Chunking breaks the query and processing into pieces. Sweeping limits the processing to new and updated documents only. Chunking options include:

- Scheduling rules to run on a daily or an ad hoc basis. When the rules are scheduled to run daily, the hours during which the rules can run can be limited. If all the documents defined by the rules cannot be processed within the specified time period, the processing continues automatically the next day.

- Specifying the query block size and number of workers for processing the federation requests.

- Specifying the batch size for the query processing.

Version mirroring

If the source repository supports versioning, the federation process maintains the relationship between the versions of the source document by creating a matching version series in the FileNet P8 object store. As new versions of the document are created, scheduled federation runs update the matching version series in FileNet P8 with the new versions.

Business process management

FileNet P8 business process management (BPM) software lets you create, modify, manage, analyze, and simulate business processes (also referred to as workflows) that are performed by applications, enterprise users, and external users such as partners and customers.

Defining business processes

FileNet P8 process definition functionality extends from the integrated expression builder—which provides a means of associating complex business rules with various routes between workflow steps—through features such as native process voting, in which the state of routed work is analyzed based on responses from all users participating in a particular task.

Workflow definitions

By creating a workflow definition, you can define the activities and resources required to accomplish a business process. A workflow definition consists of a series of process steps connected together by a series of routes, which define the sequence in which the steps are executed. The workflow definition acts as a processing template that is used each time the workflow runs, routing the work to the specified participants or processes, along with the data, attachments (documents, folders, and custom objects), and other information needed to complete the activities.

Process version management

Defined processes are stored in a FileNet P8 object store. Storing the definitions in an object store provides a mechanism for managing the lifecycle of the process definition by controlling user access and managing different versions of the same process.

If, for example, a decision is made to institute changes regarding an organization's expense approval process, a new process can be implemented with the following steps:

1. Check out the existing process from the FileNet P8 object store.

2. Modify the process using Process Designer.

3. Check the new version of the process definition back into the FileNet P8 object store.

Any new work created follows the latest approved process map available, while existing work in the system remains unaffected. (New work can also be launched explicitly referencing an older version of the process.)

The benefit is rapid implementation of changes to business processes without requiring new versions of processes to be backwards compatible, resulting in dramatic reductions in time-to-deployment and the ability to react quickly to business or transaction events. For example, the new process definition might have associated attachments that outline revised guidelines under which any new expense reports must be evaluated. This attachment can reference a specific version of the guidelines (also stored in the FileNet P8 object store) revised to reflect the policy changes. This ensures the correct guidelines are available to the participants in the approval process for both new and pre-existing expense reports.

Process submaps

Processes can be developed using a collection of nested subprocesses, or submaps. Submaps can be defined in Process Designer or they can be imported from another process definition.

When submaps are imported, the integrity of the overall process is ensured by adding any properties referenced in the imported submaps, such as data fields, workflow groups, or process milestones, to the main process data dictionary.

Navigation around a complex process definition, including multiple submaps, is simplified in the definition environment. Users can move to the submap referenced by a submap step by simply double-clicking on the step icon or browsing the list of available submaps.

While viewing a submap, the display shows all of the places in the process from where that particular submap is called; selecting one of the displayed submap names displays the associated process submap. In addition to being called directly from another map, submaps can also be invoked by error processing or the expiration of a timer.

This ability to break down processes into simpler reusable components makes complex processes easier to understand and results in a significant reduction in the total cost of development. Additionally, the reuse of predefined process submaps ensures the consistency of processing and considerably reduces cycle time and costs associated with the deployment of new processes.

Process inheritance

To establish consistent processing and expedite creation of workflow definitions for a group of related processes, a workflow author can create workflow definitions that inherit the characteristics and properties of previously defined workflow definitions.

All workflow definitions the author creates inherit system data fields and two submaps, Terminate and Malfunction, from the FileNet-provided base class for all workflow definitions. In addition to inheritance from the base class, the workflow author can define common characteristics in workflow definitions at a high level in the class hierarchy and automatically pass these characteristics to subsequently derived workflow definitions. For example, a department might process several types of insurance claims that are related in many ways, but each has a few unique processing tasks. The workflow author can create a "parent" workflow that contains all of the similar tasks, then create specialized "child" workflows—one for each type of claim—that inherit the shared processes from the parent workflow as well as contain the unique tasks each child workflow requires to complete its processing.

When creating a new workflow, the author simply selects a workflow on which to base the new workflow. The new, or derived, workflow is said to have "inherited" the items (maps and properties) of the workflow it is derived from. If changes are made to the parent workflow, the changes are propagated to all of its derived workflows. Inherited items are read-only in the derived workflow definition, but the workflow author can override any inherited item by redefining it.

Steps

The steps in a workflow either represent a specific business task or a system activity. A business task can be executed by an individual user, by a group of users, and/or by an automated application. There are several different step types:

- A launch step is the first step on the main map of a workflow.

- A general step can be either a participant step, a work queue step, or an unassigned step:

  - A participant step has an associated user or collection of users, all of whom must process the item to complete the step. The identity of these users can be defined at runtime through the use of workflow groups; thus a single workflow definition can exhibit different behavior depending on the value of the user identities contained in the workflow groups defined for that specific workflow.

  - A work queue step routes work to a specified work queue for processing by a participant from a pool of participants or by an automated application.

  - An unassigned step is one that is not assigned to a participant or a work queue. These steps are processed immediately by the system.

- A system step identifies one or more system functions that provide specific functionality within a workflow. Examples of such functionality include setting time limits for certain activities, assigning data field values, creating new workflow instances, and suspending the workflow for a specified period of time or until a specified condition is met.

- A submap step calls another workflow map. An individual submap can be referenced multiple times within a workflow definition, and multiple levels of nesting (for example, submaps within submaps) are supported.

- A start step is the first step on a submap.

- A component step executes operations in an external program, also referred to as a component. Components are controlled by using Component Manager on the Application Engine server and are configured using Process Configuration Console.

Parallel work and routing

Routing defines the order of execution for a series of steps, which can be based on specific rules and events. With the exception of the last step on the map, every step has one or more routes leading from it. You can define a route so that it is always taken or is taken only if a predefined condition is met. Workflow steps can occur in parallel to facilitate more efficient processes.

Process voting

In many cases, a single task can require action by a number of users collectively. These users can be identified explicitly or implicitly by using workflow groups.

It is vital in such cases that group actions be carried out in parallel rather than in sequence, in order to minimize processing time. If each user's response is to have an impact on the flow, it is necessary to be able to evaluate the responses as a whole in order to derive a consensus.

With vote-based routing, a number of responses are analyzed and the route of the work can be based on the following criteria:

- All responses being of a specific value (example: all users select Approve).

- Any of the responses being of a certain value (example: any one user selects Reject).

- None of the responses being of a certain value (example: no users select Reject).

- The count of responses of a certain value (example: more than one user selects Reject).

Vote-based routing criteria can also be combined with more conventional data-based decision criteria (for example, the document type is "Press Release" and all users selected Approve). Users carrying out parallel tasks can also be made aware of the responses of other users. This capability allows for collective decision-making and ensures that the appropriate level of control over the process is always maintained.

Work prioritization

By default, tasks are processed in the order in which they are received. You can modify the processing order based on any data field associated with the process, such as a required service level.

Deadlines and timers

You can use deadlines and timers to further ensure that work is processed in a timely manner. A deadline provides a time-based scheduling constraint, which requires that a step or workflow be completed within a certain amount of time. The deadline can be relative to the time the step was routed to the participant or to the time the workflow was launched. An expression builder is available to let you define a time expression for the deadline.

A participant with a deadline can receive an e-mail reminder of the pending deadline. When the deadline is passed, a visual reminder displays in the participant's inbox, and an e-mail can be sent to a configurable list such as one or more supervisors. The distribution list can be specific to each work item. This automatic process escalation has the additional benefit of proactively ensuring that certain functions or processes are completed on time and are done so without tying up resources to continuously monitor system activities.

A timer indicates a period of time during which you want a specified series of steps to process. If the timer expires before this processing is complete, processing proceeds to another workflow map that provides alternate processing of the work.

For example, in the case of tasks requiring human intervention, a backlog can result, causing a delay in processing. One late step results in the late arrival at the next step and so on, resulting in further delays. By setting a timer, the work can be automatically routed to an escalation submap when the time limit is exceeded. To assist with escalation, an associated submap might represent a "fast-track" process, or might simply reassign the value of a priority flag used to sort work in the queue. In this case, even if the work arrives last, it will be processed first, ensuring the work meets the desired timeline and maintains the required level of service and performance. If necessary, each item following a given process can have a different service level depending on the data contained in the work item. Multiple timers can be set simultaneously on each item of work, each with a different escalation process, thereby increasing processing flexibility even further.

Interrelated processes

In any business process management solution, not only can individual processes be complex, but processes can also be programmed to interact with each other. This facilitates the implementation of processes that cut across the internal and external boundaries of an organization.

An example of this is process outsourcing, where the responsibility for the execution of a particular function is delegated to an outside service provider or business partner. The outside company should have access only to the specific information required to complete the process. This can be achieved by creating an entirely separate process to represent what is to be done by the outside company. This process can be invoked from the main process. The work to be outsourced is then automatically created as a separate process, and is then merged back into the main process once the outsourced tasks have been completed.

Because the outsourced tasks are defined in a separate process, rather than a submap, they can be distinguished from the main process. This allows an entirely separate audit log that can be tracked independently from the main process.

You can use Business Process Modeling Notation (BPMN) to assist you when designing complex, interrelated processes such as these. BPMN is a standardized graphical notation designed to allow business users to create diagrams of processes using their choice of diagramming tools, and then output those diagrams in a standard format, XPDL 2.0. Advanced process modeling applications such as Process Designer can open these diagrams and map the diagram elements to workflow entities, steps, and routes. This eliminates the time-consuming task of re-creating the steps and routes in Process Designer, improving the communication between the business user and the workflow author.

Event-based workflow launching

Workflows can automatically launch when objects in an object store change. For example, if a new customer object is created, a workflow that manages new customer acquisition can automatically launch.

Graphical process design

You can use Process Designer to easily create a workflow definition through a graphical user interface. In this example, the first three steps have been added to a multi-step workflow.
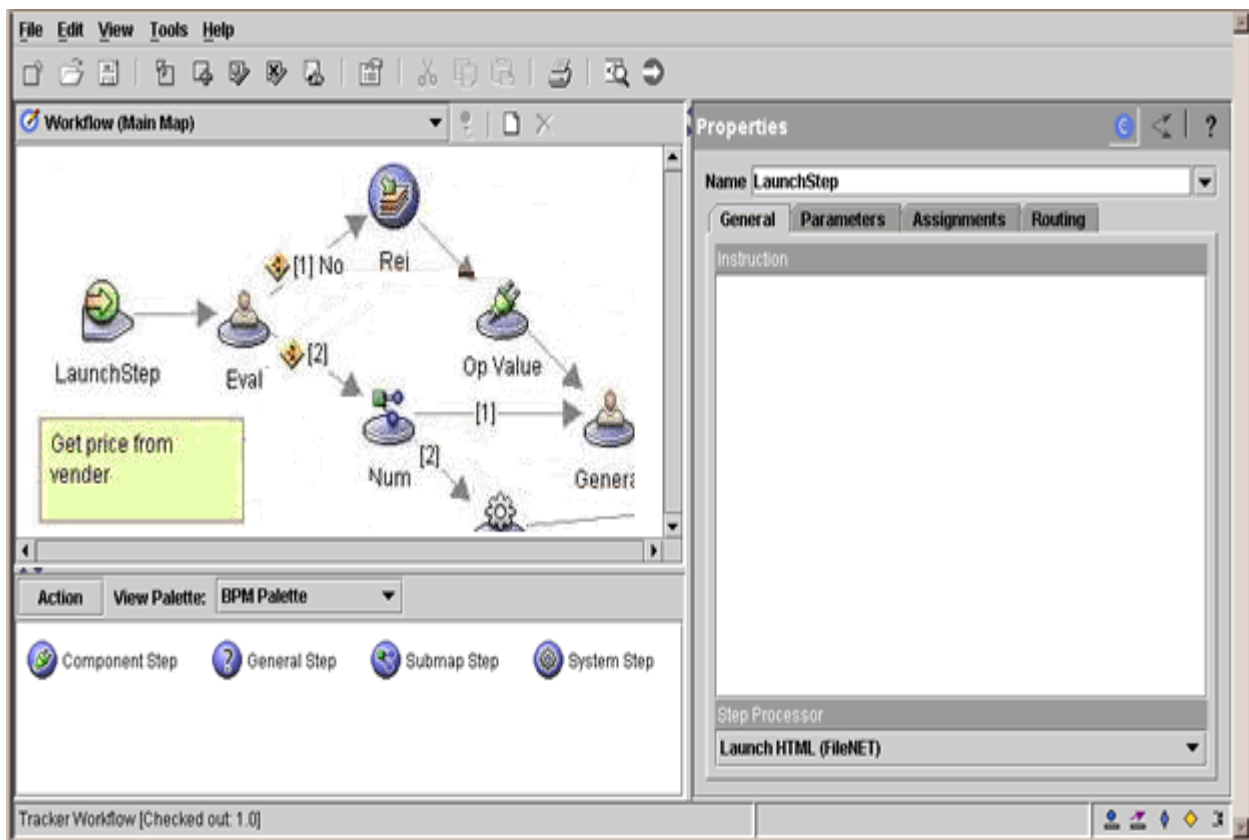
From Process Designer, you can:

- Create steps and routes using drag-and-drop.

- Define data fields that might be required while the process is executing, such as loan ID or customer number.

- Create placeholders for content-related attachments, including documents, folders, and custom objects, that users might need to view or modify during the process.

- Define expressions and rules that control routing and that can be used to set data field values.

- Specify users and groups who can participate in each step, including the ability to specify a user's role, such as the user who initially launched the workflow.

- Specify visibility of data fields and attachments for users who participate in a step.

- Pre-define steps for commonly required tasks, such as launching another workflow, waiting for another workflow to complete, and setting timers.

- Create submaps that simplify the workflow graph and that can be reused by being called from multiple steps.

- Inherit submaps, as well as data fields, attachments, and workflow group definitions, from other previously defined workflow definitions.

- Specify a programmatic component that will be executed for a step.

- Define milestones at key points in the workflow that can be used to monitor progress.

- Set up deadlines to indicate when a step must be completed.

- Validate the workflow and launch step prior to testing.

- Integrate and coordinate Web service operations for the enterprise and beyond with process orchestration.

- Process collections (multiple processes in a collection) that are saved as a single XPDL file.

- Add annotations to the workflow map. An annotation is a note that you can associate with any step or route in a workflow map. Annotations appear on the workflow map displayed in Process Designer; they do not appear in a running workflow at an individual step.

Connectivity

Rules integration

The Rules Connectivity Framework allows rule engine vendors to integrate with the process environment. By using rules integration, a workflow author can incorporate logic defined in business rules into a workflow. The business rules are created and deployed using rules software acquired from another vendor.

Rule sets allow you to separate the business rules from the business process. By having the logic external to the workflow definition, a business analyst can modify processes quickly and efficiently by adding, changing, or deleting rules.

Component Integrator

FileNet P8 provides an extensible integration capability for calling any component from a workflow step. For example, from a step you can make several calls to a Java class that updates information in a corporate customer database. This integration model makes it easy to create new, or reuse existing, components that a business analyst can then use in the graphical Process Designer tool.

The extensible model makes it possible to create connectors to a variety of technologies. FileNet P8 provides both Java and JMS adaptors that are ready for you to use without modification.

Process orchestration

Process orchestration lets you integrate and coordinate Web service operations for and beyond the enterprise. Orchestration allows you to tie XML Web services interfaces into end-to-end processes, which can include and combine internal (front- and back-office) and external (partner or customer) processes. Parts of each single orchestrated process can be executed in separate business process management or workflow engines, each under the control of different areas of an organization, and each with different sets of users and domains of trust.

Orchestration setup and management is performed through the Web Services panels in Process Configuration Console, Process Designer, and the designer's Web Services Palette. Orchestrated processes can be called from the Process Designer user interface or from an external client (such as a .NET or Java Web Services framework) using the standard request-response Web services, and can be secured with an authentication flag.

FileNet P8 orchestration supports the browsing of local drives or UDDI repositories to select XML Web services. It also supports the use of XPath for extracting information from the responses and for string replacement functions.

Work management

Work management provides users with a set of tasks that they need to perform. Work is managed in several types of queues, which are database structures that hold work items. User queues (called My Inbox in Workplace) hold work items waiting to be processed by a specific user. Public queues hold work items that can be completed by any user in the assigned group. Users processing work from such a queue can save work to their personal inbox so they can return to it later (ensuring it will not be processed by another user in the interim).

Other queues are used by the system and applications. Component queues hold work items that can be automated by using external components such as Java classes. System queues hold work items that are undergoing or waiting for certain processing by Process Engine software.

Task inbox and workflow tracking

The Workplace application and IBM FileNet P8 Portlets provide user interfaces that enable users to process their work items, view workflow milestones, examine workflows that they've launched, and track workflows for which they have been assigned responsibility. The following image shows a user's task list:

Task reassignment

Users who have been granted the ability to do so by the workflow author can manually reassign processing to another named user at any step. When reassigning a step, two options are available:

- **Delegation** – Returns an item of work once the delegated user has completed the required work.

- **Abdication** – Sends the work on to the next task once the reassigned user has completed the task.

In both cases, all of the actions taken are recorded in an audit log, which allows for necessary process deviations while retaining the same level of audit control over the process and enabling a greater level of flexibility in the manner that work is processed. In addition, it reduces the necessity to "over-engineer" process definitions and reduces the effort and time required for deployment.

E-mail notification

Users can be notified by e-mail when process-related events occur. An e-mail can be sent:

- When a new task is assigned

- As a reminder for completing a task

- When a deadline has expired.

Users who track workflows can be notified when:

- Workflow exceptions occur

- When a step has reached a deadline

- When a workflow has reached its deadline

- When the tracker has been assigned a task

Users can specify their individual notification preferences in Workplace.

Milestones

Milestones provide a way to expose the status of a workflow to users who do not need, or should not have, visibility into every step that makes up a workflow. For example, during loan approval, different milestones can be shown to the customer and the loan officer.

The following example illustrates how milestones might be used in a lending application template:
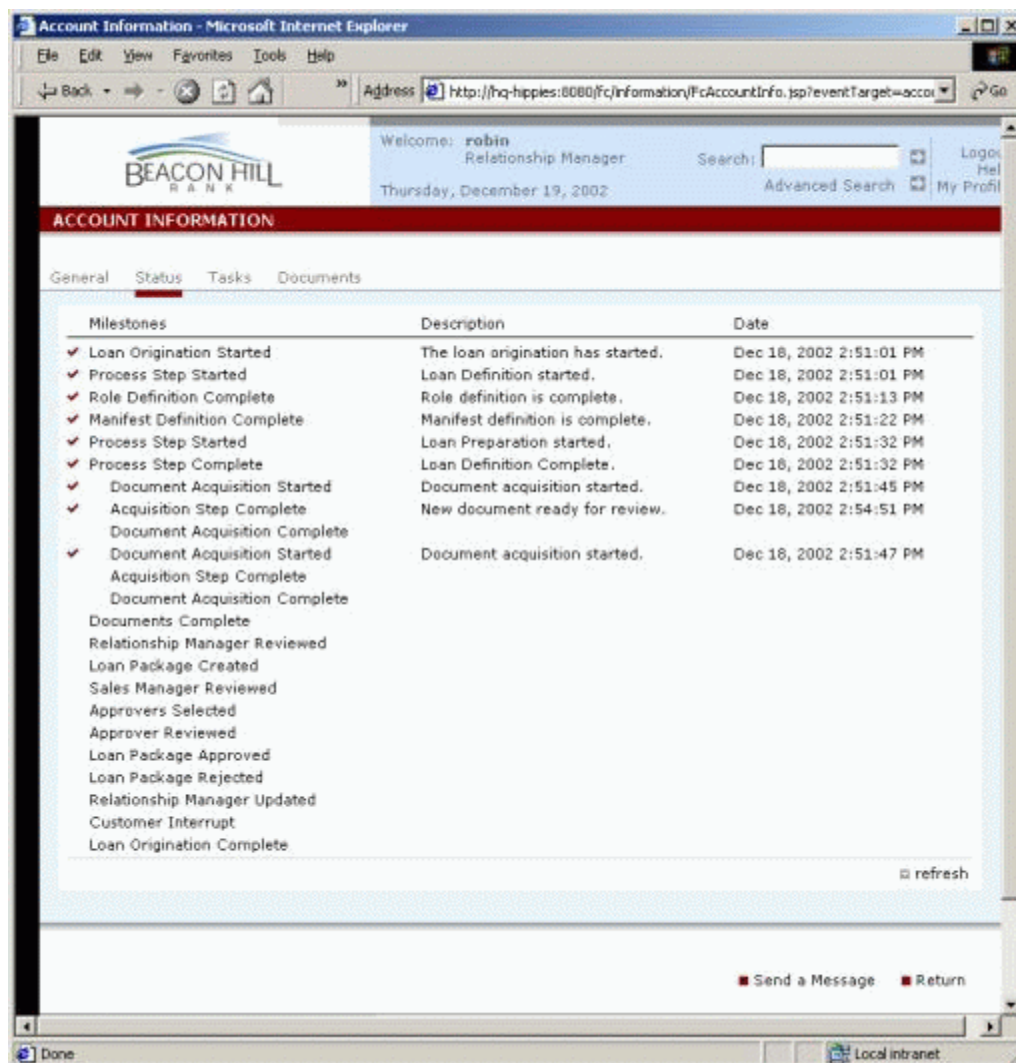
Monitor work in progress

Process Engine provides graphical tools to monitor and manage workflows in progress. A user assigned to track the progress of a specific workflow can use the Process Tracker application to monitor progress, manage participants, and modify data fields, attachments, and workflow groups used in the workflow, as well as complete or delete related work. On a bigger scale, an administrator can use the Process Administrator application to search for workflows based on a variety of criteria, then (individually or in a batch) assign or reassign work, change workflow attributes, or otherwise intervene in a workflow's normal processing.

Online configuration changes

As existing processes change and new processes are defined, it might be necessary to make modifications to the system (for example, create new work queues or add new queue fields). You can make these

changes (using Process Configuration Console or the corresponding APIs) while work is being processed without requiring users to log off the system or restarting the server software.

If you make the change using Process Configuration Console, the number of affected items is displayed. This provides a reliable indicator of the impact of the change, allowing you to decide whether to make the change immediately, or to postpone it until the performance impact to the system can be minimized.

Import and export configurations

You can easily transfer copy configuration settings from, for example, a development system to a production system by exporting the settings to an XML file and importing them in the new system. The exported file contains system-wide properties, such as specified Web applications, step processor settings, and event logging options, as well as all queue, roster, and log definitions including exposed fields and index settings.

Process analysis and reporting

FileNet P8 provides several ways to monitor and analyze your business processes.

Process Analyzer

Process Analyzer provides dynamic, chart-based analyses of workflow activity. You can analyze processes to:

- Determine cycle times for your workflow-enabled business processes

- Pinpoint bottlenecks that negatively affect production and service levels, costing you time and money

- Increase the return on investment of your workflow systems

Process Analyzer provides extensive analysis capability by enabling you to:

- Review the collected data to isolate problems and take direct control over the data elements being reviewed

- Generate chart-based reports on statistical information gathered by Process Engine

- Create and save custom charts for future use

FileNet P8 leverages OLAP (On-Line Analytical Processing) technology to provide process analytics. OLAP is a database technology that is optimized for querying and reporting, enabling fast analysis of multi-dimensional data. Excel provides the user interface for interacting with the data. Excel provides extensive charting capabilities with direct access to the OLAP database. In addition, Excel provides a low cost-of-ownership solution and a familiar user interface for analysts. FileNet P8 provides a set of standard reports that help you analyze the following areas:

- **Productivity –** Measures wait, processing, and completion time of the work items at the various steps and queues.

- **Queue Load –** Measures the number of work items added to, completed, and currently left at a particular queue or step.

- **Work Load –** Measures the number of workflows created, completed, and currently active. It also measures the average processing time of the various workflows.

- **Work in Progress –** Provides a real-time view of the currently active work items by measuring the current work item count at various steps and queues, and measuring time that the work items have so far spent at the current step.

- **Workflow in Progress –** Provides a real-time view of the currently active workflows by measuring the current workflow count, and measuring the duration of the workflows.

You can extend this reporting functionality by gathering business-specific process data and modifying the reports that FileNet P8 provides. The following report graphically shows the amount of work completed by each user.
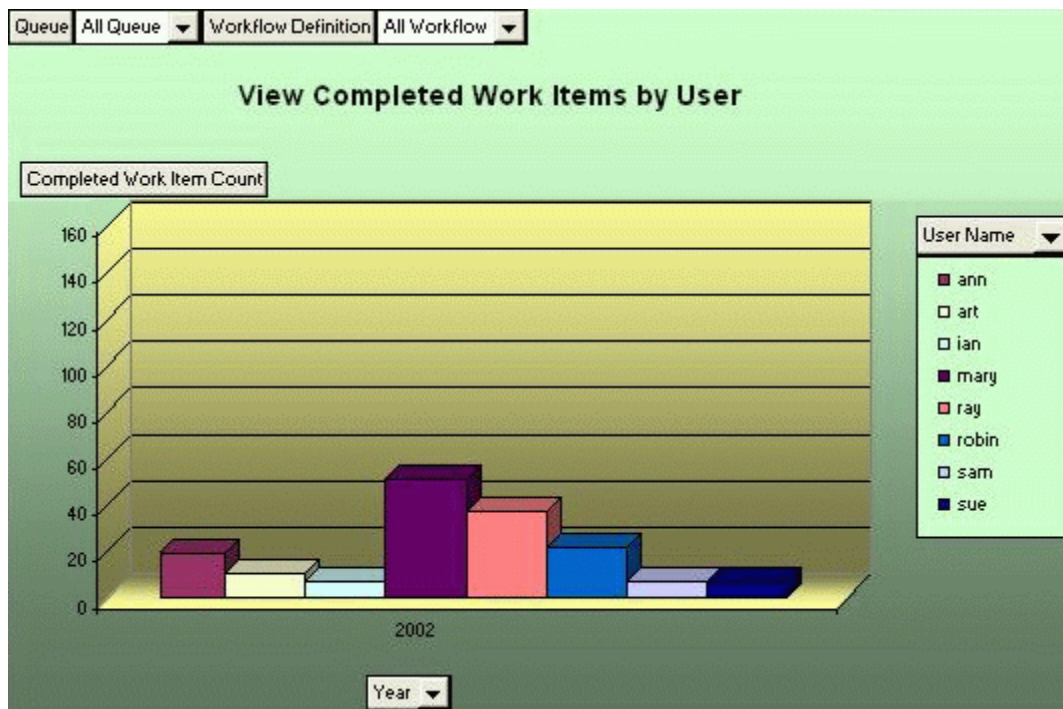
The same historical data can be viewed and analyzed in many ways. The following charts show four views of the same historical data.
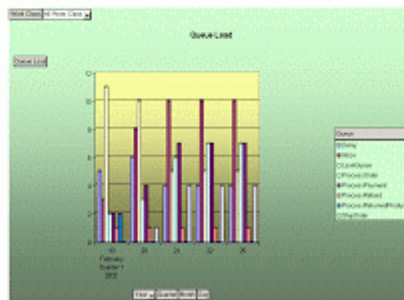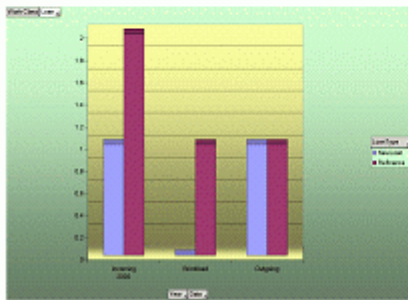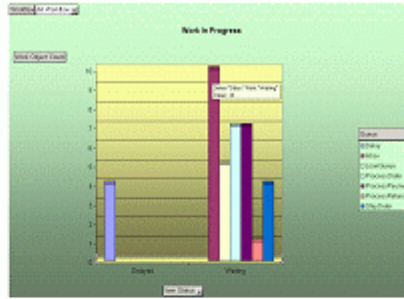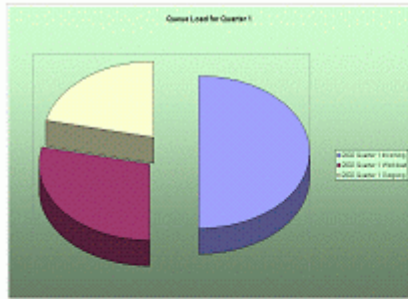
Business Activity Monitor

Business Activity Monitor (BAM) lets you monitor various aspects of the in-progress business operations. The business operations can be activities that are managed by Process Engine or activities that are managed by other enterprise applications. Using standard connections such as JMS, JDBC, Web Services, and EAI, you can connect to and collect data from external enterprise applications. You can configure BAM to collect and correlate data from various data sources and generate a holistic analytical view of the business operations. You can also configure the software to generate in-memory database views and OLAP cubes from the correlated analytical data. You can configure dashboards to display information stored in the views and cubes, and create thresholds (that is, Key Performance Indicators) and configure BAM to alert or perform actions if a threshold is violated.

WebSphere Business Monitor

IBM WebSphere® Business Monitor lets you monitor active IBM FileNet BPM processes. IBM FileNet P8 BPM software includes a CBE adaptor, a basic BPM Monitor Model, and some basic dashboard views that you configure and deploy to the WebSphere Application server and WebSphere Business Monitor server. The Business Monitor Dashboard provides customizable graphical views of data produced by running FileNet BPM processes, and includes tools to analyze trends and generate alerts based on key performance indicators.

Process simulation

Taking analysis one step further, simulation allows for the further refinement of business processes by using historical data combined with "what-if" scenarios to simulate and test business processes. An analyst can test different scenarios that might improve the business process before changing the process in a production environment.

Analysts create scenarios using the Scenario Designer—a tool that leverages the same user interface paradigm as Process Designer. A scenario specifies parameters that are used to simulate the workflow, including timeframe, work arrival patterns, resource allocation, job duty creation, costing metrics, and work shift creation and assignment. Scenarios are saved into an object store so that they can be executed and refined over time.

Workplace XT and Workplace

FileNet P8 includes two application environments to provide users with enterprise content management (ECM) functionality:

- **Workplace XT** – Workplace XT is the next-generation FileNet P8 platform Web application. Building on a familiar browse and search interface, users can quickly become productive in their content management environment. Workplace XT provides the tools (such as Process Designer and entry templates) by which business application designers can access the document management and business process management features of FileNet P8.

  The following image illustrates the Workplace XT browse page.
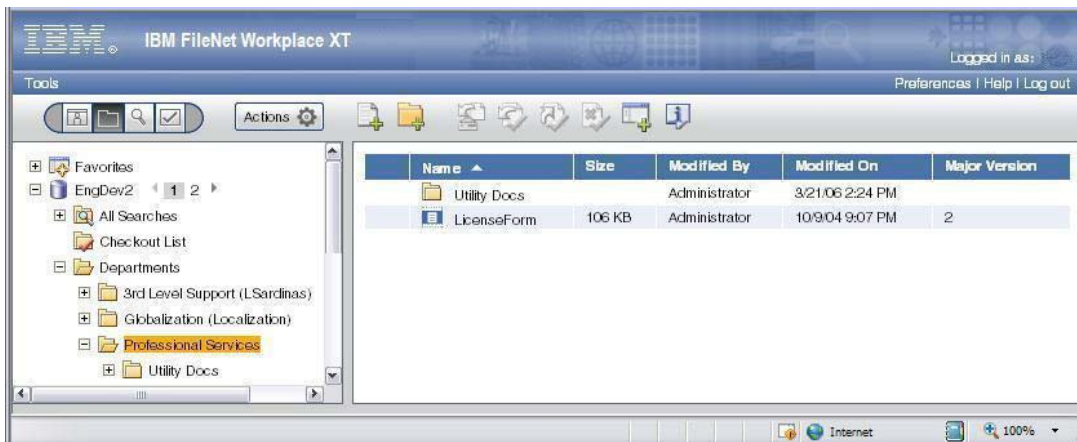
  Workplace XT automatically provides the following folders:

  - A Favorites folder where you can store links to frequently accessed folders and documents.

  - An All Searches folder that lists all predefined searches.

  - A Checkout List folder that displays all documents currently checked out to you.

- **Workplace** – Workplace is the previous-generation FileNet P8 platform Web application. You can customize Workplace for your user needs, or use it as a sample for custom application creation.

The following topics give an overview of the Workplace XT and Workplace application capabilities.

Convenience

The My Workplace portal view allows you to view several pages at once and can include a portlet for external Web sites such as your corporate Web site. For example, from My Workplace you might view items in your Inbox, browse an object store, review search results, review the list of your launched workflows, review the list of work items in a specific work queue, and see the authoring tools all at the same time, as shown in the following image:

Document management

You can perform a broad range of document management tasks, such as adding, retrieving, updating, versioning (creating different versions of a document), and deleting documents. Depending on the role and privileges assigned to you by the system administrator, you can easily:

- Add and delete documents, searches, and templates.

- Update and modify the properties (metadata) and security of the documents and objects stored in the FileNet P8 system.

- Browse and search for folders and documents.

- Create renditions of documents in other formats, and publish documents to multiple storage locations.

- Create templates for entering, searching for, and publishing objects.

Business process management

Depending on your role, you can perform many business process management tasks, including the following:

- View, perform, and reroute tasks, and launch and track workflows.

- Define and manage business processes and configure how those processes are launched.

- Examine business processes and simulate process changes to validate and assess the improved process.

Configuration

You can configure Workplace and Workplace XT to meet application needs by modifying site and user preferences. You can customize either application environment by modifying text-based input files (for example, XML) or, for Workplace, by modifying the source code. The following sections highlight examples of configuration and customization using these approaches.

Configuration using site and user preferences

Site preferences are configuration options that apply to all users, and can be set by administrators. User preferences are configuration options that apply to a single user. Configuration using site and user preferences is fully backward compatible when upgrading to a new release or installing a new fix pack level. The product documentation contains detailed descriptions of the many preferences that can be set. The following list is not comprehensive, but contains examples of changes that can be made by setting site and user preferences.

- Configure listings to display in either a detailed view or a magazine view. Detailed view shows a list of properties specified by the administrator in a tabular format. Magazine view shows a phrase that is constructed by concatenating properties along with additional text as defined by the administrator.

- Control which page displays first when a user launches the application—for example, a user can start at the Search, Browse, or Tasks page, depending on how the site preference is set.

- Override site preferences with user preferences that control what page displays when the application is launched.

- Change what a user initially sees when browsing and searching: either all object stores or specific folders or searches that have been designated as user-defined shortcuts.

- Add links to sites or tools outside of the application.

- Define how members of different access roles can access features. Users can access specific features and commands based on their access role. Members of a specific access role also can explicitly be granted or denied access. You can use access roles to prevent users from viewing specific pages such as Search, Browse, or Tasks. In addition, you can use access roles to limit user access to actions that are available on specific object types such as folders or documents.

- Configure different portal pages for groups of users that have similar needs. Users can also create personal pages. My Workplace portal pages provide quick access to several views of My Workplace from one page.

- Users can create a My Workplace page with customized content.

- Users can define default search criteria.

Customization using configuration file modifications

Both applications include several text-based configuration files that you can modify to customize the appearance and behavior of the application. When upgrading to a new Application Engine or Workplace XT version or fix pack level, these customizations are not automatically upgraded and therefore some planning is required to ensure that the customizations made in the previous version are also applied to the new version. Because the customizations are restricted to configuration files, redeploying any changes is a straightforward and simple process. The following list contains examples of customizations that are possible by modifying text-based configuration files.

- Configure the actions a user can take (for example, "Check In" or "Download") by modifying the actions descriptor XML files. These files contain the definitions of actions displayed in the context menu, multi-select menu, and on Information pages. You can also define the conditions under which the actions appear.

- Register Java Servlets that support retrieval of external documents whose content resides in another repository.

Application development

The FileNet P8 platform includes Workplace source code; therefore it is possible to modify the source code directly. Developers can use Workplace as a development base for building and testing processes and for customizing Workplace to add functionality. It is important to note, however, that modifications to source code are not backward compatible when the FileNet P8 platform is upgraded or when fix pack releases are installed. The following list contains examples of customizations you can make by modifying source code.

- Configure user tokens to allow single sign-on. Single sign-on means that FileNet P8 authentication tokens are shared between multiple Workplace and custom applications, thereby eliminating the nuisance of multiple logons for different applications. Multiple Web Application Toolkit-based applications can pass user tokens between one another. Tokens can be used in a number of scenarios, including Workplace actions, Information pages, and application integration. (Using the Web Application Toolkit, developers can also code their custom applications to handle user tokens.)

- Customize the Properties view into the provided information page or into a custom page. The administrator defines criteria on which the custom Properties view is launched, based on the content type or class ID. So, for example, documents of different content types, or folders of different class IDs, might have a different Properties view, while other views (Security, Versions, and so on) would be the same.

- Integrate viewers that support viewing a wide variety of document formats and enable users to mark up the documents.

- Add functionality to the existing Workplace capabilities, such as additional operations for selecting multiple documents and sorting documents.

- Display data from other systems, such as adding a column that shows data related to the document but which is stored in a separate application.

- Modify the look-and-feel by changing graphics, logos, and style sheets.

- Remove functionality that is not required for the application—for example, remove the Author page or restrict the user to browsing a single object store.

Application integration

FileNet P8 provides tools to help you integrate with a variety of applications acquired from other vendors.

Microsoft Office integration

FileNet P8 integrates with Microsoft® Office and Outlook so users can manage documents, e-mail, and attachments that are within a FileNet P8 object store directly from Office. FileNet P8's Outlook profiles provide an easily configurable, streamlined mechanism to capture e-mail and e-mail attachments and store them in the Content Engine. FileNet P8's Outlook support also broadens search and records management capabilities by automatically capturing e-mail transmission data, such as the To and From fields. With Office support, a user can add an Office document (such as an Excel spreadsheet or Word document) to the FileNet P8 object store by selecting tasks from familiar menus, as shown in the following image of Word integrated with Application Integration BCS.

SAP integration

IBM FileNet Application Connector for SAP R/3 (ACSAP R/3) is a modular ECM solution that seamlessly integrates with mySAP applications through the SAP ArchiveLink and Kpro interfaces. It provides outbound archiving and retrieval for SAP-generated business documents, SAP database data and reports, as well as inbound archiving and retrieval for documents generated outside of SAP. It supports "early, late, and simultaneous" document linking scenarios, including barcode processing. In addition to satisfying typical SAP R/3 document-enabling use cases, document linking to FileNet P8 object stores enables customers to

use IBM FileNet ECM solutions to improve and extend the business processes beyond the confine of SAP applications.

Portal integration

FileNet P8 provides portlets to deliver various areas of functionality. Portlets are user interface elements that can be plugged into portals such as BEA WebLogic and IBM® WebSphere®. Workplace, the FileNet P8 Web user interface, also uses these portlets to allow users to customize their interfaces and be able to view large volumes of data. Example portlets include an InBox, Public Work Items Queues, internal and external Web sites, folder hierarchy, Quick Search, Active Workflows, Administration Tasks and more. These portlets allow quick and intuitive access to common content and process management capabilities by using a configurable look and feel.

FileNet P8 provides portlets for commonly required content and process functionality within portal products. Portlets, which are user interface elements, provide the following functionality to the user:

- Browse

- Quick Search

- Public Inbox

- Inbox

- Author

SharePoint integration

FileNet P8 and Microsoft SharePoint technologies are complimentary solutions that, when used together, provide an environment where business users can easily collaborate on work-in-process documents and tasks, yet comply with content and compliance initiatives for the enterprise. SharePoint users can continue to use SharePoint's easy-to-use and familiar interfaces while FileNet P8 provides a robust content, process, and compliance infrastructure that is transparent to users. With this approach, business users can leverage Microsoft Office SharePoint Services sites and SharePoint Portal for the majority of daily content, process, and compliance activities without having to learn new applications or perform additional content or compliance management tasks.

FileNet P8 embraces and extends SharePoint Products and Technologies to allow SharePoint content to interoperate with an enterprise's existing heterogeneous computing environment. Additionally, FileNet P8 enables organizations to rapidly deploy SharePoint to address key enterprise business and IT requirements surrounding collaboration, while providing a powerful content, process, and compliance infrastructure to ensure SharePoint activities are accessible to authorized users, completely secure, and placed under full lifecycle and compliance management.

More and more companies are realizing their knowledge workers and resulting knowledge base can be a formidable competitive advantage. With SharePoint Services, organizations can easily facilitate collaboration among work groups and project teams. And with FileNet P8, they can gain more control and

leverage over that vital content through an integrated content and business process and approach, increasing productivity while meeting enterprise security, storage, and compliance requirements.

eForms management

IBM FileNet P8 eForms allows customers to create and manage online forms in a Web browser that can appear the same as hard copy paper forms, yet are more advanced, easier to use, and less costly to implement. Online forms created using eForms Designer can reduce data entry time, improve data accuracy, and are the first step in streamlining and accelerating transaction processing.

eForms provides the ability to process electronic forms as stand-alone documents or to use an electronic form as the user interface for business process workflow. eForms allows creation of high-fidelity, highly intelligent electronic forms without custom programming.

eForms provides the following features and benefits that are ready for you to use without modification.

Forms as a type of document:

- Can be filled out, saved, and stored using Workplace.

- Are tightly integrated with other FileNet P8 features such as document classes and entry templates.

- Use document property mapping to automatically populate form data document metadata with values from the form when the form is saved.

- Incorporate digital signatures.

Forms to process workflows:

- Include all of the features of document forms and allow tight integration with Process Engine and workflows.

- Automatically map values in the form to workflow data fields for use in process decisions.

- Map values from workflow data fields to form fields to pre-fill the form with process-specific data.

- Act as the user interface to workflow steps:

  - Features such as data checking and error messages make the workflow process easy to manage for the user and minimize data entry errors.

  - Fields on the form can be linked to workflow data fields without having to write HTML code and JavaScript™.

  - Page display can be controlled at each workflow step.

The following image shows an expense form acting as the interface to a workflow step:

Form Policy Wizard

A form policy is an object store document that describes how a particular form is to be used. There are two types of form policies: one for document forms and one for workflow forms. It is the form policy document that a user clicks when he wants to create a new form.

The form policy wizard is a point-and-click interface within Workplace that allows you to create form policy documents. This interface requires no coding and allows you to specify many form behaviors including:

- Which form template to display.

- The Entry Template used to store the form data in the object store.

- The mapping of form data to document properties.

- Simple customization of the form window's interface to add HTML banners and sidebars.

In addition, for workflow policies you can specify:

- Whether the form is the primary content of a step, displayed directly in the step processor user interface.

- Which pages of the form are to be displayed at a particular step.

- Which form data fields are mapped to exchange information with workflow data fields. Mapping form data allows the workflow to make decisions based upon the state of data filled into a form.

The image that follows shows the outline of a simple workflow with a single form attached to every step as the primary content.
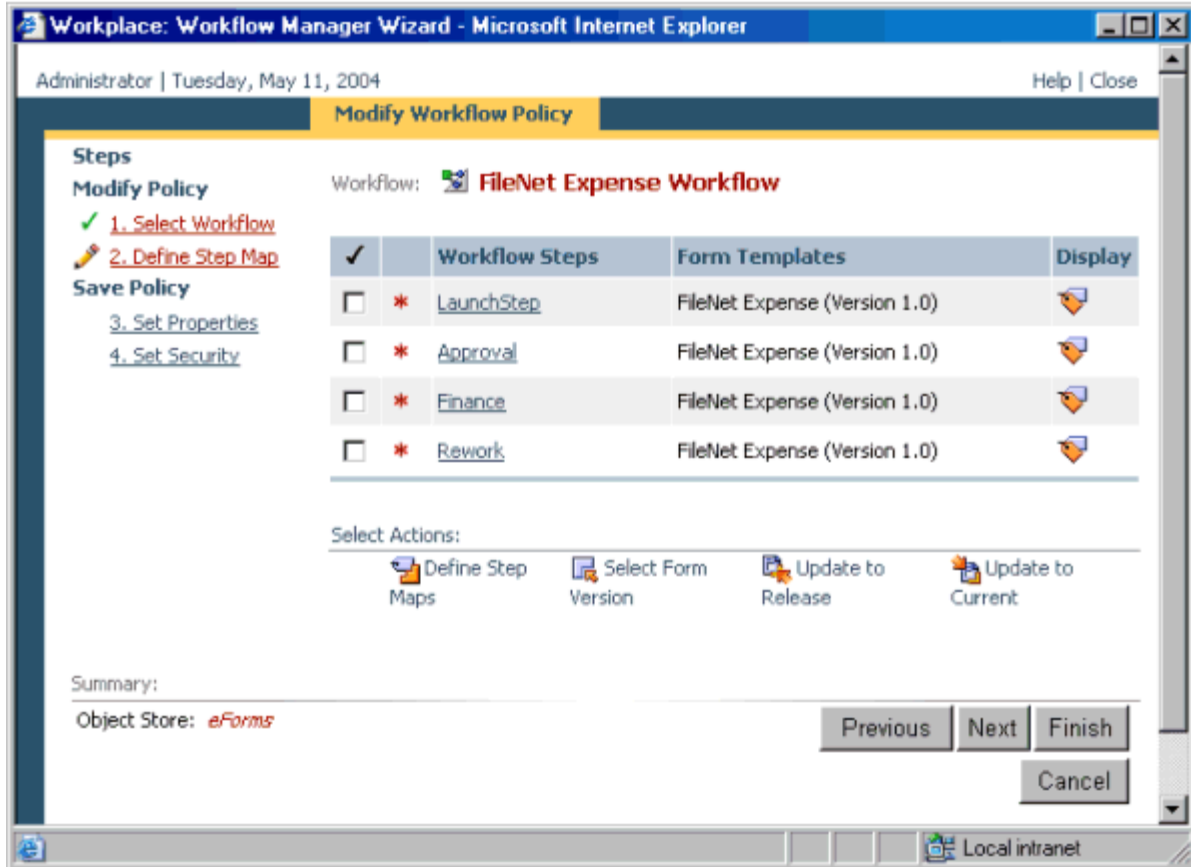
eForms Designer

IBM FileNet eForms Designer is a powerful forms design application that runs on the client desktop. You can use it to:

- Create any form.

- Add intelligence such as automatic calculations, data validations, and formatters to the form template to assist users while they are using the forms and help to prevent data entry errors.

- Configure connectivity with external data sources to enable data lookups that reduce user data entry.

You can use the integration features to easily manage form templates in eForms Designer, such as conveniently checking form templates in and out and adding form templates to an object store. Integration also provides centralized content management, version control, and security for your form template.
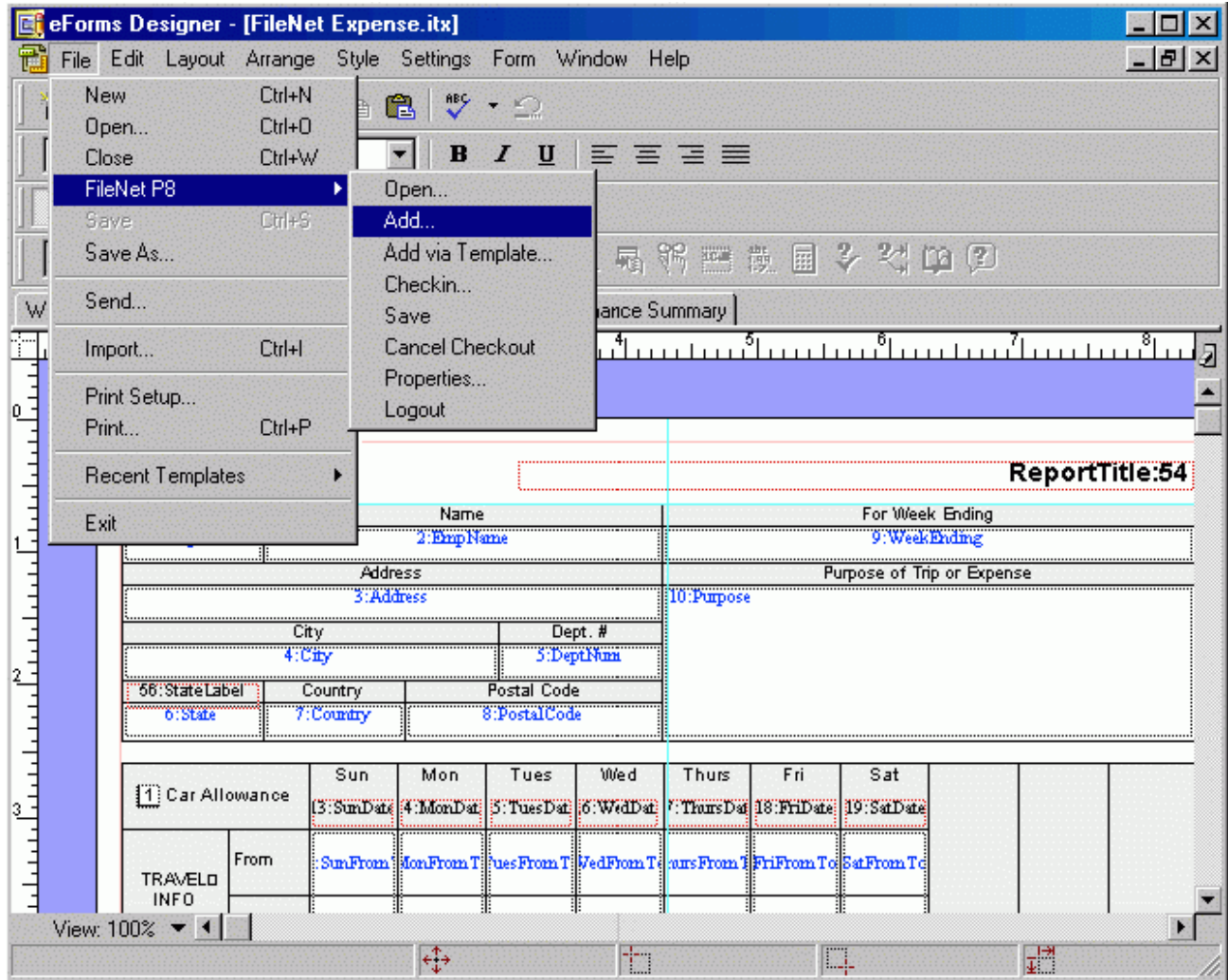
Records management

Designed to solve today's process-oriented enterprise records management and compliance needs, IBM FileNet Records Manager is a records management solution that can help companies manage risk through effective, enforceable records management policy, for achievable and cost-effective compliance.

Records Manager is a fully integrated records management solution that an organization can use to capture, safeguard, and access business records and processes, relate them to other relevant content, and cost-effectively archive and manage records according to automatically enforced administrative, legal, and regulatory guidelines. Records Manager helps an enterprise comply with regulations, enforce policies, and avoid potential legal and procedural penalties while streamlining and automating its processes, connecting with its different information systems, and accessing and managing all forms of content.

IBM's innovative FileNet ZeroClick technology is essential to delivering automated records declaration, classification, and administration. For more information, see the topic in this section.

By leveraging the core FileNet P8 content and process management capabilities, you can use the Records Manager services and applications to:

- Manage file plans and record disposition schedules using a Web-based application that is integrated with Workplace.

- Declare documents as records when adding them to the FileNet P8 object store from Workplace and Office.

- Automate record declaration using Content Engine event actions, process components, and custom applications.

- Manage documents stored in FileNet P8 object stores.

- Manage physical records, which are stored externally (that is, in non-FileNet repositories) and referenced in the Records Manager system.

- Leverage architecture that provides for managing records that are stored in other repositories.

- Create record folders to contain related records.

- Search for records.

- Place records on hold.

- Generate reports.

- Audit records.

- Manage the review and disposition of records by using provided workflows.

- Customize the workflows to meet specific business requirements.

E-mail management

E-mail messages are considered corporate records. As such, they share the same status as other business documents. Email Manager is a server-based e-mail management system that is seamlessly integrated with FileNet P8 platform and with business applications.

Using Email Manager, an organization can manage e-mail content as a part of a comprehensive Enterprise Content Management infrastructure. Email Manager, used in conjunction with any FileNet P8 object store, enables organizations to capture, organize, monitor, retrieve, and share e-mail content when and where it is needed, including using captured e-mail content to initiate, and participate in, manual and automated business processes. Using Email Manager, you can also search for and dispose of e-mail as you do for other corporate records. Because Email Manager simplifies and automates the process of capturing e-mail messages as business records, it supports proof of compliance with government regulations.

Email Manager effectively addresses a broad set of e-mail management issues including:

Storage Utilization – Email Manager's duplicate e-mail management capabilities help eliminate the capture of duplicate e-mail messages and ensure that copies of e-mails -- with or without attachments -- will be written to a FileNet P8 object store only once. By eliminating duplicate e-mails before they are added to an object store or storage device, Email Manager provides better utilization of storage and simultaneously lowers overall storage costs.

Records Management – Email Manager's dynamic records classification capabilities, and tight integration with Records Manager, enable an automated declaration of e-mail as records to address compliance and corporate records management requirements.

Electronic Discovery – Email Manager's targeted capture and filing of e-mail messages greatly improves the ease of retrieval in response to electronic discovery requests. Email Manager's personal storage file (PST) management option also ensures that all business critical messages, even those stored off the e-mail server in PST files, are appropriately captured and discoverable when needed.

Business Process Acceleration – Email Manager manages business critical e-mail messages, such as customer service inquiries, in the same object store as other corporate content, allowing an integrated and streamlined approach to managing e-mail, automatically launching related business processes, and improving responsiveness and decision making for the enterprise.

FileNet ZeroClick

Through a unique understanding of business processes and the events that trigger functionality and workflows, IBM has developed a FileNet ZeroClick solution for identifying, capturing, and managing relevant business information and processes, which become the basis for corporate "records."

Once captured, these records automatically inherit a schedule for retention and destruction, which the records management industry refers to as "disposition."

IBM provides integrated, automated (by using FileNet ZeroClick), and accurate records capture, classification, and management, which results in the following benefits:

- Automates the process of identifying, automatically capturing, and accurately filing all records.

- Stores records once and virtually replicates them, if needed, in multiple files.

- Built around the process and not just the record.

- Provides proof of adherence to process (plus content), now required for Sarbanes-Oxley, through audit and chain of custody.

- Does not require extra steps by business workers, thereby reducing user-related errors as well as time and cost factors.

System management

FileNet P8 provides a complete set of system administration tools that allow for monitoring, validation, and configuration changes from a central location with a dispersed deployment. These tools, described in the following sections, can be used to manage the entire system.

FileNet P8 also integrates with systems management tools like Tivoli®, which allow for orchestration and management of system components and resources on an enterprise level.

Dashboard

IBM FileNet Dashboard provides a central collection system for application-specific data (such as RPC counts and durations, application-specific events and metrics), environmental data (such as configuration information and version and fix pack levels), and operating system statistics (such as CPU load, memory utilization, and disk and network I/O). You can analyze the collected data using Dashboard components or export the data for analysis using industry-standard monitoring and reporting tools.

Using the Dashboard, you can monitor all of your FileNet P8 servers from your desktop. Dashboard provides the following features:

- Displays environment information received from monitored servers, including system specifications, operating system release levels, and FileNet P8 release levels.

- Captures live system and performance monitoring data from FileNet P8 products.

- Allows real-time graphing of received performance data.

- Archives data gathered for future reference and comparisons.

- Displays messages received from applications and when connections to applications are lost.

- Automatically checks for urgent messages at prescribed times, allowing for early detection and correction of performance problems.

- Exports comma-separated value (CSV) text files that you can import into tools such as spreadsheets for further analysis. You can electronically send exported files to a vendor or service provider for analysis, significantly speeding problem resolution time.

System Monitor

IBM FileNet System Monitor automates the monitoring of the entire FileNet P8 environment including FileNet P8 software components, application servers, databases, log files, network communication and devices, and the full range of IBM storage repositories. In addition to monitoring the core FileNet P8 components, System Monitor can monitor homegrown applications and products acquired from other vendors such as Adobe®, Kofax, and Captaris RightFax.

System Monitor notifies you in real time about critical and fatal problems before they negatively impact your users. System Monitor can also identify potential problems that might arise, enabling you to take preventive action before an outage occurs. The result is greater customer satisfaction and increased system uptime

that results in better meeting your Service Level Agreements. In addition, System Monitor automates over 75 manual administrative tasks, enabling you to increase administrator productivity.

System Monitor enables operators to handle FileNet P8 environments without having in-depth knowledge of, and root access to, the application itself. Automation mechanisms and knowledge provided by System Monitor allow problems to be solved by Operations and 1st Level Support while easing the burden for 2nd Level Support. The application specialists in 2nd Level Support can focus their efforts on the further development and optimization of FileNet P8 components, instead of doing unproductive routine work in daily administration, which adds little value to the business.

System Monitor provides inbound monitoring of the FileNet P8 components leveraging FileNet P8 APIs. System Monitor monitors more than 600 system and performance parameters for the FileNet P8 platform. The inbound monitoring goes far beyond general FileNet P8 system status (up or down). System Monitor provides detailed health status of individual FileNet P8 components.

System Monitor provides a knowledge base for error conditions detected by the monitoring or the log file management. This information can contain a description of the error cause and a potential corrective action. The information is obtained from IBM FileNet P8 documentation and best practices in operating FileNet P8.

System Monitor provides reports to analyze historical event data, for example, for capacity planning or quality management. Reports can be created on demand or be provided as standard reports to identify trends in the FileNet P8 environment. The event data can also be analyzed by standard reporting tools. By analyzing the information, you can identify areas for improvement and deliver a better quality of service to your business.

Following is a detailed comparison matrix of the capabilities of System Monitor and Dashboard.

Feature	System Monitor	Dashboard
Inbound system monitoring of supported FileNet P8 components	☑	
Monitoring of FileNet Storage components	☑	
Pre-defined monitors for over 600 parameters for the IBM FileNet product family	☑	
Web console for creating, configuring, and viewing monitors, setting thresholds, alerts, notification schedules	☑	
Alerts to any device: Web console, e-mail, Blackberry, pager, SMS-based devices	☑	
Monitor non-FileNet P8 products and homegrown applications through configuration with no code changes	☑	

Feature	System Monitor	Dashboard
Monitor IBM, Windows®, Unix®, and other log files	☑	
Corrective actions on over 1200 Image Services and HPII error messages	☑	
Ability to define your own corrective actions based on error messages	☑	
Support for current and previous versions of FileNet suites and products	☑	
Integrates with full range of Enterprise System Management software vendors: BMC, CA, HP, IBM Tivoli, Microsoft™, Micromuse, NetIQ and others through SNMP and other methods	☑	
Fully integrated with IBM Tivoli Framework, Enterprise Console, IBM Tivoli Monitoring and Desktop so Tivoli customers can use their environment ("Tivoli ready")	☑	
Task Execution Manager for automating manual FileNet-specific administrative tasks on remote systems and complete, distributed environments	☑	
Configurable authorization levels for different administrator roles (for example, configuration or operation)	☑	
Dashboard for the real-time display of the health and performance of FileNet products to provide better visibility into application performance	☑	☑
Performance data can be archived or saved for historical analysis and management reporting	☑	☑
Dashboard has user-defined charts of summary and detailed performance data	☑	☑

Usage Reporter

IBM FileNet Usage Reporter provides near real-time and historical usage information to allow you to monitor system usage. Usage Reporter reports the total number of users logged in at a given time, as well as the number of distinct users logged in, allowing you to drill down to view the identities of the users. You can easily export the data to other reporting tools. The following image shows a sample detail peak usage report.

Enterprise Manager

Enterprise Manager is a Microsoft Management Console (MMC) based tool that administrators use to manage Content Engine. Enterprise Manager provides a graphical, wizard-based user interface to perform the following tasks:

- Create and manage object stores and file stores

- Create and manage classes and properties

- Create and manage security policies and default security for object authorization

- Create and manage lifecycle policies

- Create and manage event subscriptions

- Import and export objects and metadata definitions as XML

- Search and perform bulk updates on search results

- Configure XML classification

- Perform document management tasks (for example, checkin and checkout)

- Configure trace logging

Process Task Manager

An administrator can use Process Task Manager to manage and configure Process-related services on the Process Engine, Application Engine, Process Simulator, Process Analyzer, and Rules Connectivity Framework servers.

Application development and deployment

FileNet P8 includes an extensive collection of development tools that span the content and process management capabilities outlined in this document. These include:

- Graphical tools for defining and designing application components (such as processes, metadata definitions, searches, and templates).

- Java APIs for programmatic access to content and process capabilities.

- A .NET API for developing Content Engine applications.

- Content Engine and Process Engine Web Service APIs for building Web-based applications that can execute on a variety of platforms and can use a variety of languages and toolkits to access most of the functionality available through the Content Engine and Process Engine Java APIs.

- Integrations with leading portal vendors for building Web-based applications.

- User interface elements that can be reused in custom applications.

- Code module capabilities where java classes containing event action code are stored in the object store and as such are easily deployable.

- Several tools to help you move content between environments, which aid in the deployment process.

For more information on deploying FileNet P8 applications, see *IBM FileNet P8 Planning and Deployment Guide*. (To download the guide from the IBM support page, see Accessing IBM FileNet P8 documentation.)

Rapid development

IBM FileNet Business Process Framework (BPF) provides a highly configurable framework for rapidly developing BPM applications.

BPF extends FileNet P8 solutions in many ways for more flexible work delivery and added case management control. BPF supports broader data entry functionality and layout with built-in validation and data types. BPF supports improvements in user efficiency by a "get next" ability to push work to users, as well as making available the traditional in-basket selection approach. Plus, vital to creating and maintaining an audit trail, cases are kept intact, with the entire collection of originating and supporting documents instantly available to users even after the case has been closed.

The BPF methodology also enhances the flexibility of the FileNet P8 system. More comprehensive data entry options, such as Pick Lists and Reason Lists, are available. Also, the ability to perform a bulk import of user IDs into all of the FileNet P8 object stores supports productivity and efficiency improvements as users can log in once and have access to data in every facet of the FileNet P8 system.

Architecture

The following sections describe the FileNet P8 platform architecture from the enterprise level as well as each product component, including an overview of services provided by the component and the component architecture.

Enterprise capabilities

FileNet P8 components provide the enterprise-level capabilities that are required for solving critical business requirements. This section enumerates these product characteristics.

Open and extensible

To promote an open and extensible environment, FileNet P8 provides APIs for developing custom applications. In addition, the FileNet P8 architecture is built on industry standards such as XML and J2EE, and the following communication protocols: Lightweight Direct Access Protocol (LDAP), Hypertext Transport Protocol (HTTP), and SOAP.

XML for application development

FileNet P8 APIs, applications, and services support and use XML in many ways, including:

- **XML as a data format –** Many FileNet P8 structures are represented in XML, including workflow definitions, entry templates, search templates, publish templates, publishing assemblies, and site/user preferences.

- **XML for data exchange –** XML is used to import and export objects and metadata definitions.

For more information, see Managing XML Content.

Java 2 Platform Enterprise Edition (J2EE) support

FileNet P8 provides J2EE Application Components and System Components that operate in J2EE Platform Products (application servers) such as BEA WebLogic, IBM WebSphere, JBoss, and Apache Tomcat. (For a description of these component types, refer to *Java Platform Enterprise Edition Specification*, v1.4 on the Sun Web site at http://java.sun.com/reference/docs/.) In addition, FileNet P8 applications leverage the J2EE application model to build multi-tier applications that deliver the scalability, accessibility, and administration required by enterprise applications. Details about how FileNet P8 is used in a J2EE environment are covered in the remainder of this section.

Communication protocols

For more information on LDAP, HTTP, SOAP, and IIOP protocols, refer to the available documentation in the public domain and on the Internet.

Scalable

The FileNet P8 components support enterprise-level scalability with a multi-tier, distributed architecture. In particular, in order to accommodate various scalability needs, FileNet P8 offers both vertical and horizontal scalability solutions. The vertical scalability of a server can be defined as its ability to handle additional workload by the addition of a proportional amount of processing power. Horizontally scalable systems can handle additional workload by increasing the size of the server farm as the workload increases. To that end, following are some highlights of how FileNet P8 is scalable:

- Application Engine can be farmed (scaled horizontally) to support increased workload requirements.

- Application Engine can be scaled vertically by either running multiple instances of a single Application Engine version on a single server, or by configuring a single instance to leverage system resources.

- Content Engine can be farmed (scaled horizontally) to support increased workload requirements.

- Content Engine can be scaled vertically by either running multiple instances on a single server, or by configuring a single instance to leverage system resources.

- Process Engine can be farmed (scaled horizontally) to support increased workload requirements.

- Database software can be remotely configured to run on a separate machine, which offers several benefits:

  - The customer has maximal control over the way resources are allocated between the application and the database.

  - The database can be located on a system that is dedicated, tuned for the RDBMS, and separately managed by a database administrator (DBA).

  - Separation from the database permits application servers such as Content Engine and Application Engine to be farmed.

- Services that access a content object store can be distributed to any number of machines to handle heavy user access.

- Multiple object stores can be distributed to different databases and machines, allowing the same system to service multiple applications.

- Users can access content that has been distributed to different repositories by using cross-repository (that is, federated) search.

- Web sites can be easily deployed to multiple targets for handling large numbers of users.

- Multiple Rendition Engines can be configured to support large numbers of format conversions.

The following diagram shows how a FileNet P8 system might be farmed.
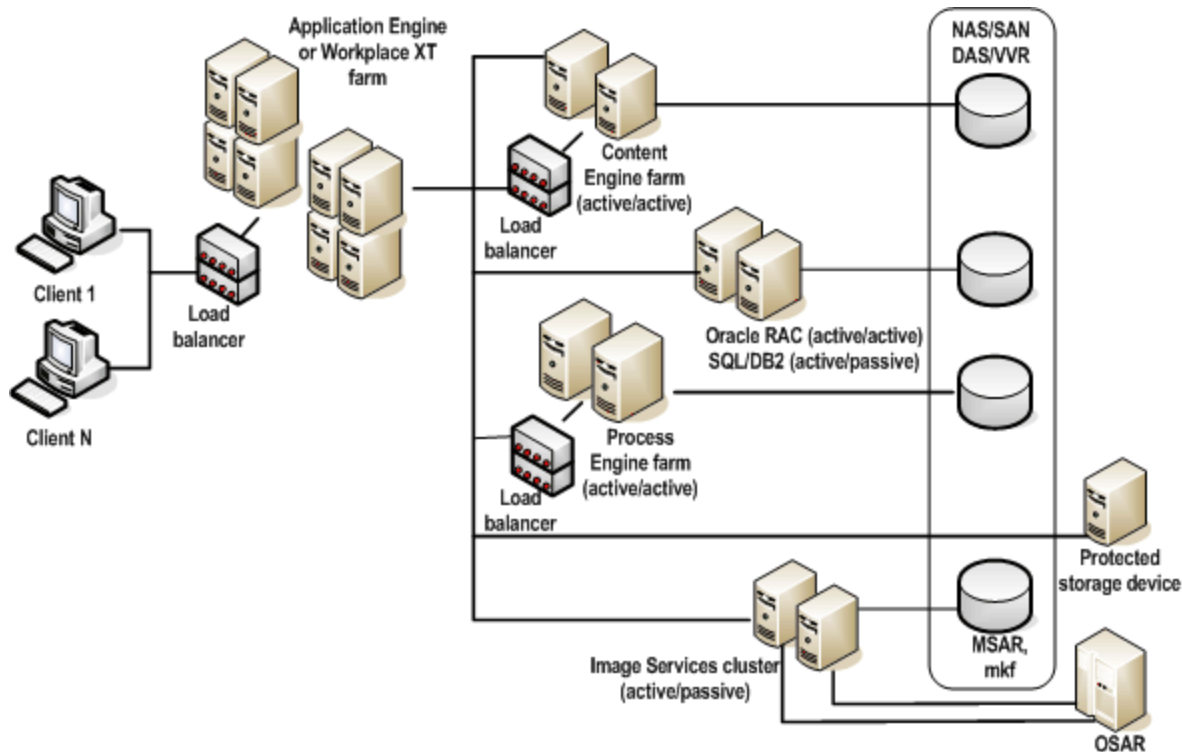
Scalability and Web server size

Best practices dictate that you precede large system deployments with capacity planning. During this process, information is gathered about the workload that the deployed system will be required to support. That information is then used as input to models for determining the CPU, network, and I/O capacity that will be required to accommodate the planned workload. Once that determination has been made, a second issue must be addressed: how should the required resources be partitioned?

High-availability concerns usually steer deployments away from having a single large server and toward smaller and more numerous servers. This direction is consistent with creating an architecture that facilitates software performance and scalability. Especially with regard to the Web-tier deployments, systems that use smaller servers tend to be better able to exploit their computing resources. For the majority of workloads, two single-CPU systems will be able to support more throughput than one dual-CPU system.

To eliminate single points of failure, and to maximize the use of available capacity, IBM recommends that Web-tier servers be single- or dual-CPU systems.

Highly available

The FileNet P8 family of products provides robust software services, a critical requirement for highly available systems. Software robustness, however, is not sufficient when systems can fail due to both hardware and software problems. IBM recommends using commercial high-availability solutions that address the failures that can occur in a complex networked environment. FileNet P8 components leverage several technologies to maintain availability in the event of a local component failure such as the loss of a

server, disk, or network device: farming at the Web server layer, and clustering technologies for backend services and databases.

Web server farms

Web server farms provided through hardware or software load-balancing technology enable high availability at the Web server tier. A farm is a set of load-balanced servers that are clones of each other, each actively providing the same service, with the same applications and same binding to servers in the tiers below them. Farms are best practice for server tiers that are relatively static in terms of content, because that makes it easy to maintain them as clones of each other. The FileNet P8 Web-based components have been certified to operate within Web and application server farms such as BEA WebLogic clusters and IBM WebSphere clusters. These types of farms provide for server redundancy with the added value of scalability, because all of the servers in the farm are active. Application server farms can be combined with hardware-based load-balancing solutions such as Cisco routers or software-based solutions such as Network Load Balancing (NLB).

A load-balanced Web server farm provides both better availability and better scalability than a single Web server. When a Web server fails, the load balancer automatically detects the failure and redirects user requests to another server in the farm, thereby keeping the service available. Administrators can increase Web site performance and capacity by simply adding servers to the farm.

Server clusters

For a complete highly available solution, Web servers and all supporting servers need to be configured for high availability. In a multi-tier architecture, this includes application servers implementing business logic and any servers in the data tier (for example, databases). These servers are typically data-driven, with large amounts of data, and a constant stream of new or modified data. Using a server farm approach with data replication is not as appealing for these tiers because of the difficulty in maintaining data synchronization in the face of frequent change. Instead, server clustering products are used which are based on the concept of shared data storage instead of a separate storage device per server. In this case, two or more servers share the same highly available data storage device, such as a SAN, NAS, or RAID disk array. The shared storage device incorporates redundant copies of the data but appears as a single shared drive to the servers, thereby avoiding the need for data replication between servers.

Only one server in a data-driven cluster is active at a time, unlike a server farm, in which all servers are constantly active. Thus there is no load-balancing requirement in a server cluster. Instead, clustering software is employed to detect a failure on the active server and trigger a "failover" to the passive server. The passive server is also running the clustering software in a standby mode, waiting to take over in the event of a failure on the active server. The clustering software thus has three responsibilities: monitoring the health of key applications and processes on the active server, stopping the applications and processes if a failover is required, and starting the same applications and processes on the passive server when it becomes active in a failover.

Disaster recovery

The previous section on high availability discusses putting into place systems and devices that assure your FileNet P8 system is available to your customers and employees with as little downtime as possible. High

availability addresses localized failures such as a hard disk crash or server failure. But what if your entire production system is lost due to a catastrophic event such as earthquake or flood? Disaster recovery goes a step further than high availability by planning replication of your entire data center.

The first step in disaster recovery planning is to decide which approach you want to take:

The low-cost option is to take backups of all application data at regular intervals (preferably daily) as well as less frequent full backups, and to keep copies of the data offsite. All data is then available, but to set up a replacement system after a disaster occurs can take days or weeks. The danger of this solution is that the application will be completely unavailable for a period of time after a disaster occurs and some amount of data (data changed or created after the last backup) is permanently lost. In addition, this solution is difficult to test on a regular basis, because the replacement system is assembled only after a disaster occurs.

A hot site recovery service acquired from another vendor is a more expensive solution that offers quicker recovery, based on dedicated hardware for the recovery system. This service consists of a shared resource environment provided for disaster recovery on a subscription basis. As with the preceding solution, periodic backups are stored off-site. In this case, though, hardware and other resources are standing by at a disaster recovery site shared with other clients of the service vendor. IBM ECM Lab Services offers this as a turnkey solution in partnership with Sungard. Sungard provides the server recovery facility, the hardware infrastructure, telecommunications, and workstation recovery capabilities, while IBM delivers dedicated technical support, software, and all necessary documentation. Regular testing is done to insure the validity of the recovery. This solution provides end-to-end service including both clients and servers.

The most expensive solution, but the one that offers the quickest and most complete recovery, is a redundant data center at a remote disaster recovery site, outside the potential disaster impact zone. The recovery site has a dedicated copy of the production system hardware, possibly without the extra servers that were needed to make the production site highly available. Rather than relying on data backups and restores, this approach typically employs a storage solution from vendors like EMC, Network Appliance, or Hitachi that enables data replication to the recovery site. At their best, these replication products can provide real-time protection of the data, so that the data— right up to the moment of disaster—is available at the recovery site. They also offer less costly options for replication that allow some lag in the replication, and hence a small loss of data in a disaster. For transactional systems like FileNet's, best practices call for the recovery site to be a hot standby as shown in the following diagram. This reduces the complexity of the disaster recovery solution and avoids conflict resolution problems between active sites. This solution can and should be tested on a regular basis.
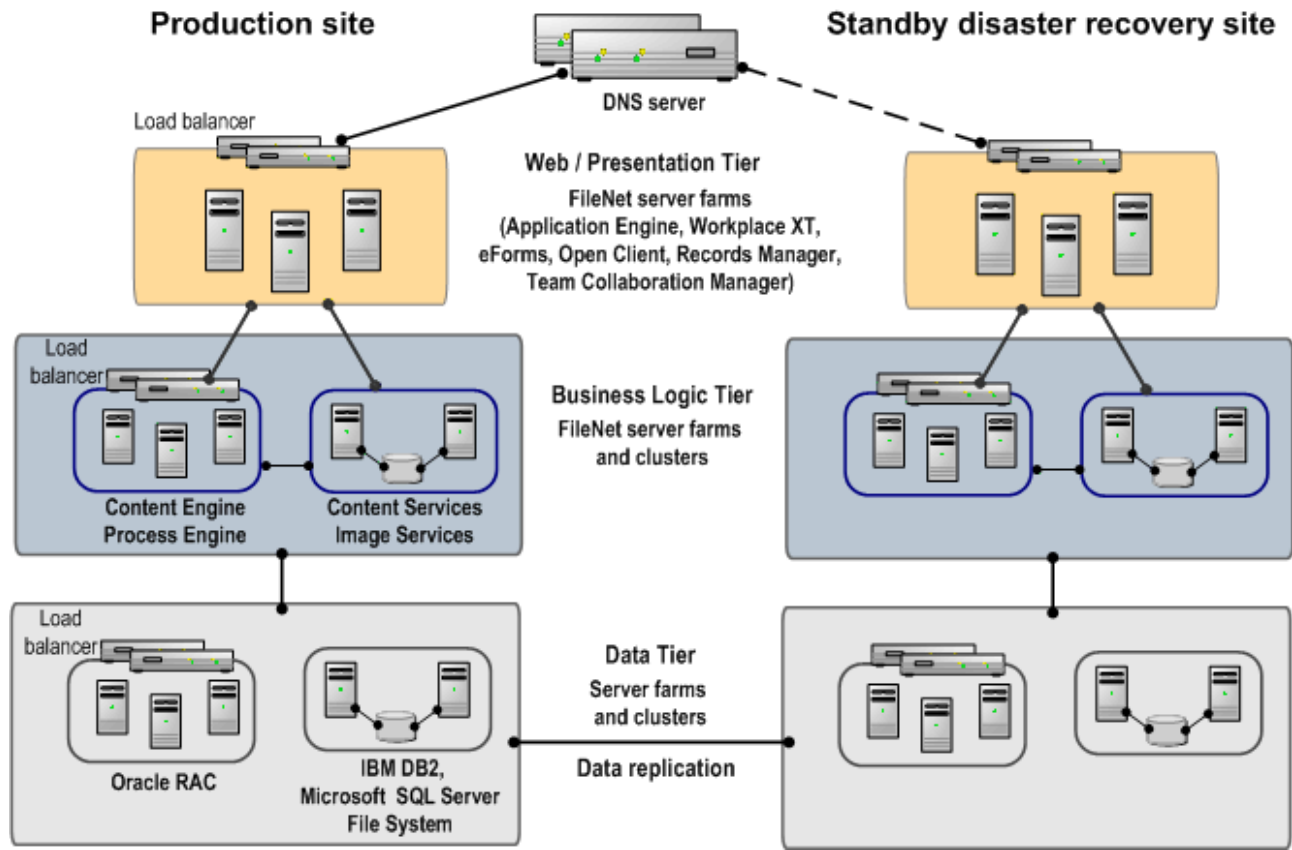
Data replication tools

Some of the tools from other vendors that can be used to create a geographically remote mirror of a running system (including database, file storage, and full index data) are:

- VERITAS Volume Replicator

- SAN/NAS storage vendors (for example, EMC Symmetrix Remote Data Facility, EMC Centera asynchronous replication, and Network Appliance SnapMirror)

- Geographic Remote Mirroring (GeoRM) for AIX®

- Oracle Data Guard (covers only the Oracle databases; other replication products are needed for the other data)

Section 508 compliant

FileNet P8 ensures Section 508 Compliance based on Electronic and Information Technology Accessibility Standards, published by the U.S. Access Board on December 21, 2000, at 36 CFR Part 1194. Compliance is achieved by incorporating Section 508 Subpart B – Technical Standards, as applicable, in all design, development, and testing activities. In particular, FileNet P8 addresses Section 508 product compliance in existing and new releases during all phases of product development, including design, design review, prototyping, construction, testing, quality assurance, and training of support staff.

IBM tests all user interfaces for Section 508 compliance, covering the following main areas:

- Can the user interface be navigated using only the keyboard?

- Can the user interface be read with a screen reader? (JAWS is most commonly used for this testing.)

- Can the font in the UI be enlarged for users with vision difficulties? (Zoomtext is most commonly used for this testing.)

- Browser accessibility features (formatting preferences and style sheets).

- Operating system accessibility features (for example, keyboard, sound, display, and mouse-related features).

In addition, IBM evaluates new user interfaces to ensure that color is not the only way to discriminate between items on the screen. Known limitations are documented. Installers, back-end office administration tools, and tools that require visual capabilities (for example, use of IBM FileNet Image Viewer) are exempt from full compliance   testing of these tools is limited. As such, product testing efforts focus on addressing components of Section 508, Subpart B, C and D.
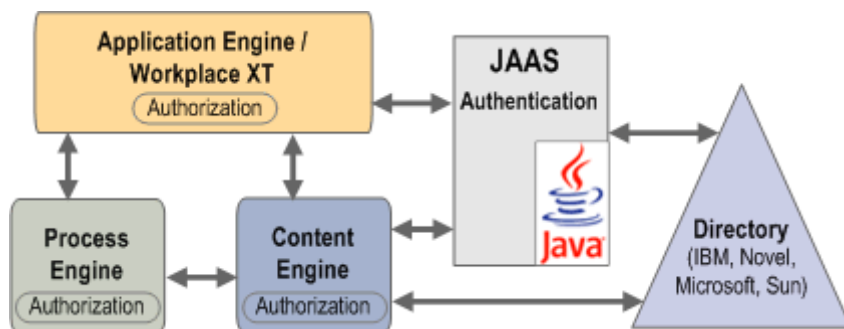
Secure

FileNet P8 provides an authentication model based on J2EE and the JAAS standard, a rich set of features for defining authorized access to IBM FileNet P8 objects, and a complete auditing system. This provides a secure environment for managing content and processes.

IBM FileNet P8 is integrated with your existing security databases, removing the need for you to create specialized security databases for your IBM FileNet P8 applications or to replicate information that is already in place in your organization.

Authentication

FileNet P8 leverages the Java Authentication and Authorization Service (JAAS) to provide authentication capabilities, as shown in the following diagram. The JAAS standard forms the framework for security interoperability in the J2EE world, enabling a wide range of integrations with security environments acquired from other vendors.

JAAS provides a policy-based, pluggable framework for reliably and securely determining who is invoking a Java application. The design of JAAS as a pluggable framework is a key architectural benefit of J2EE. It

allows applications to remain independent of the underlying authentication technologies, and it allows 3rd party application server vendors, authentication providers, and single sign-on providers to package solutions that can be leveraged by all J2EE applications and clients. Authentication is performed by the J2EE application server, rather than by FileNet P8. In a user environment, new or updated single sign-on solutions can be plugged in without requiring modifications to the client and server applications that are already deployed.

In addition to supporting pluggable authentication, the JAAS standard also allows for stacked authentication, wherein more than one step might be required to complete successfully in order for authentication to succeed. For example, a configuration might require a user name and password authentication, as well as a separate biometric authentication.

While JAAS is used to establish a caller's identity, an LDAP directory service is used to establish each user's group memberships. Enterprise identity management is provided by leveraging directory service products, currently Microsoft Active Directory, Sun Java System Directory Server, IBM Tivoli Directory Server, and Novell eDirectory. You can choose to create the various FileNet P8-defined users and groups in your directory service, or you can map these required roles to your own accounts, keeping FileNet P8's directory service footprint to a minimum. FileNet P8 supports groups that can include any number of users and other nested groups. Also, it honors any account states and restrictions (such as disabled and restricted logon hours) defined by the directory service.

Single sign-on

Implementation of the Content Engine server as a J2EE application allows FileNet P8 to take advantage of integrations between the J2EE Application Server vendors (such as IBM, BEA, and JBoss) and the leading single sign-on (SSO) solution providers such as IBM's Tivoli Access Manager and CA/Netegrity's SiteMinder. Refer to *Single Sign-On Integrations Via JAAS* in the online FileNet P8 documentation for further details on support for SSO environments.

Authorization

FileNet P8 provides fine-grained access control for all resources. For example, content is secured by assigning users and/or groups with access control levels for viewing content, updating content, updating metadata, updating access rights, and publishing. Default access rights can be defined for each class of object, and security policies can be used to make it easy to predefine access rights for many documents.

You can set security levels on workflow rosters, work queues, user queues, and component queues. The security levels you set affect the user's access to the work items contained in the roster or queue.

Security and records management

Support for FileNet P8's records management solution includes a security subsystem that was defined in accordance with the UK National Archives Standard for Record Repositories and DoD 5015.2. These security features include both hierarchical and non-hierarchical protective markings, secure cross-object store references, and control over property modification. While primarily designed for records management applications, in which content can be declared as a record, security markings are available for any FileNet P8 application.

Security and the Web

FileNet P8 supports the SSL (Secure Sockets Layer) protocol, which uses encryption to provide communication privacy over the Internet. In addition, IBM certifies that FileNet P8 components run correctly under a variety of firewall configurations.

Document audit trail

Auditing is one of several features the system administrator can use to monitor Content Engine activity and provide audit trail capabilities. For example, a Document class might be configured to create audit entries whenever documents of that class are checked in.

Applications can configure audit logging for common Content Engine activities, including those listed in the following section, and can enhance audit logging with application-specific custom audit definitions.

Standard Content Engine event audits are created by the user and logged automatically by the application when the appropriate event or operation is performed. Custom properties can be added to hold application-specific data.

Among the standard activities for which audit trails can be maintained are the following:

- Object creation

- Updates

- Deletions

- Denials of access

- Audit configuration changes

- Queries on auditable objects

Auditable Content Engine objects include Document, Folder, CustomObject, Link, ReferentialContainmentRelationship (RCR), and Annotation objects. ClassDefinition objects can also be audited for changes to the object store's metadata.

Audit events are stored in a table in the object store. They can be queried, and can be exported to XML format for use by reporting tools acquired from other vendors.

An audit entry in an audit log includes:

- The event or operation that triggered the audit entry (for example, creation of a new object)

- The user who triggered the entry

- Audit type (Audit Success or Audit Failure)

- Date and time of the entry

- The object associated with the event or operation (class, ID)

- Success/fail response from the event or operation that triggered the entry

Property sheets of auditable objects include an Audit History property sheet, as follows:
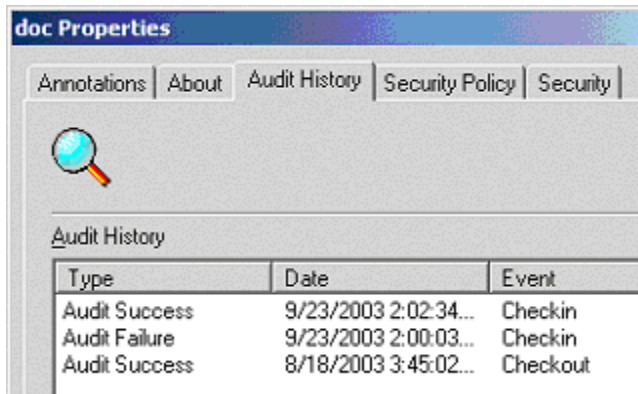
Markings

A powerful feature called security markings offers the ability to secure an object based on the value of one of its properties. Similar to the way in which object property values are assigned by using choice lists, markings offer the additional functionality of setting security on an object based on selected values. A marking set typically contains several values called "markings." Each marking has an associated list of users, each of whom has access rights to the object. Users not explicitly given access through the marking are denied access to the object. Markings do not override the normal security on the object, but work with it to provide an additional security layer. Markings are heavily used by the Records Manager application.

As an example, a system or security administrator uses Enterprise Manager to create a marking set called "Clearance" that contains a set of possible values for a property, defined as follows:

- **Top Secret** – Accessible only by the executive team

- **Confidential** – Accessible by the management team

- **Internal** – Accessible by all company employees

- **Public** – Accessible by all

Continuing this example, the administrator creates a property, arbitrarily called "Security Level," on a document class and associates the property with the "Clearance" marking set. When a new document based on this document class is created, a user with appropriate access rights can assign values from the "Clearance" marking set to the "Security Level" property for this document instance. Based on the value set for this property, security restrictions are placed on the document in addition to normal document security. For example, only users with Top Secret clearance can view the document. Note that Content Engine supports the capability to add custom properties to the base Document class. As a result, you might configure all documents in an object store to include a "Security Level" property that is associated with a marking set such as the one described previously.

Marking sets can function as an independent list of unrelated markings, or can be configured to function in a hierarchical fashion where the top-most marking in the list encompasses all of the markings and access rights below it.

Connectivity

FileNet P8 provides tools for integrating enterprise applications and information systems. In addition to providing a complete set of programmatic interfaces that can be used for performing a variety of integration tasks, the Process Orchestration and Component Integrator components specifically target typical integration requirements.

Internationalized

FileNet P8 architecture is Unicode (UTF-8) enabled to support global, multilingual system configurations. The user interface and message components are combined into language packs (in JAR files) for the primary language plus separate files for additional secondary (localized) languages. Multiple secondary languages and language packs can be installed on a single system. In a multilingual environment, the language settings are determined from the browser settings (user interface language) and the operating system (date/time format, currency settings, and so on).

FileNet P8 core components

The FileNet P8 family of products includes back-end services, development tools, and applications that address enterprise content and process management requirements. This section includes an architectural overview of these FileNet P8 product components.

As shown in the center of the following diagram, FileNet P8 provides the baseline components for enterprise content management solutions, including Content Engine, Process Engine, Application Engine, Workplace XT, and Rendition Engine. These components address enterprise content management (ECM) and business process management (BPM) requirements. Additional components can be added to a system to enable additional capabilities. Content federation capabilities for accessing enterprise-wide content are shown at the bottom of the diagram, and include Content Federation Services for Image Services, which enable applications built on the FileNet P8 platform to natively access Image Services documents. Records Manager, IBM FileNet P8 eForms, Email Manager, Capture, Process Analyzer, Process Simulator, and Image Services Resource Adapter can be purchased for additional licensing fees. The IBM FileNet P8 Portlets component is made available at no extra cost and includes BEA WebLogic Portlet and IBM WebSphere Portlet. Note that some components are available on a different release cycle than the FileNet P8 baseline components. Contact your service representative for information about the availability of and licensing for these additional components.
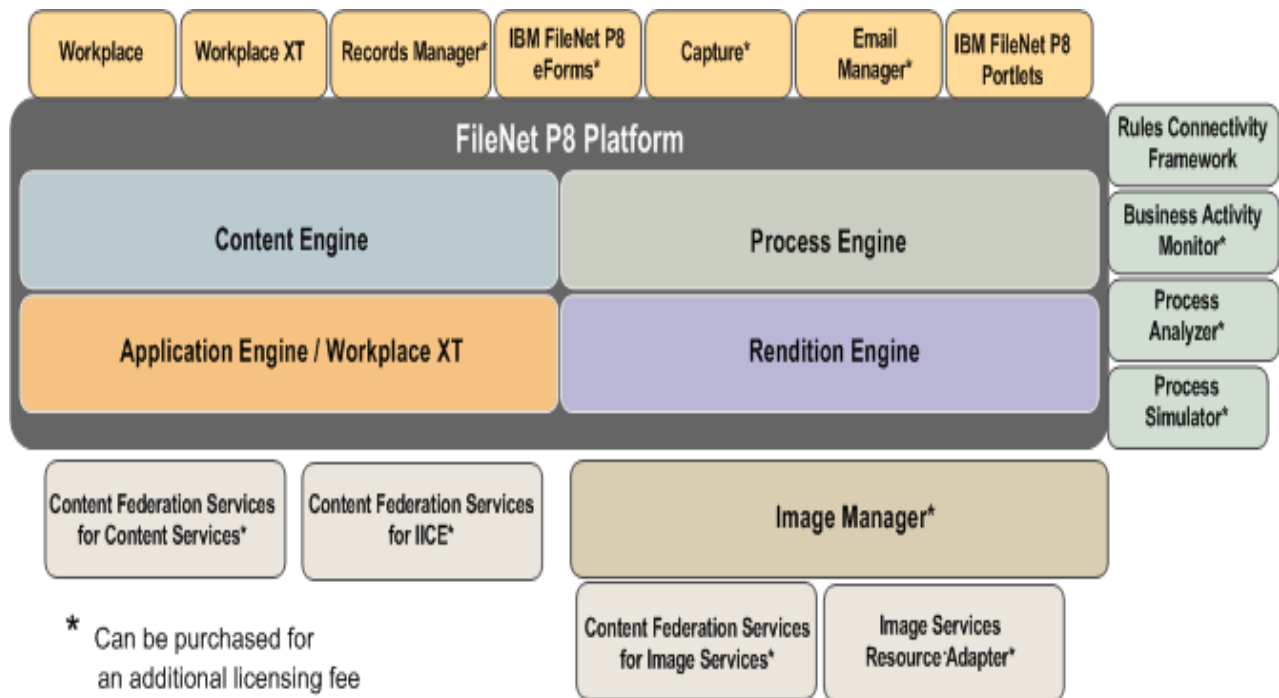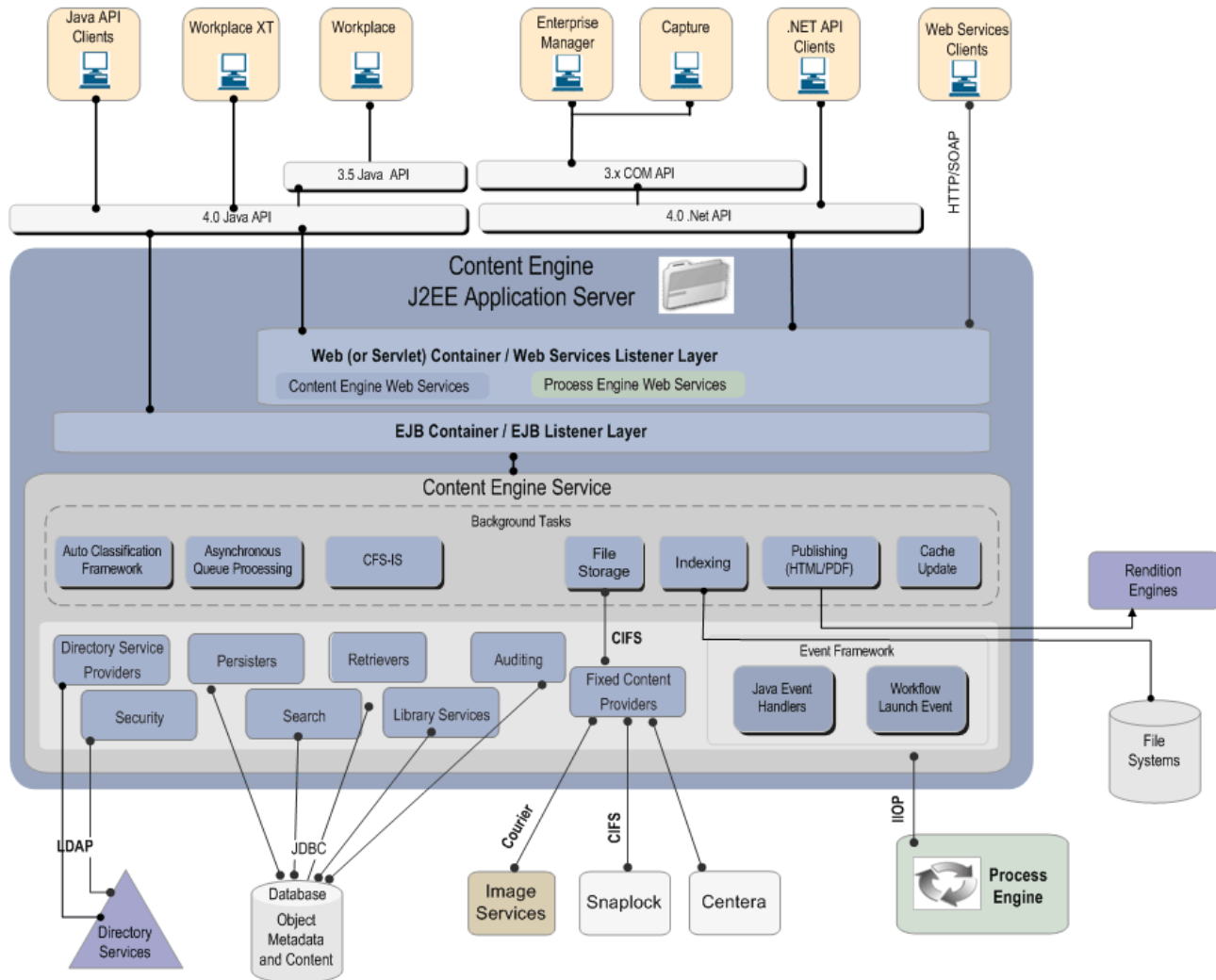
Content Engine

Content Engine is designed to handle the heavy demands of a large enterprise. It is capable of managing enterprise-wide workflow objects, custom objects, and documents by offering powerful and easy-to-use administration tools. Using these tools, an administrator can create and manage the classes, properties, storage, and metadata that form the foundation of an Enterprise Content Management system.

Key architectural aspects include:

- **Object-oriented, extensible metadata model –** Enables Content Engine to provide complex and flexible data representation; and a rich event framework provides the means to trigger an action in response to activities performed against Content Engine objects.

- **Application Programming Interfaces (APIs) –** Provide an extensible platform for development and cross-repository queries, and lets administrators configure systems programmatically. A Java API provides a rich set of Java classes which map to object store objects, such as Document, Folder, or Property Description; a Web Service API allows customers to author applications in a platform and language independent manner by exposing the object model in a small number of generic methods suitable for deployment in a Web environment; a .NET API (which is functionally equivalent to the Java API) provides for development of applications using the .NET platform.

- **J2EE-compliant application server –** Offers reliability, scalability, and high availability features, as well as support for a wide range of operating system platforms, application servers, and database technologies.

- **Scalable –** Can be deployed to suit the demands of the enterprise. As the enterprise's needs change, you can reconfigure the system by replacing, adding, or removing servers or applications without bringing the system down. You can add members to Web server farms and Content Engine server farms at any time.

- **Unicode-based –** Unicode is a universal character encoding standard that enables Content Engine to be targeted to multiple platforms, languages, and countries without re-engineering, and it allows data to be transported through many different systems without corruption.

The following diagram provides a detailed view of Content Engine architecture.

The services and components included in Content Engine are:

- **Content Engine Service** – Accepts incoming requests to create, update, and retrieve objects (such as documents, folders, and business objects). It handles database transactions required for managing one or more object stores. An object store is a repository for storing objects and the metadata that defines classes of objects. A Content Engine Service can access all object stores in the FileNet P8 domain. The Content Engine Service is responsible for adding new content and retrieving objects and content from an object store. In addition to servicing requests from ECM applications, the Content Engine Service hosts various background tasks that maintain all of the resources associated with each object store.

  - **Persisters** – System persisters handle ExecuteChange requests from the client. There is a hierarchy of persisters which mirrors the class hierarchy. These persisters are responsible for adding, updating, and deleting various objects to the object store.

o **Retrievers** – System retrievers handle requests from the client, such as GetObject, ExecuteSearch, and GetSearchMetaData. There is a hierarchy of retrievers which mirrors the class hierarchy and these retrievers are responsible for retrieving various objects from the object store and returning them to the client.

o **Security –** Provides a fine-grained security model for authorizing access to objects and content; interacts with directory service products, such as Microsoft Active Directory, Sun Java System Directory Server, and Novell eDirectory, to authenticate users and retrieve user and group account information.

o **Search –** Provides property-based search capabilities. Content Engine Search capability enables users to simultaneously search multiple object stores.

o **Auto-classification framework** – Automatically assigns incoming documents of specified content types to a designated document class. Support is provided for the auto-classification of XML documents, and a classification framework is available for creating custom classifiers for content types other than XML.

o **Event framework** – Enables configuration of actions in response to specific activities that occur on objects stored on a FileNet P8 object store. For example, when a new document is created, you can configure an event to launch a workflow that is used to further process the document.

o **Event auditing** – Enables monitoring of the activity on the FileNet P8 system. Actions that occur on objects cause an automatic logging of an audit entry for later analysis.

o **Library services** – Provides document versioning, automatic content classification, and document lifecycle and metadata management capabilities.

o **Content-related services** – Provides the capability to store and retrieve content securely from different types of storage devices, and provides all of the functionality necessary to manage this content. A document or annotation consists of both properties and content. Properties are stored in the relational database that holds the object store. By default, the content (for instance, a Word or PDF document) is stored in the object store database as well (referred to as "database storage"). You can choose instead to have content stored on a network file server (a "file storage area") or on a custom storage device (a "fixed storage area").

  ▪ **File storage** – The Content Engine Service can manage one or more file stores, allowing document content to be stored on a network file server.

  ▪ **Fixed content providers** – Manages one or more Image Services and other storage devices for managing content in those repositories.

  ▪ **Full-text indexing** – The Content Engine Service manages interactions with the IBM FileNet P8 Content Search Engine. You can enable full-text indexing for one or more document classes and annotation classes, allowing full text searches to be performed against their contents, as well as their properties.

- **Content cache** – If content caching is enabled, the Content Engine Service stores copies of recently created or accessed content in a local network file share so that subsequent retrievals are more efficient. Content caches are primarily used in geographically distributed environments, where access to remote content storage resources might be over a slow network link.

- **Document publishing** – The Content Engine Service manages requests to render documents into Adobe Acrobat PDF format, or into HTML format. When a request is made to transform content to an alternate format, the Content Engine Service submits format translation requests to a Rendition Engine.

- **Content Engine Transport** – The Content Engine Service provides two transport mechanisms that applications can use to access the Content Engine server.

  o **EJB Transport** – The EJB transport is an Enterprise Java Bean (EJB) that runs in the J2EE EJB container of the application server that hosts the Content Engine Service. Access through this EJB is referred to as the Content Engine "EJB Transport". The EJB transport can only be accessed through the Content Engine Java API. All access to the Content Engine Service ultimately flows through the EJB transport (the Content Engine Web service is itself a client of the EJB transport). You can configure clients of the 4.0 Content Engine Java API and the 3.5 Content Java API to use the EJB transport when accessing the server.

  o **Web Services** –The Web services transport can be accessed directly by Web service-based applications. You can configure clients of the 4.0 Content Engine Java API and the 3.5 Content Engine Java API to use the Content Engine Web service. Clients of the Content Engine .NET API, and COM Compatibility API must use the Content Engine Web service.

  The Web services enable access to content and process capabilities over the Web. The Content Engine Service hosts two Web services: Content Engine Web Services (CEWS) and Process Engine Web Services (PEWS). Hosting these two Web services together allows them to share common resources and a common environment. Both are implemented, deployed, and managed using the Systinet Server for Java toolkit, and both leverage the same set of J2EE JAAS login modules.

  The CEWS and PEWS APIs provide developers with a complete set of interfaces that supports custom application development in a Web environment. These two interfaces share many similarities, including:

  - Both are compliant with WS-I (Web Services-Interoperability) Basic Profile.

  - Both implement the same standard WS-Security authentication profiles. The Username Profile allows clients to provide their user name and password credentials to authenticate with the Web server by using the customer enterprise directory service (such as Microsoft Active Directory or Novell eDirectory). The Kerberos Profile allows .NET based Web service applications that are running in a

Windows-based Integrated Login environment with Active Directory, to authenticate using their Windows operating system credentials.

- Both allow you to develop applications and FileNet P8 extensions using Visual Studio .NET and Rational® Application Developer.

- Both implement a service-oriented architecture, and are designed for stateless operation. That is, each request to the Web server is independent of any other, with no client-specific state held by the server once the request has been completed. This architecture facilitates scalability and high availability by allowing client requests to be load-balanced for multiple hosts.

CEWS is a low-level API that exposes the full Content Engine feature set. All authoring and administrative functionality is available in addition to the basic document management features. Advanced batching capability is available, allowing arbitrarily complex sets of data to be retrieved in a single round trip to the server, and arbitrarily complex sets of updates to be performed in a single round trip. These capabilities combine to allow powerful and highly efficient applications to be built.

PEWS is a high-level API based upon the Process Engine Java API. It presents an interface that should look very familiar to Process Engine Java API developers. PEWS is focused on Process Engine runtime functionality, allowing for rapid development of step processors and applications that launch or monitor workflows. It does not provide process authoring or administration capabilities. PEWS is designed to work in tandem with Process Orchestration capabilities.

- **.NET API** – The .NET API provides a .NET Common Language Runtime (CLR)-compliant API for accessing the full capability of Content Engine. The API accepts user name and password credentials for authentication. The .NET API works only with the WSI Listener.

- **Java API –** The Java API is an interface-based API that provides access to the full capabilities of Content Engine. The Java API can be configured to run across either the Enterprise Java Beans (EJB) or Web Services Interoperability (WSI) protocols. When running across the EJB listener, the Java API leverages the full capability of the J2EE framework to propagate security and transaction context. The Java API also leverages a JAAS context, if available.

- **Enterprise Manager** – A Microsoft Management Console (MMC) snap-in application through which you can manage Content Engine services and object stores.

Process Engine

Process Engine provides software services for managing all aspects of business processes (also called workflows), such as process execution, process routing, rules management, process simulation and modeling, and workflow analysis. Process Engine components allow you to create, modify, and manage workflows implemented by applications, enterprise users, or external users (such as partners and customers).

The following diagram provides a detailed view of Process Engine architecture:
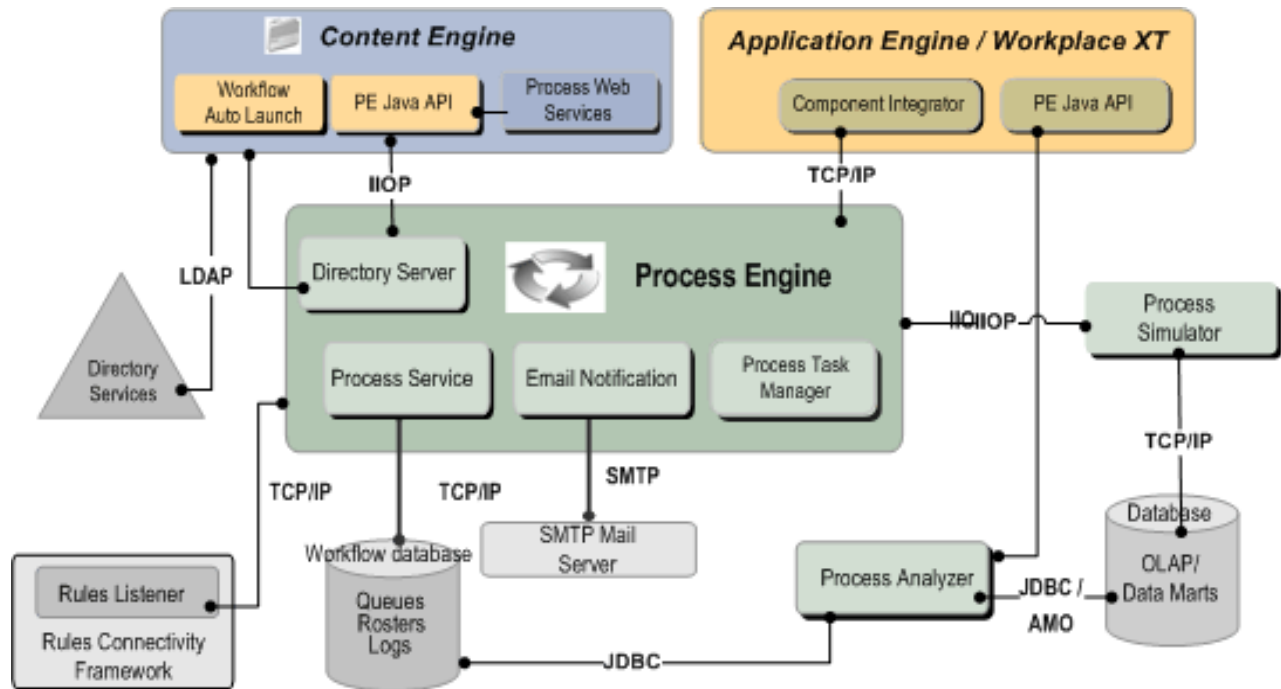
The preceding diagram includes the following services and management components:

- **Process Service –** The core of the business process management system. It provides workflow services on behalf of Process Engine.

- **E-mail Notification –** Enables automatic transmission of e-mail to users when specified process-related events occur; e-mail notification can also be used to track workflows.

- **Rules Connectivity Framework –** Provides a framework for rules integration. A process designer or business analyst creates business rules using rules software acquired from another vendor and associates the rules with the steps of a workflow. When a running workflow encounters a rule, Process Engine sends a request to the Rules Listener to execute a rule set. Rules Listener executes the rules in the rule set and returns the results to Process Engine. The rules engine provider must implement a specific API in order to integrate with Process Engine.

- **Process Task Manager –** Provides administrative tools for configuring and managing process-related services on the Process Engine server.

- **Process Analyzer –** Provides analysis capabilities to determine cycle times, find bottlenecks, and generate reports and charts to analyze the processes deployed in the Process Engine system.

- **Process Simulator –** Allows you to simulate processes or perform "what-if" scenarios with hypothetical or historical data.

- **Workflow database –** Contains Process Engine configuration information as well as running instances of workflows.

- **Component Integrator –** Provides an extensible integration framework that allows you to easily create connectors. Java and JMS adaptors are ready for you to use without modification.

- **Process Web Services –** Provides a Web services API to Process Engine.

- **Workflow auto-launch –** Event-driven component that launches workflows in the Process Engine system. You can configure this component to launch a specific workflow in response to specific events. For example, filing a document in a folder might launch a review process in the Process Engine system.

- **Directory Services –** Directory service functionality is accessed through the Content Engine server. Content Engine supports several LDAP providers and single-sign-on (SSO) functionality. This pass-through approach simplifies directory services configuration by centralizing it for both Content Engine and Process Engine.

Application Engine

Application Engine is the FileNet P8 component that hosts the Workplace Web application, Workplace java applets, and application development tools. It is the presentation layer for both process and content. Application Engine protects user credentials passed between Workplace and Content Engine and, if configured, provides SSL security.
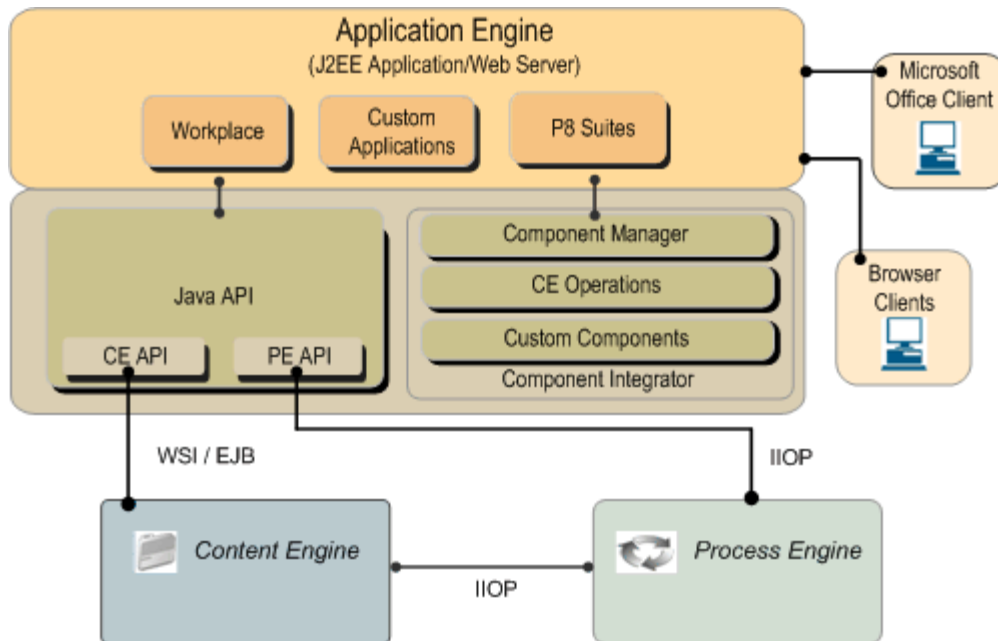
The services and applications included in Application Engine are:

- **Workplace –** An end-user Web application that provides access to the document management capabilities and is tightly integrated with the business process management capabilities of FileNet P8.

Workplace also supports extended FileNet P8 capabilities such as forms management, records management, and portals.

Workplace is built using the Web Application Toolkit and runs within a Web container on a J2EE application server. Workplace leverages the latest JSF, Web 2.0, and AJAX technology for rendering many of its user interface elements, such as query results and folder contents. The majority of functionality provided in Workplace is zero-download HTML that runs under many browsers on any platform. Advanced capabilities that are graphic-intensive are provided as Java applets, including Process Designer, Process Configuration Console, Process Administrator, Scenario Designer, Scenario Console, Search Designer, and Publish Designer.

- **Integration for Office –** Enables users to easily manage Office documents and Outlook e-mail messages within a FileNet P8 object store. Users can store, search, and retrieve documents, e-mail, and attachments directly from Office menus. In addition to securing and versioning Office documents, users can browse object stores and insert properties into Word and Excel documents. Users can also use entry templates to add documents to an object store and launch approval workflows. For more information, see Microsoft Office integration.

- **WebDAV Servlet** – Allows users to create and edit documents and manage Content Engine files from WebDAV-compliant applications such as Word or DreamWeaver.

- **Component Integrator –** Makes it possible to interact with an external entity—a "component", such as a Java object or (Workplace only) JMS messaging system—from a workflow step. Component Integrator handles the import of Java classes and manages the communication between Process Engine and the interfaces. Component Integrator provides the Web Services invocation framework for Process Orchestration, which uses SOAP/HTTP as its transport protocol.

Component Integrator includes adaptors, which are interfaces that communicate events from Process Engine to external entities such as Java objects. Adaptors interact with different types of components from a workflow step. (FileNet P8 provides Java and Java Message Service (JMS) adaptors for calling Java components and posting messages to message queues. Our examples show only a Java adaptor.)

As shown in the following diagram, the first step in the flow of control is the configuration and deployment of components. An administrator registers the component using Process Configuration Console, which in turn creates a component queue. Next, the administrator deploys the necessary JAR files for the component on the Application Engine server and registers the component in Process Task Manager. Process Designer retrieves configuration information from Process Engine and now a user can create workflow definitions. The user creates a step and selects the registered component and method to invoke, specifying the workflow fields that will be passed as parameters to the method at runtime. The workflow definitions (requests for work) are then transferred to Process Engine queues.
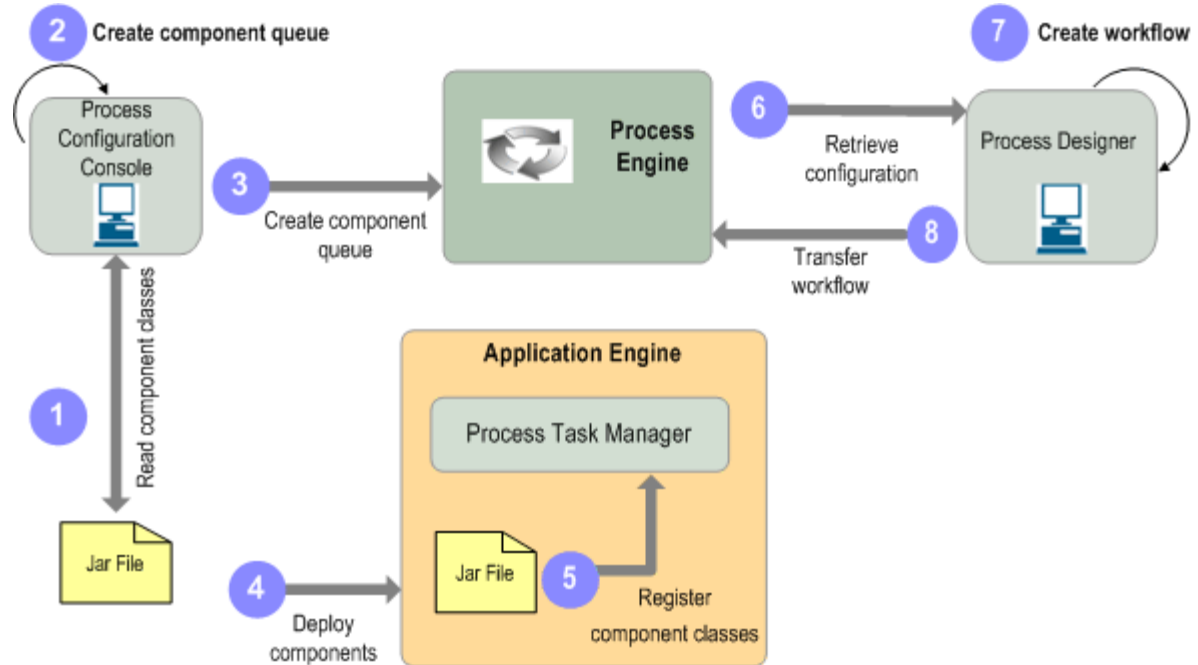
The following diagram depicts the runtime interaction of the Component Integrator with Application Engine services (such as Component Manager), Process Engine queues, and a custom entity. When the workflow process is executed, the Component Manager retrieves the request from the component queues and invokes the components through the adapters. For each step in the workflow, the following general sequence of events takes place:

- Information is sent to the component through the adapter.

- The component performs its work and interacts with the custom entity.

- The result of the work is saved in the step.

- The step is completed.
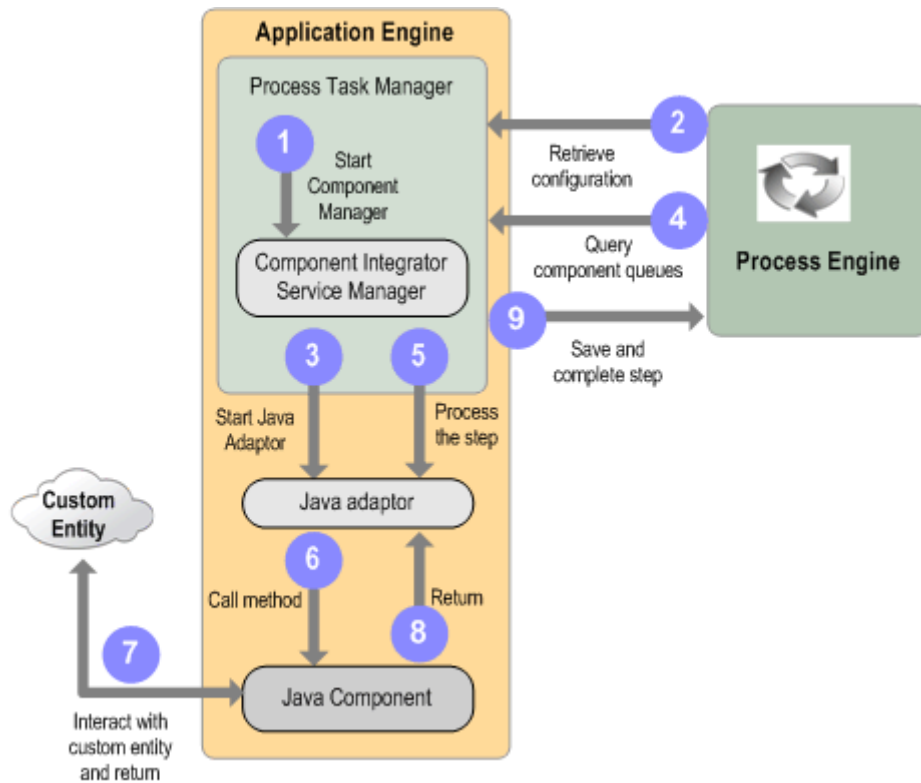
- **Content and Process Java APIs** – An extensive set of Java classes for programming custom applications and for extending the provided applications. These classes provide programmatic interfaces for interaction with the Content and Process Engine software. Developers can use these APIs to build a variety of applications, including those that rely on a J2EE Web container (Java Server Pages and Java Servlets), Enterprise Java Beans (EJB) container, or J2SE stand-alone Java applications, as shown in the following diagram.
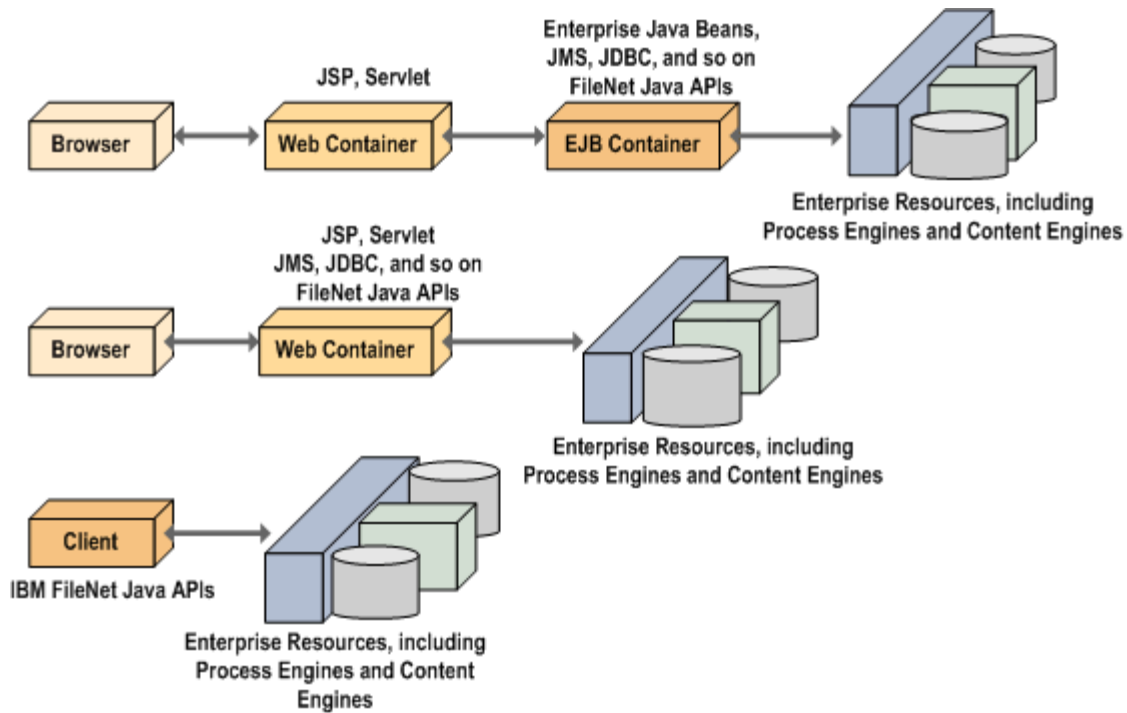
- **Content Engine .NET API** – An extensive set of C# classes for programming custom applications and for extending the provided applications. These classes provide programmatic interfaces for interaction with the Content Engine software. Developers can use these APIs to build .NET-related applications.

- **Web Application Toolkit –** Provides an extensible framework and reusable modules for building Web applications. The Toolkit provides application developers with access to Content Engine, Process Engine, and servers acquired from other vendors. It supplies the behaviors and data structures for authentication, event routing, state information, preferences, localization, and other features of robust and scalable applications. In addition, the Toolkit's reusable UI component model facilitates the development of a robust HTML-based application user interface with little or no DHTML/JavaScript required. IBM has developed a number of Web applications from the Toolkit, including Workplace and Records Manager.

As shown in the following diagram, additional functionality provided by Workplace includes user interface components—including JSP pages that specify the page layout, JavaBeans™ that render user interfaces, XSL documents that are used to control how XML returned from the Java API is rendered in the user interface, and Cascading Style Sheets (CSS) that define fonts, colors, and other formatting options.
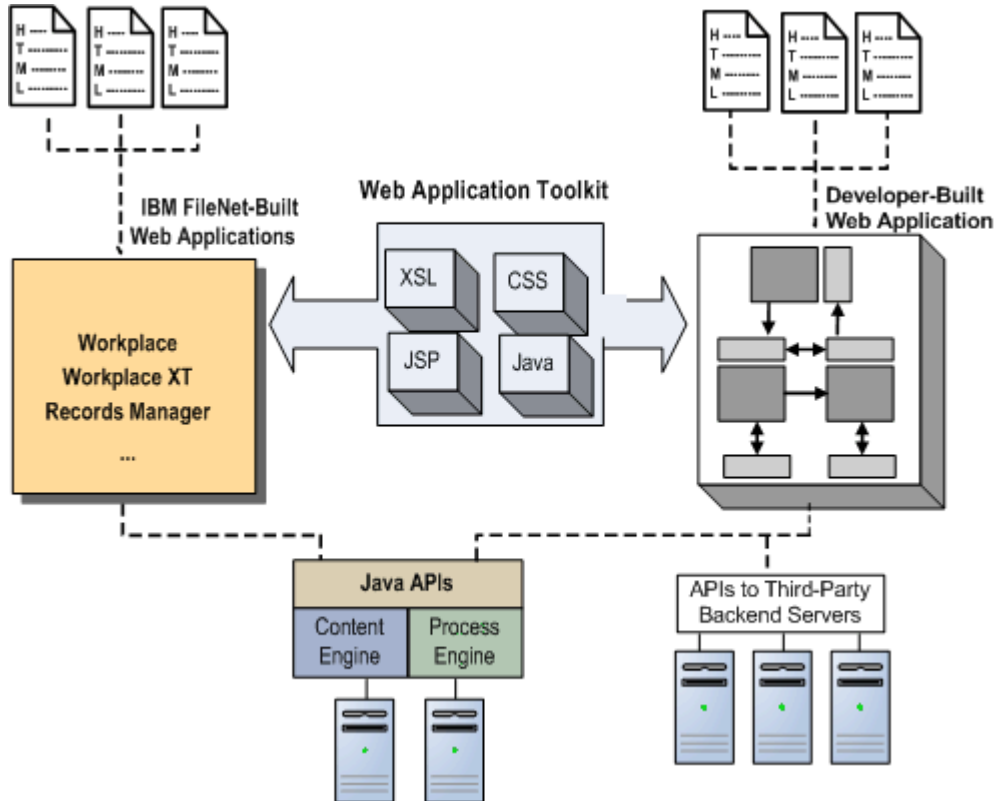
- **Application Integration Toolkit**

  The Application Integration Toolkit is a full-featured API that enables third parties to integrate their Windows-based client applications with Workplace. FileNet P8 uses this toolkit to provide integration with Excel, PowerPoint, Word, and Outlook. In addition, FileNet eForms Designer uses this toolkit. Customers and partners can use the toolkit—including the reusable user interface—to integrate with other applications with very little coding.

  The toolkit contains a set of COM servers that are installed on the Windows-based client and Java servlets that run on the J2EE application server. The Application Engine UI Service, which supports calls from both thick client applications and Web-based thin clients, provides the toolkit's reusable Workplace JSP pages and wizards. Together, these components provide developers with a complete set of content management functionality, including the interactive user interface.

- **Application Integration ExpressAddIn**

  The Application Integration ExpressAddIn is a developer tool contained in the Application Integration Toolkit that enables rapid integration of the IBM FileNet Application Integration infrastructure into applications acquired from other vendors, such as Office.

  The tool is a COM server that defines incoming and outgoing interfaces that enable two-way communication between client applications and the Application Integration framework. While similar to the IBM FileNet Application Integration Toolkit in some respects, the ExpressAddIn leverages significantly more infrastructure code, and thereby greatly reduces the burden on integration developers. The ExpressAddIn is delivered with the Add-in sample application, which demonstrates how to use the ExpressAddIn to integrate an application with the Application Integration infrastructure. Developers can customize the sample code to rapidly integrate their application into the framework and easily customize an integration to meet specific needs.

- **Application Engine UI Service**

  Application Engine UI Service is an Application Engine service used by applications to access eForms, to capture content through Entry Templates, to check in content, to gain access to work tasks and step processors, and to select objects through browsing and/or searching. Application Engine UI Service supports calls from both thick-client applications and Web-based thin clients. A Web-based application must be built from the Web Application Toolkit. A thick client must be built from the Application Integration Toolkit.

  The following diagram illustrates how an external thick-client or Web-based application interacts with the Application Engine UI Service. In this example, an external application uses the Application Engine UI Service to call Workplace JSP pages (thereby leveraging Workplace user interface components).
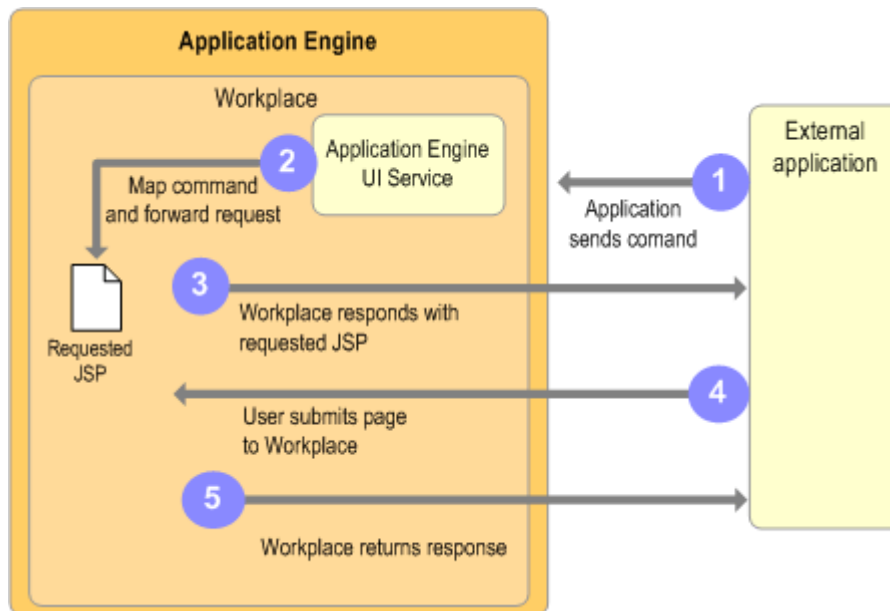
1. The external application sends a command to Application Engine UI Service. Thick clients use an XML-based protocol to communicate with Application Engine UI Service, whereas thin clients use request-based command response URLs.

2. Application Engine UI Service handles authentication, maps the command to a JSP request, and forwards the request to Workplace.

3. Workplace responds with the requested JSP page.

4. The user submits the page to Workplace.

5. Workplace returns a response to the external application.

The following diagram illustrates how an external Web application interacts with the Application Engine UI Service. Note that the Web application must be built from the Web Application Toolkit. The initial page launched from the browser-based client shown in the diagram would be from the external Web application.
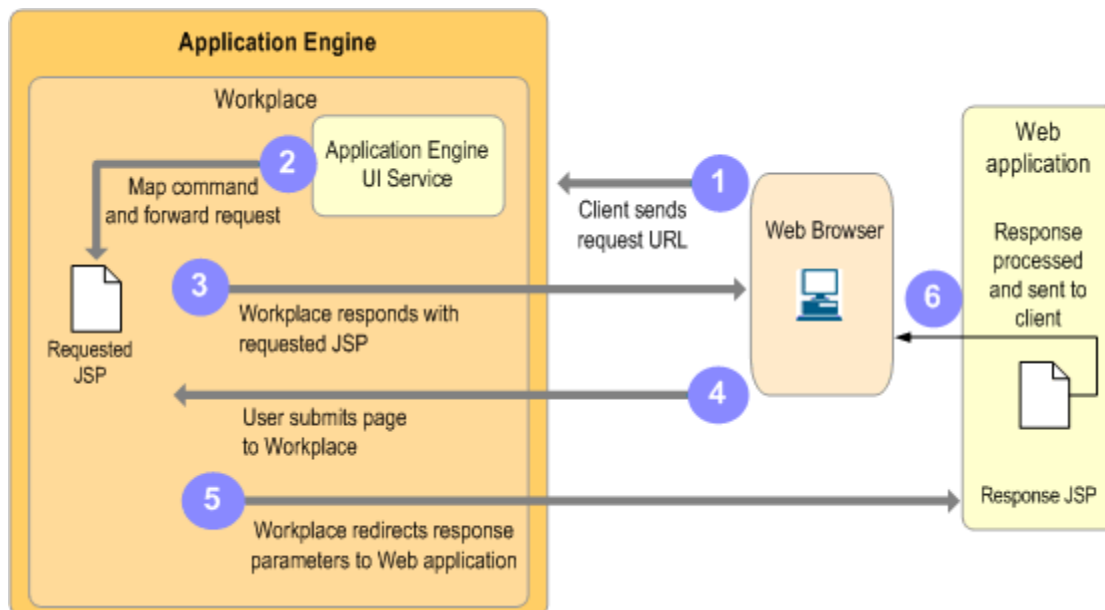
1. A browser-based client sends a request URL to the Application Engine UI Service. The request URL is an ID-based command, and includes a response URL parameter that specifies a page in the Toolkit-based Web application that will handle the response information returned by Workplace.

2. Application Engine UI Service handles authentication, maps the ID-based command to a JSP request, and forwards the request to Workplace.

3. Workplace responds with the requested JSP page.

4. The user submits the page to Workplace.

5. Workplace redirects response parameters to the Toolkit-based Web application specified in the request URL.

6. The response is processed and passed to the client.

Workplace XT

Workplace XT is an optional FileNet P8 component (similar to Application Engine) that hosts the Workplace XT Web application, providing access to the process and content functionality of FileNet P8. You can install Workplace XT in addition to or in place of Application Engine. Workplace XT protects user credentials passed between Workplace XT and Content Engine and, if configured, provides SSL security.
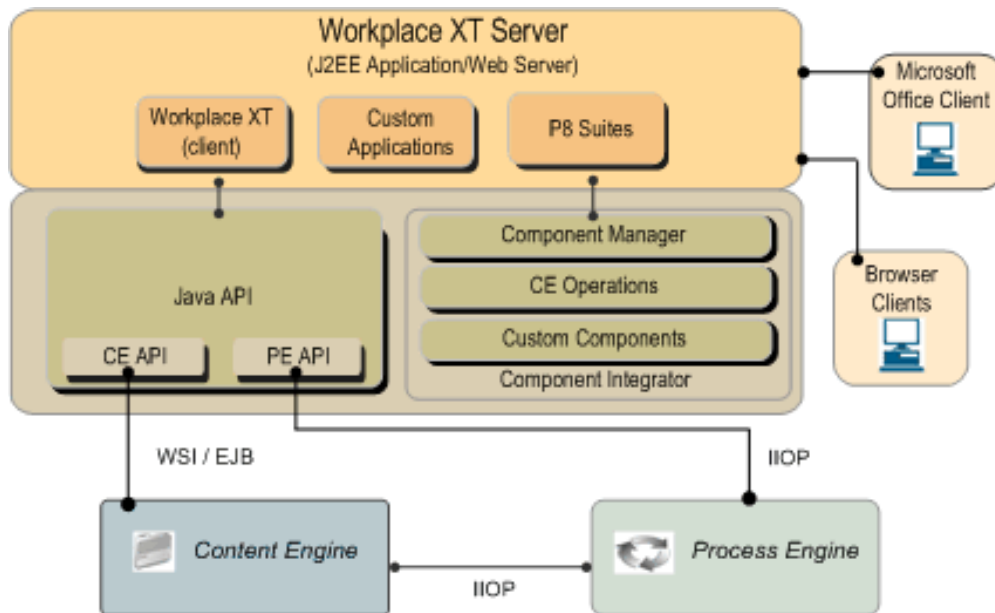
The services and applications included in Workplace XT are:

- **Workplace XT (client) –** An end-user Web application that provides access to the document management capabilities of, and is tightly integrated with the business process management capabilities of, FileNet P8. Workplace XT also supports extended FileNet P8 capabilities such as forms management, records management, and portals.

  Workplace XT runs within a Web container on a J2EE application server. Workplace XT leverages XSL for rendering many of its user interface elements, such as query results and folder contents. The majority of functionality provided in Workplace XT is zero-download HTML that runs under many browsers on any platform. Advanced capabilities that are graphic-intensive are provided as Java applets, including Process Designer, Process Configuration Console, Process Administrator, Scenario Designer, Scenario Console, and Search Designer.

- **Integration for Microsoft Office –** Makes it possible for users to easily manage Office documents and Outlook e-mail messages within a FileNet P8 object store. Users can store, search, and retrieve documents, e-mail, and attachments directly from Microsoft Office menus. In addition to securing and versioning Office documents, users can browse object stores and insert properties into Word and Excel

documents. Users can also use entry templates to add documents to an object store and launch approval workflows. For more information, see Microsoft Office integration.

- **Component Integrator –** Makes it possible to interact with an external entity—a "component," such as a Java object or JMS messaging system—to accomplish a workflow step. Component Integrator handles the import of Java classes and manages the communication between Process Engine and the interfaces. Component Integrator provides the Web Services invocation framework for Process Orchestration, which uses SOAP/HTTP as its transport protocol.

  Component Integrator includes adaptors—interfaces that communicate events from Process Engine to external entities such as Java objects. Adaptors interact with different types of components from a workflow step. (FileNet P8 provides Java and Java Message Service (JMS) adaptors for calling Java components and posting messages to message queues. Our examples show only a Java adaptor.)

Rendition Engine

Rendition Engine converts documents from a variety of formats, such as those produced by Office programs (Word, Excel, PowerPoint) to PDF and HTML formats. Multiple Rendition Engines can be configured to support large numbers of conversion requests.

The following diagram provides a detailed view of Rendition Engine architecture and its interaction with Content Engine.
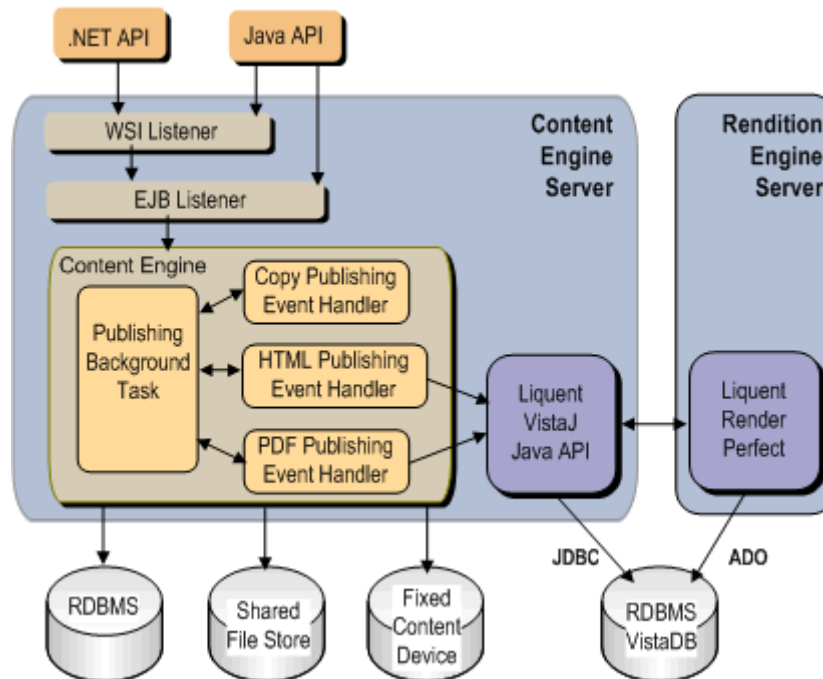
An application requests that a document be published by submitting a publish request through one of the FileNet P8 APIs. A publish request might simply create a copy of the document in a new folder, with new security attributes, or it might also transform the document content to another format (HTML or PDF). The publish request is queued in the Content Engine server. The publishing background task dequeues the

publish request and calls the appropriate publishing event handler. If a format conversion is a part of the publish request, then the handler sends the document to the Rendition Engine server to perform the necessary transformation.

Content components

Image Services Resource Adaptor

The Image Services Resource Adaptor (ISRA) is a set of Java classes for interacting with IBM FileNet Image Services. These classes conform to the J2EE Connector Architecture (JCA). Using this programmatic interface, you can connect to an Image Services repository. You can then query for and retrieve documents from the Image Services repository, as well as create documents to be stored in the repository.

Note that Content Federation Services for Image Services, described in the following section, offers deeper integration of Image Services and FileNet P8 object stores than ISRA.

Content Federation Services for Image Services

Content Engine document catalogs can directly reference content stored in Image Services content stores. Content can be stored in either Content Engine- or Image Services-based storage infrastructure.

The following diagram illustrates how Content Engine can use Image Services as a high-volume capture system. Large volumes of documents are captured (by using scanning or other high-volume document capture application on the Image Services system) and indexed and stored on Image Services object stores. The two options for indexing are: a) index to the Content Engine catalog only, or b) index to the Content Engine catalog and the Image Services catalog. In the latter option, updates to Image Services catalog entries are automatically propagated to the Content Engine catalog. In addition, the Image Services Catalog Export tool is available to migrate existing Image Services catalog entries to the Content Engine catalog. Since the location of document content is transparent to all applications, a user can view a document stored in an Image Services repository through an Image Manager viewer application, such as IDM Desktop, and from a FileNet P8 application, such as Workplace, without knowledge of where the document is stored.
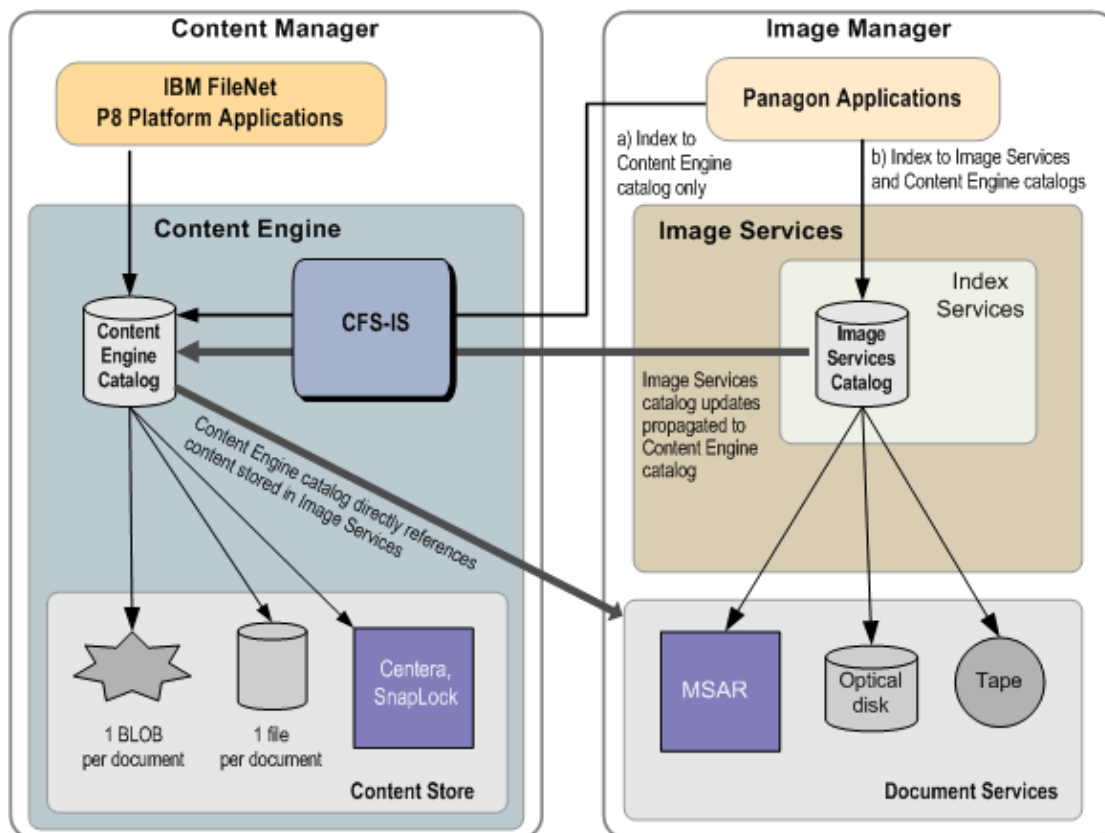
As shown in the following diagram, Content Engine can use Image Services as a fixed content storage device. FileNet P8 applications can retrieve content stored on Image Services-based systems through Content Engine.
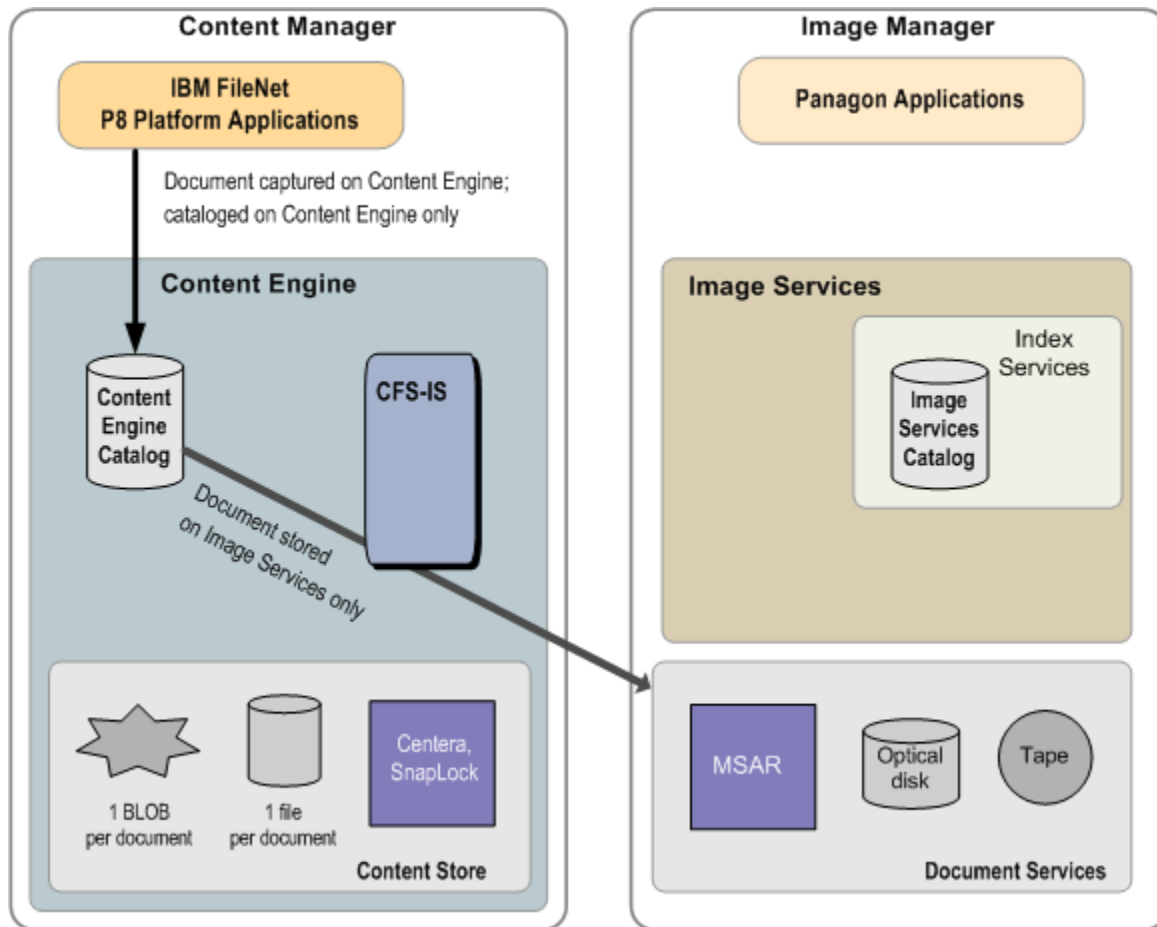
In this scenario, documents are captured in the FileNet P8 object store and cataloged only on the Content Engine catalog. Content Engine makes calls to CFS-IS to store content to and retrieve content from Image Services content stores. Documents captured and indexed in this manner are accessible from FileNet P8 applications, but not from Image Manager applications.

You can view more Information about CFS-IS planning and installation in *IBM FileNet P8 Content Federation Services for Image Services Guidelines.* (To download the guidelines from the IBM support page, see Accessing IBM FileNet P8 documentation.)

Content Federation Services for Content Services and IICE

The following diagram illustrates the CFS architecture and process flow.
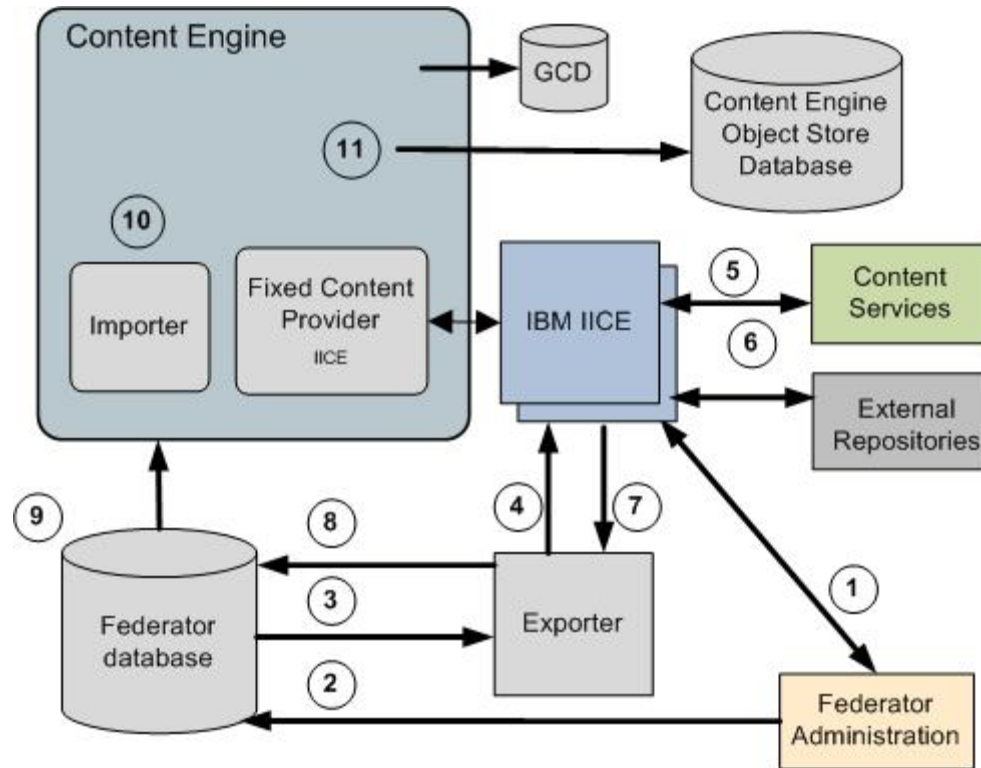
1. Federator Administration is configured to access the external repositories through Information Integrator Content Edition (IICE).

2. Federator Administration stores rules and queries for retrieving external documents in the federator database.

3. The federator database reads the rules and queries and passes a request to the Exporter.

4. The Exporter sends a request to IICE to retrieve the document references and metadata.

5. IICE sends the queries for the document references and metadata to the external repository.

6. The external repository sends the document IDs and metadata to IICE.

7. For items that match the queries, IICE sends the document references and metadata to the Exporter.

8. The Exporter updates the federator database with the document references and metadata.

9. The federator database sends the document references and metadata to the Importer.

10. The Importer creates a list of items to federate.

11. The Importer adds the document reference and the related metadata for each external document to the FileNet P8 object store.

CFS-CS and CFS-IICE components

The primary components are the Exporter, the Importer, and the federator database. The Exporter uses administrator-defined rules and queries from Federator Administration to build a list of documents in the external repository to be federated. The Importer federates the listed documents into a FileNet P8 object store. The federator database contains a set of tables used exclusively by the Exporter, a set of tables used exclusively by the Importer, and a set of queue tables that define the interface between the Exporter and Importer. The federator database can reside on any accessible server.

Exporter

The Exporter performs the following tasks:

- Executes rules against an external repository. Documents that match the criteria are queued for federation. Rules are executed in order of priority, as indicated by the rule's location within the list of rules.

- Populates a table in the Federator Database with documents or version series update requests that match the criteria specified by the rule. The table contains the metadata (in XML format) of the documents to be federated. Each document has a unique identifier, which is the same as the URN stored in Content Engine.

- Maps properties from the external repository to FileNet P8 properties, using mapping rules established in the Data Map Designer application.

- Configures the batch size for a federation run according to supplied values. The batch size establishes the number of items that the exporter can process in one run, and the query block size determines the size of each query chunk.

CFS-CS and CFS-IICE support the use of multiple Exporters to enhance performance.

Importer

The Federator Importer performs the following tasks:

- Retrieves batch and content information from the federator database.

- Keeps track of what has been imported and determines if a request is a first-time import or a re-import request (property or version updates to a previously exported document).

- Creates federated content in FileNet P8. Adds the URN link (similar to a URL) to the document or document version in the external repository.

- Maintains a mapping of the external version series data to the FileNet P8 version series in the federator database.

Federator database

The federator database stores the following information:

- The metadata of the documents selected for federation from the external repository.

- Federation status and timestamp information for batches and for individual items.

- CFS-CS and CFS-IICE configuration items, including logging levels, and IICE authentication credentials.

- CFS-CS and CFS-IICE rule information, including the constructed queries (that form part of a rule), the target Content Engine folders (required in each rule), and, optionally, records declaration options for records-managed documents.

Capture

IBM FileNet Capture provides the capability to add documents to the FileNet P8 Content Engine or Image Manager repository. Capture supports a variety of document capture methods including scanning, faxing, and importing files. Capture supports automating the document entry process to provide production-level capabilities.

Capture supports a wide variety of scanners from low volume personal scanners to high volume production scanners. In addition, the FaxEntry portion of the Capture product provides an automated method for automatically capturing inbound faxes. The Capture File Import capability provides a method to import existing documents from user directories.

The following diagram illustrates a simplified architectural view of Capture:
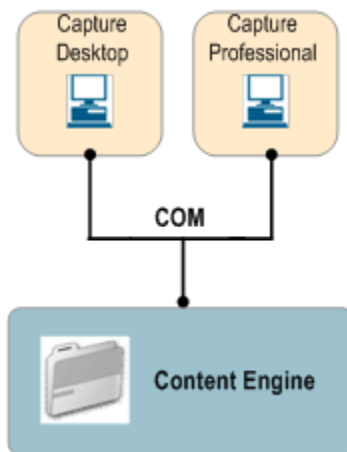
As shown in the following diagram, Capture uses a SQL Server database as an intermediate repository for batches of scanned images and indices which are then committed to a FileNet P8 object store. The separate database for batches allows the majority of capture processing to be performed with little impact to Content Engine performance. The diagram also shows how Capture workstations can be distributed to handle higher image throughput requirements.
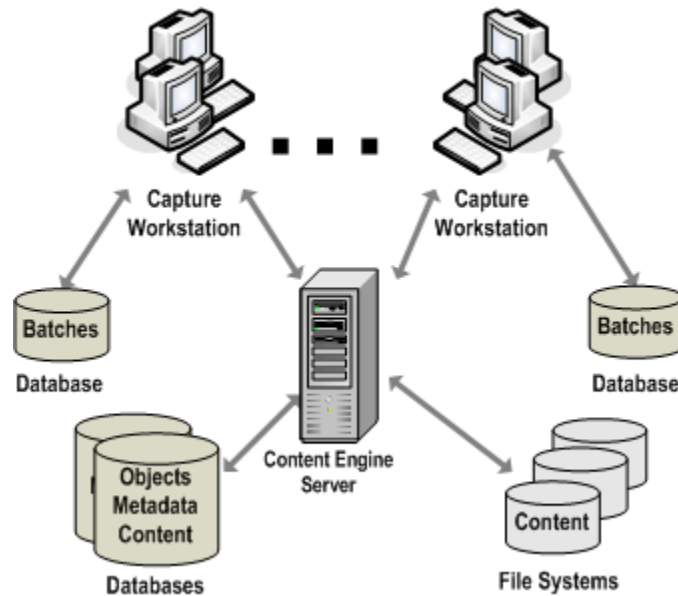
The following diagram illustrates the flow of documents during the Capture process.
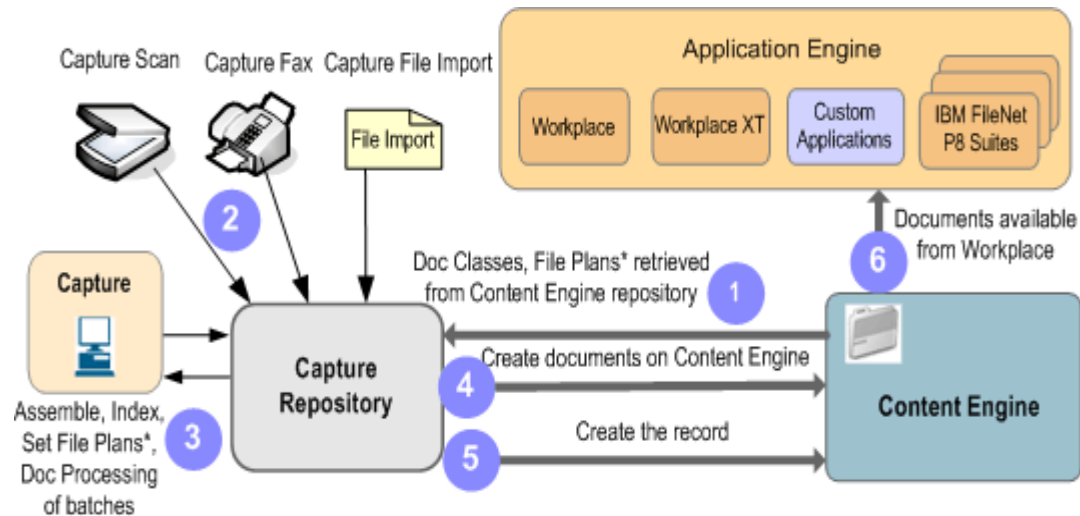
1. Document classes and File Plan* information are retrieved from Content Engine into the Capture repository.

2. Images or documents enter the Capture system (through scanning, inbound fax, or import) and are grouped into batches for further processing.

3. A variety of Capture functions can be performed on these batches, including assembling, indexing, document processing (such as de-skew and de-speckle), selecting file plans, and assembling the images into documents. These Capture functions are performed on the batches in the Capture repository and do not affect the performance of the FileNet P8 object store.

4. After all necessary Capture operations have been performed, the batches are moved to the FileNet P8 object store using the Content Engine COM API. During this process, all of the documents in a batch are moved to the FileNet P8 object store and the appropriate document class properties are set.

5. If a Records Manager File Plan has been specified, a record is also created in the File Plan Object Store (FPOS) and associated with the created document.

6. After the documents are moved to the FileNet P8 object store, they can be accessed (viewed, searched for, etc.) from Workplace and other FileNet P8 client applications.

Capture Toolkit

The Capture Toolkit enables customers and partners to build additional functionality for the image capture process. This additional capability can be plugged into Capture, or can be a separate, stand-alone application.

Process components

Process Analyzer

Process Analyzer supports monitoring and analyzing business processes by collecting workflow events from Process Engine and generating statistical data and reports. Process Analyzer leverages OLAP (On-Line Analytical Processing) technology for fast analysis of multi-dimensional information, drill-down from summary view to details, and enables you to interactively explore business process data from different perspectives.

The following diagram shows the architecture of Process Analyzer and how data flows from Process Engine to Process Analyzer.
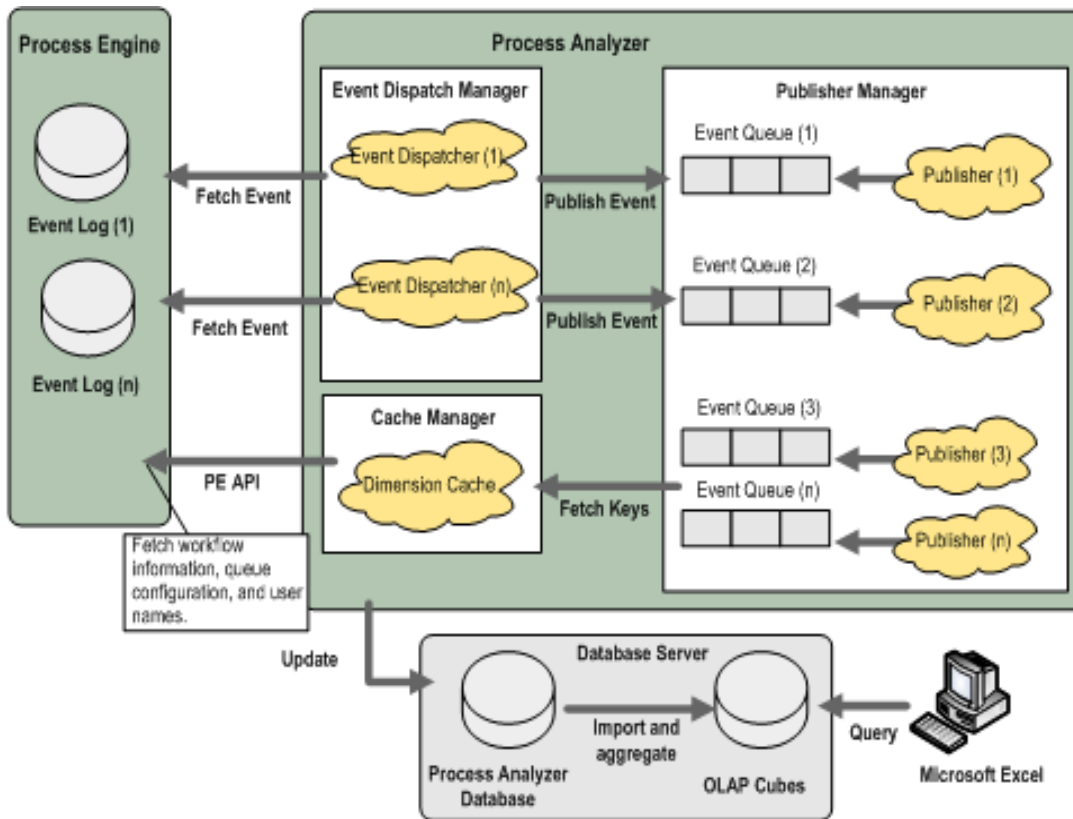
Process Analyzer has an Event Dispatch Manager. Its job is to manage the Event Dispatcher threads. The Event Dispatch Manager creates an Event Dispatcher thread for each event log in the Process Engine database. The Event Dispatchers are responsible for querying and retrieving the events from the event logs in the Process Engine database. The Event Dispatchers push the events into event queues. There is a Publisher Manager and it is responsible for managing the Publisher threads. The number of Publisher threads is configurable by the user. The job of the Publisher threads is to retrieve events from the event queue and process them. Part of that process is to interact with the Cache Manager to retrieve the necessary configuration data (that is, workflow definition, queue, and user) to process the events. At the end of the publishing interval, the Publisher Manager initiates the Publisher threads to update the fact tables in the database. At that time the fact tables are updated with the raw statistical data created while processing the events. The statistical data in the fact tables eventually is used to build the OLAP cubes. Ultimately, the OLAP cubes provide the information required to generate Excel (or other spreadsheet) charts for the user.

Business Activity Monitor

The following diagram provides an overview of Business Activity Monitor (BAM) architecture.
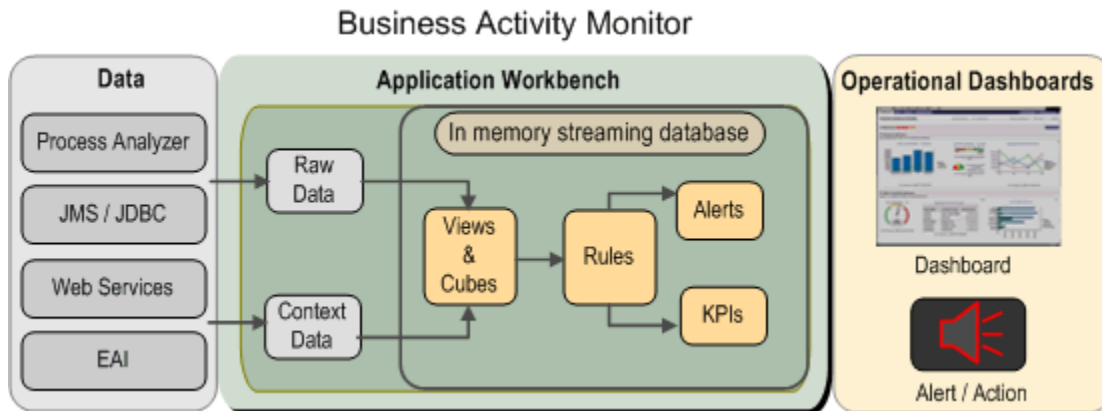
BAM collects data from Process Analyzer through its built-in connection, and can connect to and collect data from external enterprise applications through standard connections such as JMS, JDBC, Web Services, and EAI. BAM has an in-memory streaming database. You can also configure the software to generate in-memory database views and OLAP cubes from the correlated analytical data. You can configure dashboards to display information stored in the views and cubes, and create thresholds (that is, Key Performance Indicators) and configure BAM to alert and/or perform actions if a threshold is violated.

Process Simulator

Process Simulator performs "what-if" scenarios, providing business analysts with important information they can use to streamline business processes. The following diagram illustrates how a simulation is run and provides an overview of Process Simulator architecture.
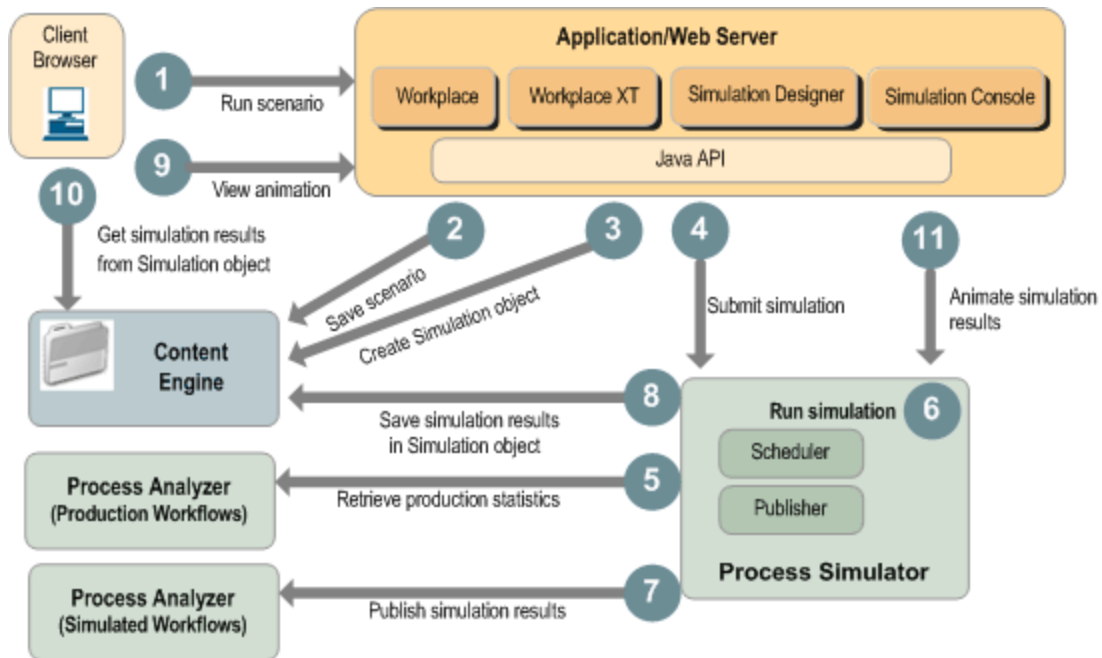
1. An analyst using the Simulation Designer in Workplace creates a scenario and requests that it be run.

2. The scenario is saved to a FileNet P8 object store. The saved scenario can be modified and saved as different versions of the original, and each version can be re-run as needed.

3. A Simulation object is saved in the FileNet P8 object store. The Simulation object is updated with the simulation results once the Simulation Engine has completed the simulation.

4. The scenario is submitted to Process Simulator as a simulation. (A simulation is a runtime version of the scenario.)

5. Process Simulator retrieves statistics about the workflows in the simulation from Process Analyzer.

6. Process Simulator runs the simulation.

7. Process Simulator publishes analytical results of the simulation to Process Analyzer.

8. Process Simulator saves the results of the simulation in the Simulation object in the FileNet P8 object store.

9. The analyst selects a simulation object from Workplace and makes a request to view an animation of the simulated workflow.

10. The simulation results are retrieved from the Simulation object in the FileNet P8 object store.

11. The Animator animates the simulation based on the retrieved simulation results.

Business Process Framework

The following diagram shows how the Business Process Framework (BPF) integrates with FileNet P8.
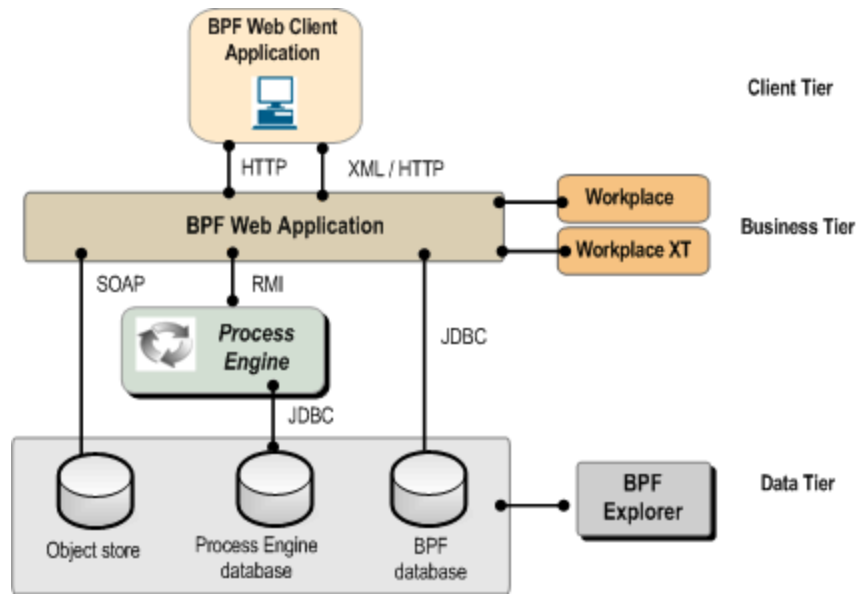
The **Data Tier** refers to the persistence layer, core FileNet P8 platform services, and the services provided by other vendors that serve as a repository for the BPF domain object model and configuration. The Data Tier is scalable.

- BPF uses a FileNet P8 object store for storing and managing the business object data. The BPF Business Tier uses the Content Engine Java API for all content related operations.

- BPF uses a Process Engine database for storing and managing the business object process state. The BPF Business Tier uses JDBC to retrieve workflow queue contents and the Process Engine Java API for all process-related operations.

- BPF Web Application settings, domain object model, and data visualization definitions are stored in a dedicated BPF Metastore database. The BPF Metastore database can be collocated with the Process Engine or Content Engine database server, or can be installed on a separate database server. The BPF Business Tier uses JDBC to communicate with the BPF Metastore.

The **Business Tier** provides a structural model for the application data and a set of procedures for performing business related activities. The BPF Business Tier:

- Is shared by multiple clients.

- Provides a framework for case management transactions, ensuring that the information in the case model remains in a valid state.

- Provides a persistence mechanism for application data.

The Business Tier is subdivided into the following parts.

BPF Business Services – A set of modules that sit above the Data Tier and encapsulate the case management business logic. BPF Business Services are not hosted as an independent service. BPF Business Services expose a set of high-level functions that allow clients to manipulate the BPF business objects:

- o Create, retrieve, update, and delete

- o Manipulate the process state

- o Perform event logging

BPF Web Application – Provides the Inbasket and work presentation user interface. The BPF Web Application is hosted within a J2EE application server environment and uses BPF Business Services. The Web browser-based client communicates with the BPF Web Application directly or through a load balancer.

The BPF Web Application contains a feature-rich browser-based user interface, with nearly thick client functionality. The architecture is optimized for scalability and contains a myriad of user productivity features. In order to implement this feature-rich user interface and to leverage as much FileNet P8 functionality as possible, BPF attempts whenever possible to re-use and extend upon the user interface functionality provided by the Web Application Toolkit and Workplace as opposed to designing and creating a custom Web Application from scratch.

Workplace is a J2EE application built on top of the Web Application Toolkit (WAT). WAT is a reusable components framework for building custom applications. It provides API level interfaces that are supported by IBM FileNet.

The overall design uses the Content Engine and Process Engine API directly where applicable, the WAT API where possible, re-uses the Workplace Integration Servlet-provided user interface elements where available, and develops any additional custom code necessary to meet the functional requirements.

The BPF Web Application is built around an AJAX (Asynchronous JavaScript and XML) development model, also utilizing the Web Application Toolkit (WAT). The BPF Web Application exposes Case Management services by using an Action Dispatcher servlet. The Web client uses XMLHTTP (AJAX) for communication with the Action Dispatcher. The design of the BPF Web Application minimizes the client-server traffic by updating the browser only with the data that has changed. The BPF Web Application also contains a set of client-side JavaScript functions that employ data validation techniques, minimizing unnecessary round trips to the Web server.

The BPF Web Application user interface is metadata driven and is built dynamically upon user request. The metadata describing the user interface layout is stored in the BPF Metastore database and is managed by using the BPF Explorer configuration tool.

When a user makes a request to retrieve or manipulate a business object:

- The business object presentation manifest and any associated business rules are retrieved from cache or the BPF Metastore (if the data is not present in cache).

- The business object data is retrieved from the Process Engine and Content Engine servers.

- Both data and presentation manifest are packaged as an XML message.

- The XML message is sent to the browser.

- The browser will render the HTML user interface by using the client side XSL transformation and make the business object available to the user.

The BPF Web Application services are exposed as an action dispatcher, which is a passive listener accepting and processing requests from the Web browser clients. The action dispatcher accepts the services requests and provides a response back as XML messages over HTTP based protocol.

The BPF Web Application was designed using standard J2EE technologies and should work in any J2EE compliant application server environment.

The BPF Business Tier can be scaled up using multiple CPU servers or multiple servers using J2EE clustering. When J2EE clustering is used, clients of this tier see a single server, with a virtual IP address. This IP address is mapped to multiple physical servers by using a load balancing (either hardware or software) solution. The overall system can scale to handle an increased load, by adding additional servers to the cluster.

The BPF Business Tier is not entirely stateless, requiring a sticky connection to be maintained to a specific server. The load balancing solution should support the sticky connection feature.

High availability in the BPF Web Application Tier is achieved by creating a cluster with multiple servers. If a single server goes down, the load balancer will detect this condition and stop routing requests to this server. Access to the cluster is routed through the remaining servers automatically.

Although the BPF Web Application is not entirely stateless, it provides a means to address high availability requirements by storing a snapshot of the session state on the client. When the Web client determines that the Web session is lost, which is typically due to Web session expiration, the Web client recreates the session and allows the user to continue working after logging back on to the BPF Web Application.

The **Client Tier** provides two clients of the Business Tier:

- **Web Client –** The BPF Web client provides a nearly thick client feature-rich interface to the BPF Web Application and is comprised of the HTML, JSP, JavaScript pages, and other objects. The BPF Web client uses HTTP and XMLHTTP to access the BPF Web Application services. The Web

client contains a set of client-side JavaScript functions that employ data validation techniques, minimizing unnecessary round trips to the Web server.

- **BPF Operations –** BPF operations are a set of functions that are packaged as a Component Integrator-based Java class. BPF Operations create and manipulate case objects by utilizing BPF Business Services from a step on the workflow map.

Application components

IBM FileNet P8 Portlets

IBM FileNet P8 Portlets provide a portal integration for BEA WebLogic Portal and IBM WebSphere Portal. In addition, customers and partners can create their own portlets using the FileNet P8 Java APIs. In addition, IBM distributes the source code for the provided portlets so that customers and partners can modify and extend the capabilities if desired.

Architecturally, the IBM FileNet P8 Portlets leverage the Web Application Toolkit, as shown in the following diagram, which illustrates the IBM FileNet P8 Portlet integration with BEA WebLogic Portal.
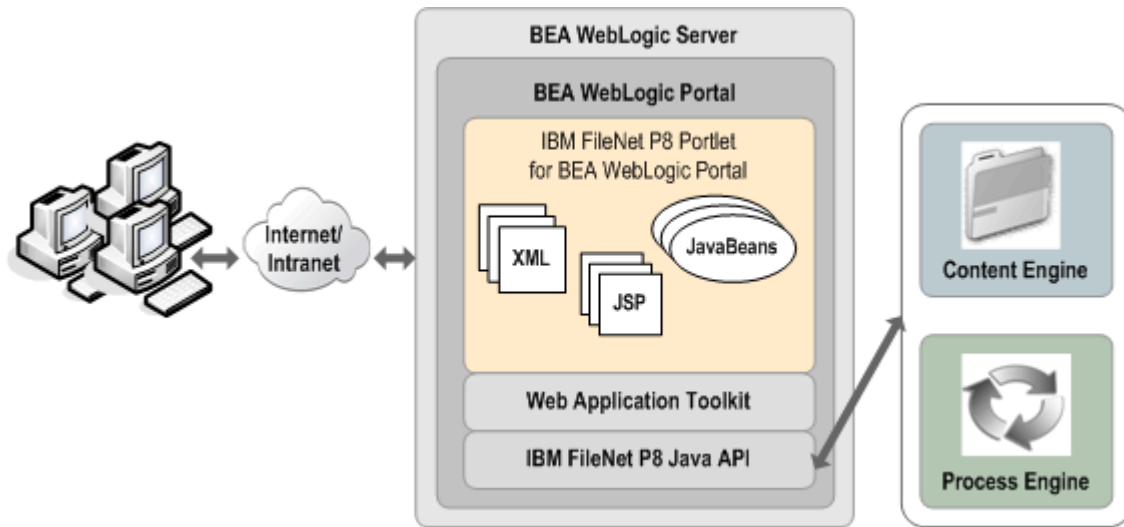
eForms

IBM FileNet P8 eForms is installed on the Application Engine or Workplace XT server, and form solutions are configured using the Workplace or Workplace XT applications' administrative wizards. Users interact with eForms documents that reside in an object store directly from within their application or by using links on a custom Web application that point to them using a URL to the application server.

The following diagram illustrates a Windows-based client using the eForms Designer application to design and deploy form templates, a client browser acting as a solution designer configuring a solution, and a client browser acting as a form user. Most communications are with Application Engine or Workplace XT,

which in turn communicate with Content Engine, Process Engine (in the case of business process workflow-related forms), customer databases, and print services (for forms rendering capability).
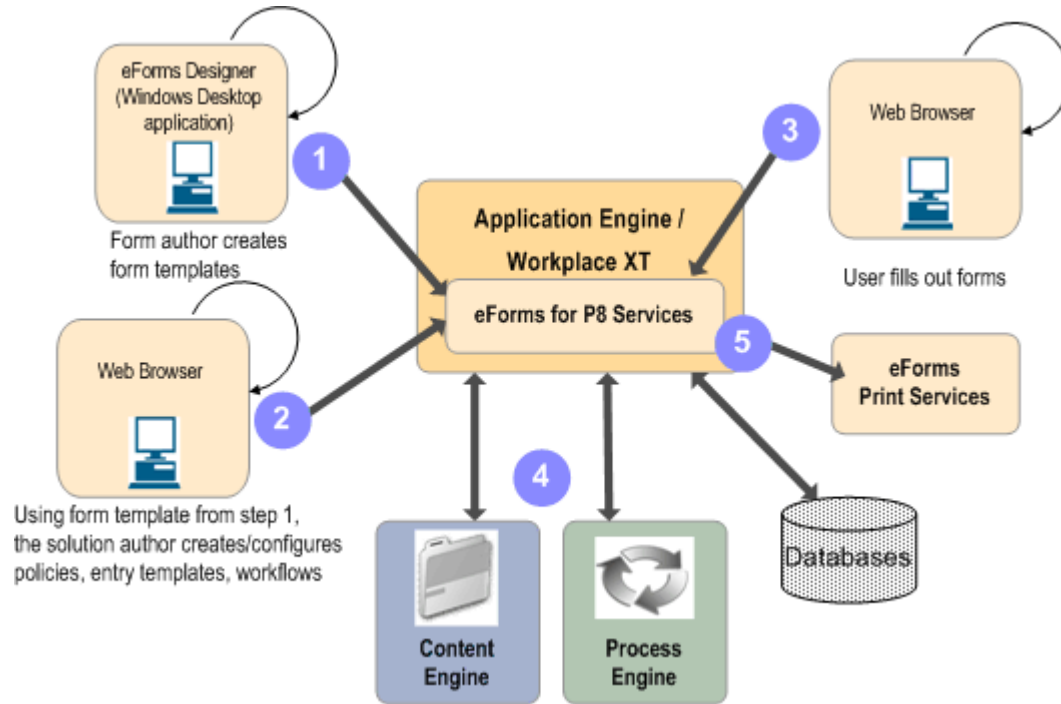
A form template author uses eForms Designer, a powerful forms design application that runs on the client desktop, to design and store form templates. The tool has a direct connection through Application Integration to a FileNet P8 object store to allow easy storage and retrieval of form templates. The form template author typically associates a form template with a form policy. (If a form template is not associated with a form policy, a user can open a form based on that template, but cannot enter data into the form.)

A solution author (who can be the same person as the forms template author) creates workflows, entry templates, and form policy documents that together define how a form will behave when used by a user. Using the Form Policy Authoring Wizards in Workplace, the solution author performs such tasks as associating a workflow with a form, configuring form fields to map to workflow data or to document properties, and so on. The resulting form policy document is then saved to the FileNet P8 object store and is subsequently used when a user wants to create a new form.

A user can click on a link to a form policy to begin filling in a new form, or a new or partially completed form can be displayed as the interface to a workflow step (when the user clicks on a work item from a Process Engine work queue, for instance). eForms retrieves the information from the form policy document, the form template, and any workflow data and renders them into an intelligent HTML-based form that appears in the user's Web browser. No desktop applications or browser downloads are required.

While completing a form, a user can enter values that trigger data lookups to other corporate databases, resulting in form fields being filled in automatically. Users can also be given the option to generate a high-quality PDF version of their completed form that can be printed or saved. This feature uses the eForms Print Services, a component that, in addition to generating PDFs, can be called programmatically to generate TIFF versions that can be used as part of an archival process. eForms Print Services runs in a

Microsoft .NET environment and is typically installed on the Content Engine server, although it can be installed on a separate Windows-based server running IIS and .NET.

Form data documents, produced as a result of completing and saving a form, are also stored and versioned in a FileNet P8 object store using the settings specified in the Form Data Entry Template associated with the form's policy.

Records Manager

Records Manager includes the following components:

- The Records Manager application, a Web-based application that runs on the same server as Workplace or Workplace XT.

- Data models that, when imported into FileNet P8 object stores, define the object classes that represent the file plan. The application includes a Base data model as well as data models that comply with U.S. Department of Defense standard DoD 5015.2 Chapter 2 and Chapter 4 and Public Records Office standard PRO 2.

- Workflows that implement common records management disposition activities, such as record review, interim transfer, transfer, export, and destroy.

- Process operations that enable declaration of attachments from a business process.

- Disposition sweep Java program that calculates record retention and can be distributed to a number of servers to support large repositories.

- Hold sweep Java program that places holds on entities that meet the criteria of existing conditional holds.

- The File Plan Import Export Tool that allows you to import an external file plan or export a file plan to an external environment (for example, importing a file plan from a test environment to a production environment).

- Crystal report templates for generating reports.

- Records Manager Java API that you can use in conjunction with the Content Java and Process Java APIs to create, browse, search, and manage records; place records on hold; browse and manage file plans; initiate disposition; and calculate retention.

Email Manager

Email Manager is a server-based e-mail management solution that is offered as both a FileNet P8 suite and as a purchasable add-on to other suites. Email Manager is designed to automatically identify incoming and outgoing e-mail messages that contain content of business value and import them into a FileNet P8 system, from where they can be managed in the same way as other business documents. Email Manager integrates with the most popular corporate e-mail systems: Microsoft Exchange, Lotus® Domino®, and Novell GroupWise mail servers and their respective e-mail clients.

Email Manager operates as a Windows-based service. It is recommended that Email Manager reside on its own server, separate from the rest of the FileNet P8 system. The product consists of the following four major components.

- **Configuration Manager** – The desktop application used by Email Manager Administrators to set up and configure the system, define profiles and rules for e-mail capture, and define post-processing options. A .NET Web service must be installed on the server on which the Configuration Manager resides.

- **Configuration Database** – The database in which the configuration information defined in the Configuration Manager is stored. The Configuration Database does not have to reside on the same server as the Email Manager application. It can reside on any drive to which Email Manager has access privileges.

- **Email Connector** – Monitors the journaling folder on one or more Exchange, Domino, or GroupWise servers on the network and enables Email Manager to evaluate each incoming or outgoing e-mail against the profiles defined by the Configuration Manager.

- **File Importer** – Interfaces with object stores using the COM API to place captured e-mail messages into the object store and with the Java API to pull content from the object store. The File Importer supports FileNet ZeroClick architecture for Records Manager.

The following diagram illustrates the seamless integration with FileNet P8:
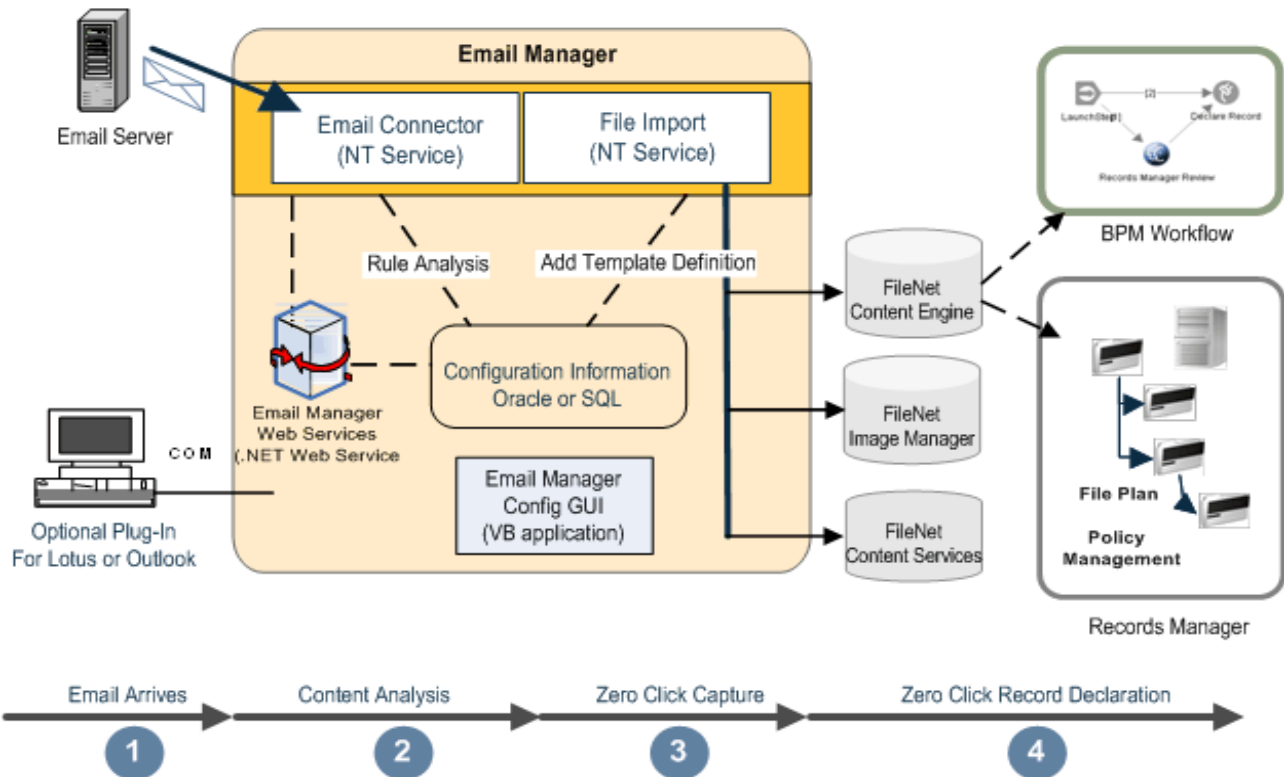
1. **E-mail arrives**: E-mail messages arrive at an e-mail server. A duplicate is made and deposited in the journaling folder while the original is passed to the recipient(s). There are 3 ways to capture e-mail messages: automatically with rules-based capture; manually by dragging or copying a message; or manual indexing.

2. **Content Analysis**: The properties, metadata, and content of each e-mail message are analyzed according to pre-defined business rules. If the e-mail triggers a business rule, the e-mail is flagged for capture into FileNet P8.

3. **Automatic Capture**: Without any user intervention, the e-mail is added to a FileNet P8 object store based upon the Email Manager "profile" defined by the Configuration Manager.

4. **FileNet ZeroClick Record Declaration**: E-mail messages can be declared as records at the time they are added to a Records Manager-enabled object store. The e-mail records can then be retrieved for subsequent processing, for example, as part of a business process management workflow, or the records can be disposed of according to a records management disposition schedule.

The following illustration depicts how Email Manager interacts with a mail server and a FileNet P8 object store:
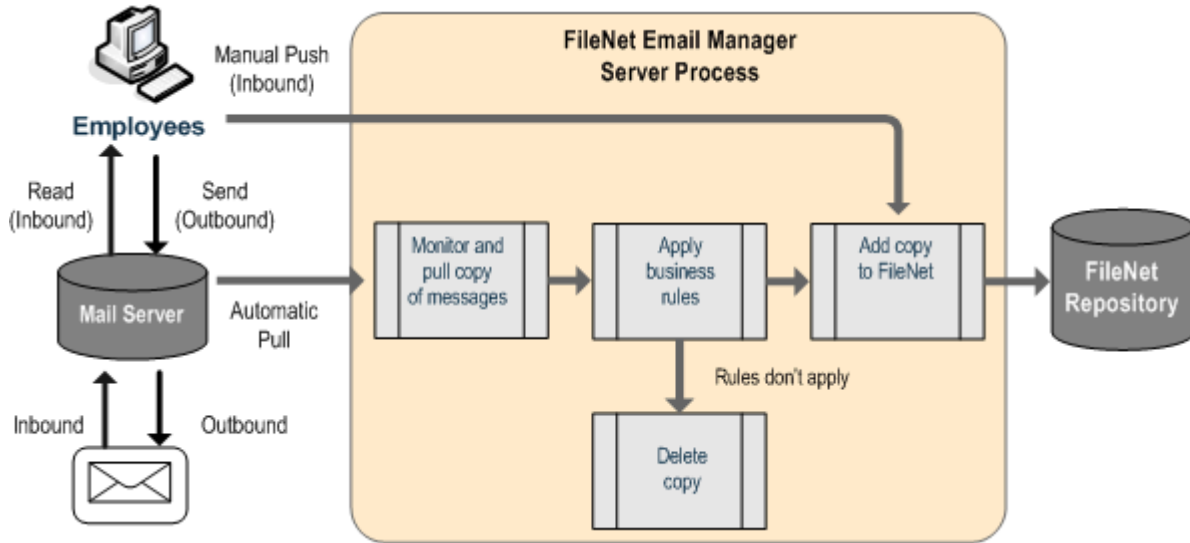
The Email Manager server is an unattended server process that automatically applies a set of configurable business rules to all incoming and outgoing e-mail to determine whether an e-mail should be added to a FileNet P8 object store. If the e-mail meets a specific rule, the Email Manager server process adds it to the object store according to the specifications in the profile associated with the rule. This process is called automatic capture.

The Email Manager client is either an Outlook Add-in, or is integrated with a Lotus Notes® or GroupWise client, each of which can be used to perform manual capture—selectively adding e-mail messages to the FileNet P8 object store.

IBM FileNet Connector for SharePoint Web Parts

IBM FileNet Connector for SharePoint Web Parts allows you to work with documents stored on a FileNet P8 object store. Web parts allow you to browse or search an object store, work with your documents, participate in business processes through personal and public inboxes, and manage the user profile database.
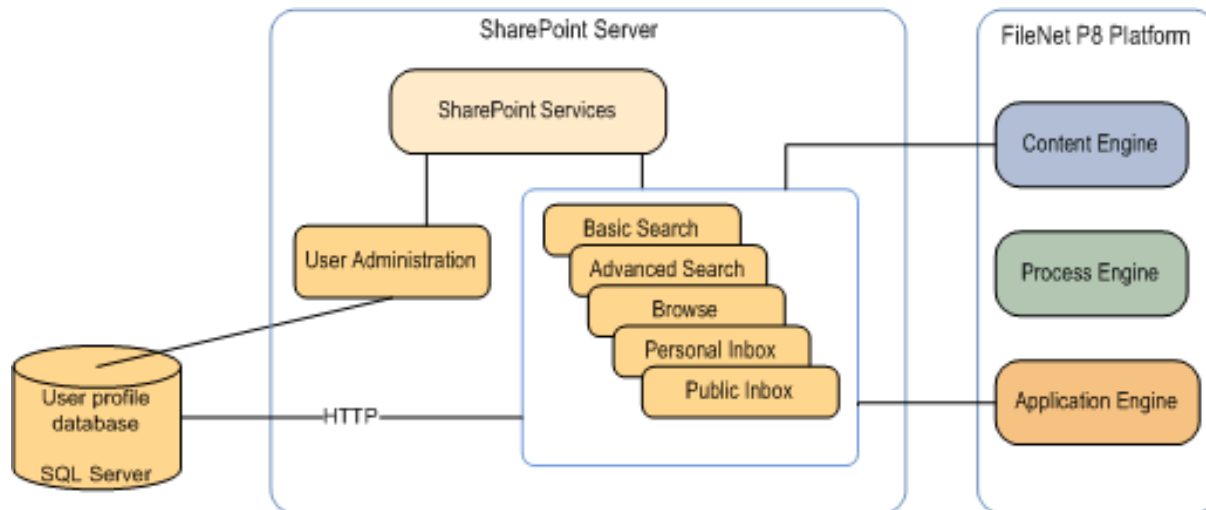
As shown in the preceding diagram, IBM FileNet Connector for SharePoint Web Parts is installed on a SharePoint server. The Web parts communicate with Content Engine and Application Engine, and indirectly with Process Engine, using Web services.

The user profile database links a user's SharePoint user account to the corresponding FileNet P8 user account; this profile allows the user to subsequently access the FileNet P8 system without logging in each time. You can locate the profile database on the SharePoint server or on a remote server. Web parts communicate with the profile database to retrieve and update user credential information.

IBM FileNet Connector for SharePoint Document Libraries

IBM FileNet Connector for SharePoint Document Libraries provides the ability to archive documents by moving or copying the documents from a SharePoint site document library to a FileNet P8 object store. When a SharePoint document is moved to an object store, vital information about that document, such as its assigned properties, can be mapped to corresponding properties provided by the FileNet P8 platform.

IBM FileNet Connector for SharePoint Document Libraries is comprised of the following components, as illustrated in the following diagram:

- The FileNet SharePoint Connector Service reads data from the SharePoint system and sends it to the Task Routing Service.

- The Task Routing Service processes task routing rules to determine the tasks to perform when moving a file from a SharePoint document library to a FileNet P8 object store.

- The FileNet P8 target connector contains the information required to access the Content Engine server. It also specifies the object stores where the SharePoint documents are to be transferred.

- The FileNet Remote Connector Web Service is a proxy that translates requests from a FileNet SharePoint Connector Service running on a separate machine into native SharePoint API calls and then converts the responses back again. The FileNet Remote Connector Web Service allows the source collector to run on a separate machine from the Task Routing Service.

- Configuration Manager is the desktop application used by IBM FileNet Connector for SharePoint Document Libraries and FileNet P8 Content Engine administrators, to set up and configure the system, define profiles and rules for SharePoint capture, and define post-processing options.
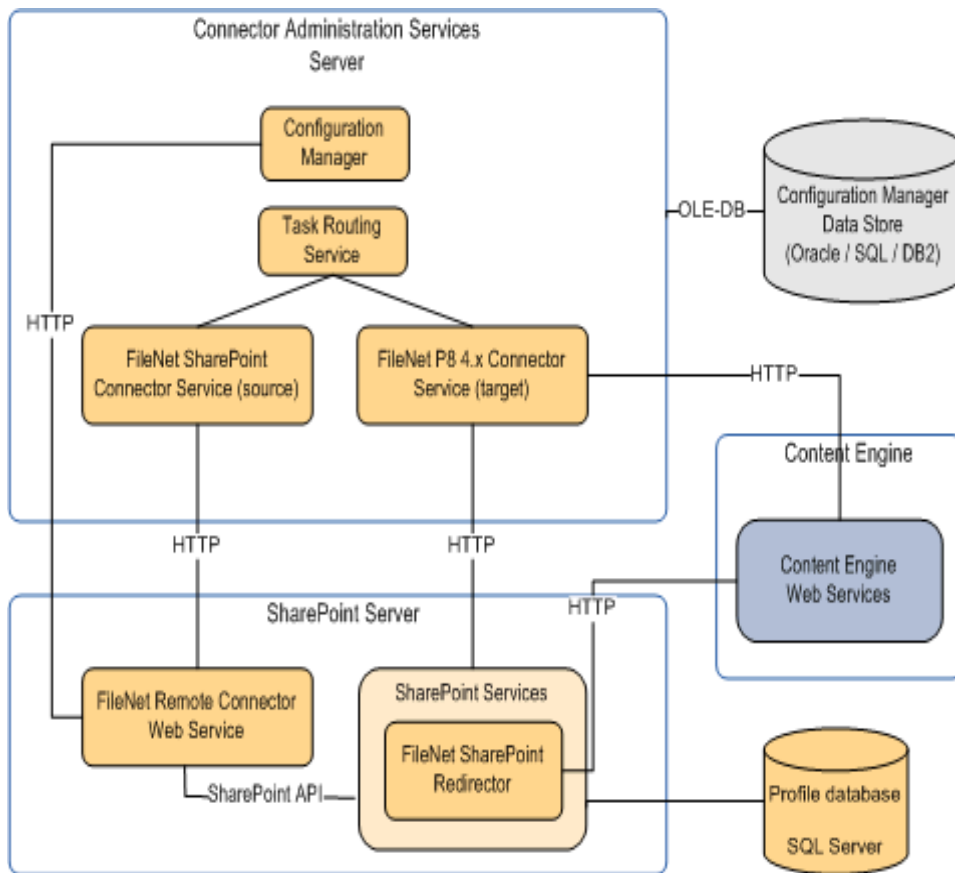
The preceding diagram shows IBM FileNet Connector for SharePoint Document Libraries installed on three physical machines: the Connector Administration Services server, the SharePoint server, and a Content Engine server. The servers communicate over Web services. You can optionally collocate the Configuration Manager Data Store with the Connector Administration Services Server and the profile database with the SharePoint server.

Application Integration

Application Integration integrates Microsoft Office 2003 and Microsoft Outlook 2003 and earlier versions with Workplace XT or Workplace. Application Integration communicates through the application integration components to Java servlets on the J2EE application server where Workplace XT or Workplace is installed. The Application Engine UI Service provides the user interface, which is displayed on the client by an ActiveX browser control. The re-use of the Workplace XT or Workplace user interface through the Application Engine UI Service on the Windows-based client provides rich feature parity between Office Integration and Workplace XT or Workplace.

Application Integration BCS

Application Integration BCS integrates Microsoft Office 2007 and Microsoft Outlook 2007 with Workplace XT. Application Integration communicates with FileNet P8 using a .NET API that is implemented on top of FileNet P8 Web services. Application Integration BCS leverages Microsoft Office 2007 VSTO technology to integrate with the Office Ribbon and to create custom task panes. Custom task panes use .NET user controls to expose FileNet P8 properties, navigation, and search functionality. The controls are stored in a common .NET library and are shared by Word, Excel, PowerPoint, and Outlook.

Application Connector for SAP R/3

The architecture of Application Connector for SAP R/3 (ACSAP R/3) has been largely dictated by the requirements of the SAP ArchiveLink interface.

The SAP ArchiveLink interface requires a server-side component that fulfils archiving-related requests, such as archiving/retrieving document or data objects, searching for a pattern in a print list, and handling background tasks such as automated barcode processing. This component has been implemented as the ACSAP R/3 Gateway to interface between SAP R/3 server processes and the FileNet P8 object stores. It has been designed to provide maximum empathy with the FileNet P8 architecture. It comprises four core subsystems that respectively handle configuration information, object store connectivity, as well as protocol and barcode processing.

The Configuration subsystem maintains the gateway's configuration information and provides a mechanism to share it with additional gateways when deployed in distributed environments. The Barcode subsystem interacts with SAP R/3 through remote invocation of RFC functions using SAP's Java Resource Adapter (JRA). The Protocol processing subsystem handles HTTP communication with the SAP R/3 system leveraging the HTTP capabilities of the underlying J2EE Web Application Server. The Content Repository subsystem is responsible for transferring content between FileNet P8 object stores, using FileNet P8 APIs.

ArchiveLink also requires client-side components for selecting, viewing, and linking documents that are stored in FileNet P8 object stores. For linking purposes, the FileNet P8 client communicates with the SAPGUI through OLE Automation. It also includes FileNet's zero foot-print Java Viewer for displaying documents served by the ACSAP R/3 Gateway.

A typical SAP R/3 configuration in connection with ACSAP R/3 and FileNet P8 object stores includes:

- One or several R/3 systems and related application servers and databases.

- One or several FileNet P8 object stores, coupled with optical or magnetic libraries, and possibly fronted by imaging application servers for distributed scanning operations.

- One or several ACSAP R/3 Gateways; the number of gateways is determined by the need for distributed processing and/or scalability.

- Capture workstations for large scale mailroom operations for scanning, QA, and indexing.

- A number of IDM Desktop for R/3 workstations for linking documents already committed to a FileNet Imaging back end.

- A number of thin client workstations for Web-enabled document viewing.

Administrative components

Dashboard

IBM FileNet Dashboard is a mechanism for collecting and distributing performance data on FileNet P8 products installed at a site, independent of the operating system. Dashboard is comprised of two major components:

- **Listener** – Receives a stream of events and other performance information from an application, and sends the data to a Manager for analysis and storage. A Listener is the equivalent of an SNMP "agent" and offers clients the ability to optionally accumulate and aggregate performance data over some period of time. A Listener waits passively for Managers to connect to it and to inquire about its performance counter values. In the absence of a Manager connection, the presence of a Listener component is nearly invisible to an outside observer. Dashboard offers Java and Microsoft Visual C++ implementations of the Listener.

  Data exposed by a Listener falls into one of three categories: environmental, static, and dynamic.

  - Environmental data is information about the application's operating environment, such as the operating system name and version number, the number and type of CPUs, the amount of available memory, and so on.

  - Static data is specific to the application that is implementing Dashboard features, but does not change as the application runs. Examples of static data are the application's name and version number. Static data for Java applications also includes information about the Java Runtime environment, such as its version number, maximum memory, and so on.

  - Dynamic data includes data supplied by the operating system, such as CPU load, in-use memory, network traffic, and so on, as well as application-specific data (counters, durations, and other data provided by the application to the Manager). Consequently, the amount and type of dynamic data varies depending on the level of instrumentation that is implemented in the application, and it can change from release to release of the product.

- **Manager** – Connects to and retrieves reports from a Listener and processes the collected data, such as storing it, displaying it graphically, or using it to perform other application-specific tasks. Dashboard provides two Managers: Dashboard and Archiving Manager.

    - **Dashboard** monitors performance.

    - **Archiving Manager** saves historical or real-time data collected from the Listeners in a cluster to files that you can subsequently load in the Dashboard for analysis, reporting, and so on.

The following diagram depicts the overall architecture of the Dashboard:
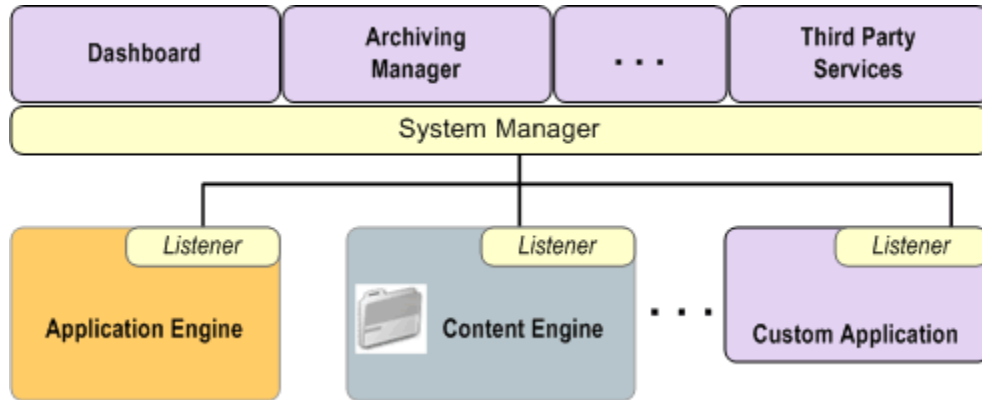
TCP sockets connect the Dashboard and the Listeners. Each Listener maintains a history of its application's events. Managers can request this historical information at any time. The Dashboard automatically requests this history as soon as it connects to each Listener. A Listener's history can be saved for later analysis using the Dashboard or a stand-alone Archiving Manager.

The following FileNet P8 components support Dashboard:

FileNet P8 Component	Components Included
Content Engine	• Listeners for Content Engine • Dashboard
Image Services Resource Adapter	• Listener for ISRA • Dashboard
Process Engine	• Listeners for Process Engine • Dashboard

System Monitor

IBM FileNet System Monitor collects and distributes performance data on FileNet P8 products installed at a site, independent of the operating system. System Monitor is comprised of the following components.

- **Management Server** – Receives, processes, and stores events from Management Agents. Events can be provided to the Web Console, used for standard reporting, or forwarded to other systems. Events are forwarded using e-mail or SNMP MIB, from which they can be imported into enterprise system management tools. Events can also be written to a System Monitor log file for analysis. This allows operators or help desk employees to see FileNet-related events in their management consoles rather than the Web Console.

- **Management Agent** – Monitors the health parameters (monitors and listeners) and log files. The Management Agent runs on each managed component, and can execute tasks such as restarting a component that has failed.

- **Web Console** – A browser-based graphical user interface that accesses the Event Console and the System Monitor tools such as the Task Execution Manager or the Reporting or Knowledge Base. The Event Console allows an administrator to see the current status of all managed components at a glance and to drill down into details to troubleshoot and identify the root cause of problems.

- **Monitoring Manager** – Graphical user interface used to add, delete and configure monitors on the managed components. You can configure the execution period of the monitor, the thresholds and related severities, and activities to be performed automatically if a certain threshold is met.

- **Task Execution Manager** – Graphical user interface which executes tasks on the managed components. System Monitor provides tasks for configuration of the systems as well as for the automation of routine work such as restarting a component or shutting down an entire FileNet P8 domain. Since the Task Execution Manager performs the task, the administrator does not need to know the details of how to do it.

Usage Reporter

IBM FileNet Usage Reporter consists of three components:

- **Listener** – Each supported FileNet P8 component reports usage events to the Listener in real time. The Listener summarizes the data over the course of the day, storing as much as seven days' worth of data in memory.

- **Usage Collector –** The Usage Collector runs on an automated schedule, typically once per day. It connects to each Listener by using TCP and downloads any new usage data. The Usage Collector stores the data permanently in a local database.

- **Usage Reporter –** An administrator can run the Usage Reporter at any point to generate reports. The Usage Reporter connects to the local database and displays the information. The information can be exported in various formats (CSV, HTML, or XML) for additional report analysis by other tools such as Excel.

Appendix A – Architectural diagrams

IBM FileNet P8 Architecture
HIGH LEVEL VIEW

Web Clients

Browser	Portal	WebDAV Client

Application Integration Toolkit

eForms Designer	Office, Outlook Integration

Presentation Layer

Records Manager	FileNet P8 Portlets	Image Manager	Workplace XT	Workplace	eForms

Business Logic

Services Layer

Content Engine	Process Engine	Image Services	Rendition Engine

Connectivity

Data Layer

File System	Business Applications	EMC Centera/ Snaplock	Database	Magnetic	Tape	Optical

IBM FileNet P8 Architecture
DETAIL VIEW

Presentation Layer

Records Manager	eForms	FileNet P8 Portlets	Workplace	Workplace XT	Servlet

Business Logic Layer

Records Manager Services

Forms Manager Services

Java Connector	RM Operations
JMS Connector	CE Operations

Component Integrator

Content & Process COM and Java APIs

ISRA

Web Clients

- Browser
- WebDAV Client
- Portal

Application Integration Toolkit

- eForms Designer
- Office, Outlook Integration

Services Layer

Web Services Clients | Web Services Clients

Content Engine

Process Engine

Image Services

Rendition Engine

EAI Connector

Process Analyzer	Process Simulator	Rules

Client Applications

- Enterprise Manager
- Capture
- Process Designer
- Rules
- Analytics Reporting (Excel)
- BAM

Data Layer

Repository

EMC Centera/ SnapLock	Database
File System	Image Services

Business Applications

Directory Services

OLAP

Database Warehouse

Repository

Optical	Database
MSAR	Magnetic Tape

Appendix B – Acronyms

ADSI – Active Directory Service Interface

API – Application Programming Interface

BAM – Business Activity Monitor

BPF – Business Process Framework

BPM – Business Process Management

BPMN – Business Process Modeling Notation

CEWS – Content Engine Web Services

CIFS – Common Internet File System

CFS-CS – Content Federation Services for Content Services

CFS-IICE – Content Federation Services for Information Integrator Content Edition

CFS-IS – Content Federation Services for Image Services

DCOM – Distributed Component Object Model

EAI – Enterprise Application Integration

ECM – Enterprise Content Management

IICE – Information Integrator Content Edition

IIOP – Internet Inter-ORB Protocol

LDAP – Lightweight Directory Access Protocol

NLD –Network Load Balancing

OLAP – On-Line Analytical Processing

OLEDB – Object Linking and Embedding Database

PEWS – Process Engine Web Services

RMI – Remote Messaging Interface

SNA – System Network Architecture

TCP/IP – Transmission Control Protocol/Internet Protocol

URN – Uniform Resource Name

XPDL – XML Process Definition Language

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all

conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

U.S. Patents Disclosure

This product incorporates technology covered by one or more of the following patents: U.S. Patent Numbers: 6,094,505; 5,768,416; 5,625,465; 5,369,508; 5,258,855.

IBM ®

Program Number: 5724-R74, 5724-R76, 5724-R81, 5724-R86

Printed in USA