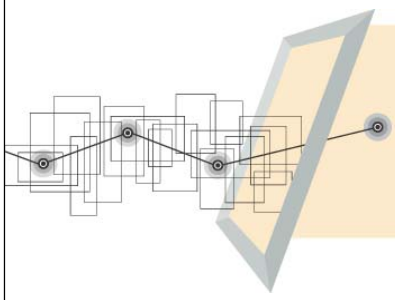
APPENDIX E

FileNet Business Process Manager



FileNet Business Process Manager

A Technical White Paper
August 2003

Introduction

Companies need every advantage to differentiate themselves in order to improve their competitive edge and ultimately to maximize shareholder value. Business processes, involving people, information systems and content, are woven throughout organizations, and represent unique paths of decision-making. They dictate the quality and efficiency of services delivered, and represent a tremendous opportunity to streamline, automate and optimize business operations to enhance business performance.

Companies are gaining a true competitive edge through ***Business Process Management*** (BPM), which automates, integrates and optimizes business processes to speed critical business decision-making, increase operational efficiency, lower the cost of doing business, and improve customer service.

Today's BPM solutions must be scalable, and deliver sophisticated capabilities with ease of use. They must provide an enterprise infrastructure that unifies business processes and related content with those business applications and systems that actually run and drive the business.

BPM solutions boost bottom-line performance because they let businesses deliver their products and services faster, more efficiently, more accurately, and at lower cost. BPM is an essential and required technology solution for companies because it enables businesses to define, control, and most importantly, optimize how processes and transactions are managed and how business decisions are made. BPM allows an unprecedented level of self-awareness so that these processes can be tuned to their optimal level, providing companies the means to react to issues before they become problems and identify opportunities with increased speed and agility – maximizing the value that they deliver to customers.

FileNet Business Process Manager is a solution suite based on the FileNet P8 architecture that delivers the enterprise scalability and performance required to handle the most complex business processes, the most demanding content challenges, and integration to all your existing business applications and systems.

This white paper will talk to the capabilities and benefits of FileNet Business Process Manager, and how it offers a unique BPM platform providing unparalleled levels of flexibility, power and performance to address the exacting demands of the organizations of today and tomorrow.

Table of Contents

The Power of Business Process Management

The power of BPM manifests by providing enterprises with the ability to extract every ounce of value from their business processes, thereby maximizing overall corporate value. The bottom line for BPM is that it enables enterprises to differentiate themselves, in order to successfully compete in their respective markets, and ultimately to help deliver maximum return to their shareholders.

BPM provides the following core benefits:

- **Reduced Transaction Processing Costs and Errors**
  Process automation reduces the need for human intervention in routine or low-risk business decisions. This eliminates both the cost and potential for errors associated with human-based workflow and allows companies to apply expert human resources when and where they add the most value to the decision-making process. Companies can then realign their resources to focus on higher revenue-generating activities.

- **Increased Business Agility**
  The ability to adjust processes on-the-fly offers companies increased business agility to address opportunities as market conditions changes. However, it is not enough for an organization to merely redefine its business processes; it must optimize these processes continuously. Processes supporting millions of transactions and thousands of users can be quickly deployed or modified as needs warrant.

- **Faster Decision Making, Streamlined Processes, Reduced Cycle Time**
  Automating processes reduces cycle time. The faster you can process a business transaction, the more transactions you process. The more transactions you can process, the more business you can service. Faster decision-making not only shortens cycle time, it reduces handling costs – and ultimately increases profitability.

- **Streamlined Exceptions Handling**
  No matter how well designed a process may be, there will always be work exceptions. An organization's exceptions-handling process is a major determinant of overall service. If an exception is handled expediently, it can present an opportunity to win customer loyalty. Through disciplined exception handling, companies can further differentiate themselves from their competition.

- **Enhanced Operational Visibility**
  BPM enables operational visibility, which is critical to both business agility and continuous process improvement. This enhanced operational visibility helps companies respond quickly to changing market conditions, so they can leverage windows of opportunity, minimize competitive threats, ensure regulatory compliance and make better decisions faster.

FileNet Business Process Manager

The FileNet Business Process Manager delivers on the value and promise of BPM by uniquely combining powerful process automation, comprehensive application integration and sophisticated process optimization services. In order to better understand the power of the FileNet Business Process Manager Suite, it is important to first understand the foundation capabilities provided by the underlying FileNet P8 architecture.

FileNet P8 Architecture

The FileNet P8 architecture (Figure 1) combines content, process and connectivity to facilitate highly scaleable, high performance Enterprise Content Management (ECM) and BPM solutions. Foundation capabilities include:

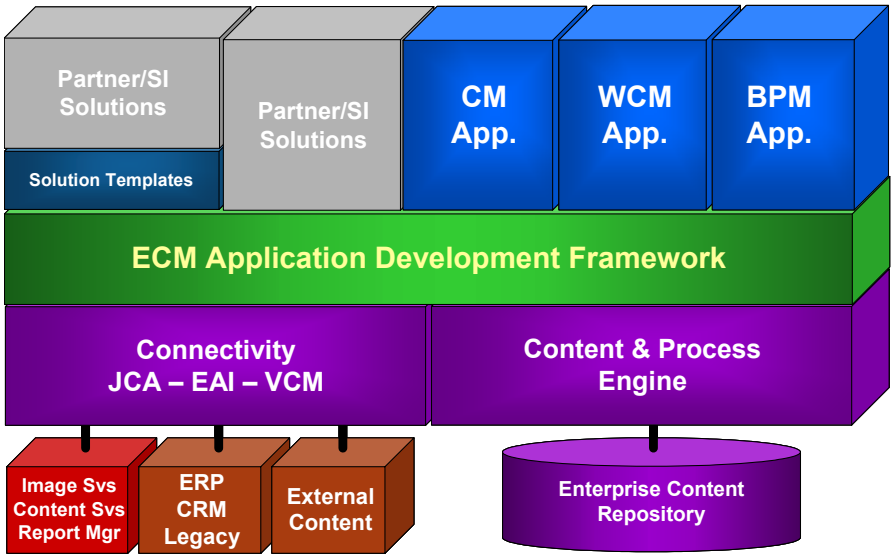


Figure 1- FileNet P8 architecture

ECM Application Development Framework or Application Engine

The Application Engine hosts the API and various process specific out-of-the-box tools and capabilities available to the user. All of the tools are web-based and can be accessed via a web browser.

- **Process Engine**

  The Process Engine is responsible for the management of the flow of work through business processes being managed; this also hosts the underlying database, unless a separate database server is configured. The Process Engine represents the sixth generation of FileNet BPM technology from delivering unheralded levels of functionality, scalability and performance

- **Content Engine**

  The Content Engine is responsible for the management all types of content, and provides a sophisticated content management repository. This content is stored and managed as a series of business objects, which could be scanned images, electronic documents or any digital content (e.g. video clips etc.). This capability is also leveraged to manage of the definitions of the business process definitions, which are stored as XML documents in the repository.

  The Content Engine is able to provide a full-featured ECM environment to manage both process definitions and other related enterprise content. The "Publish and Subscribe" event-based model at the heart of the Content Engine facilitates the highly agile active process management that is a fundamental part of the Business Process Manager.

All of the functionality is 100-percent web-based; both the application components and API are accessible via a browser. All that is required to perform any function is a browser and, of course, the appropriate security credentials.

The FileNet P8 Business Object

At the heart of the FileNet P8 architecture is the concept of the *Business Object*. The business object allows for constructs to be defined and related directly to the business transactions and associated events that are being managed. Business objects describe both the transaction and behavior that dictates how they perform in a business process.

This allows organizations to more accurately model how they want to perform business operations thereby facilitating the best possible business alignment.

Business objects are arranged in a hierarchy and as such, benefit from an object oriented representation by allowing attributes of the business objects to be reused and inherited.

Such attributes include: Data fields (including those that are externally referenced via EAI), Security, Process Definitions and Events.[1]

EXAMPLE: Loan Business Object

A loan business object (Figure 2) is created as a container for those data fields which are common for all loans, such as: a Loan Identifier, Loan Amount, Loan Term and Interest Rate.

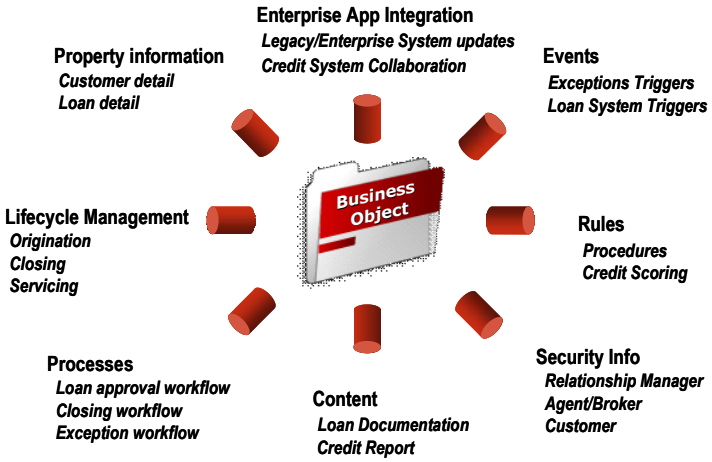


Figure 2 – a Loan business object

The various types of loans and the various loan-related transactions are subsequently represented as *subsets* of the Loan business object, inheriting the attributes of the Loan business object and incorporating additional attributes that distinguish the specific type of loan.

This significantly speeds the deployment of new implementations and more importantly, minimizes the total cost of ownership of such deployments, further driving down operational costs and increasing business agility.

Data Aggregation

Business objects can be used to aggregate all the relevant information pertaining to the specific business interaction being managed.

[1] See *"Event-based Architecture"*

The business object can contain links to content such as electronic documents or scanned images stored in a content repository or even a URL or UNC link to content stored outside the system.

Additionally, FileNet's Virtual Content Management capability provides access to content that is also stored and managed in other content repositories.

This data aggregation can be further enhanced through the use of EAI (Enterprise Application Integration), where data residing on other information systems can be dynamically linked to the business object and made available during transaction processing.

This ability to bring together all the relevant information pertaining to a particular business transaction ensures faster, better decisions by presenting users with all the relevant information needed to perform a given task so that they can focus on the task rather than searching for the latest information.

Event-based Architecture

The FileNet P8 architecture incorporates a sophisticated event-based model. This enables processes to be constructed as highly agile process segments rather than large monolithic processes. This approach results in a dramatic reduction in the total cost of ownership and increases the ability to react to change.

The Business Process Manager provides a "publish and subscribe" event model. As business objects are created, modified and deleted, theses events can be subscribed to and have processes associated with them. Consequently these "process segments" are launched the instant an event occurs; this ensures that processes are extremely receptive to change.

For example, once a mortgage application business object is created, the process for managing the application process is instantly created and executed. If the business object is subsequently modified (possibly by raising the amount to be borrowed, as a result of the borrower wishing to reduce the down payment), then this modification will instantly trigger an event that would update the existing business application and automatically invoke the appropriate review process as required.

Events can also be externally driven. Transaction events based in the enterprise business application, (such as an SAP, Siebel or legacy application) can trigger the invocation of a specific business process (process segment).

As previously mentioned, these event subscriptions can be inherited via the business object hierarchy, thereby minimizing the total cost of development, accelerating the speed of development, while at the same time maximizing process flexibility.

Business Transaction and Content Lifecycle Management

The FileNet P8 architecture facilitates the lifecycle management of the business transactions represented by the various business objects defined and stored in the Content Engine.

By subscribing to certain key events in the lifecycle of a particular business transaction and associating defined business processes with the occurrence of such events though a *workflow subscription*, it is possible to comprehensively orchestrate the transition between the various stages in the transactions lifecycle.

This capability ensures that throughout the entire lifespan of the business objects being managed, the Business Process Manager is invoking the right process at the right time, while retaining the highest level of business agility.

Business Process Manager Design Drivers

The design center for the FileNet Business Process Manager leverages the foundation FileNet P8 architecture capabilities noted above and incorporates the following key elements:

- Sophisticated Process Design and Ease of Use
- Business Agility
- Process Automation
- Seamless Integration
- Integrated Content and Transaction Lifecycle Management
- Process Tracking and Notification
- Process Performance Analysis
- Process Optimization Services
- Unmatched Scalability and Process Performance

These design center capabilities should be considered fundamental requirements and critical success factors when evaluating any BPM solution.

Process Design – Ease of Use

The unparalleled functionality offered by the FileNet P8 process definition environment facilitates the implementation of highly complex, interrelated processes. The integrated

expression builder allows complex business rules to be associated with the various routes between the steps.

Process Definitions

Processes are easily defined using the Process Designer (Figure 3), a web-based graphical design environment.

Process definitions consist of a series of process steps connected together by a series of routes, which define the sequence by which the steps are executed.

Defined processes are stored in the FileNet Content Services as XML. This graphical representation is the same representation that is used for process simulation and to present status information including existing process audit data (both subjects are discussed later in this document).

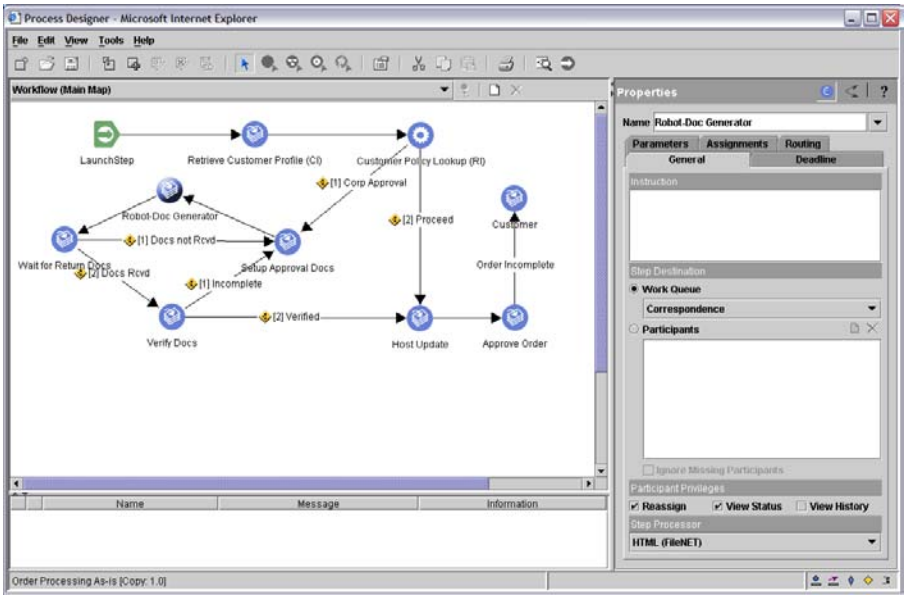


Figure 3 - the Process Designer interface

Process Steps

The steps in a process either represent a specific business task or a system activity. A business task can be executed by an individual user or group of users and/or automated application. There are 6 different step types (Figure 4) available for use in the Process Designer.

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 may be defined at runtime through the use of *Workflow Groups*, thereby a single process definition can exhibit

different behavior depending on the value of the user identities contained in the *Workflow Groups* defined for that specific process.

Figure 4 - the different Process Step types

A *Queue Step* routes work to a specified work queue where it may be accessed by either an individual user or automated application with access rights to that queue.

A *System Step* identifies one or more system functions that are to be executed during the process. These can include:

- Process *Timers* automatically invoking a process sub-map (discussed later) when a specific timer expires.
- Process *Checkpoints* provide the ability to "roll-back", either partially or completely, an item of work.
- Data Field *Assignments* assign values, either explicitly or implicitly based on data attributes of the item of work.
- *Delay* and *Wait for Condition* cause the item or work to pause in the process for a period of time and trigger the continued processing once an item of work has been processed or a condition has been fulfilled. *Create* Workflow causes the creation of another process.

A Sub-map Step routes work to a sub-map defined in the process definition. An individual sub-map can be referenced multiple times in a definition and multiple levels of nesting (i.e. sub-maps within sub-maps) are supported.

The *Launch Step* defines the step where the process instance is created; this only applies if the creation is done by a user using the out-of-the-box application, rather than achieving this programmatically via the API.

The *Component Step* defines a step where the execution of the task is carried out by a software component rather than a user. These software components are controlled via the Process Engine's *Component Manager* and are configured in the Process Configuration Console.

- A *Message Queue* – the Component Manager engine leverages Java Message Services (JMS) running on the application server platform to insert a message on a configure message queue. The message queue attributes are defined manually in the Process Configuration Console.
- A *Java Component* – the Component Manager invokes a specified interface contained within a java component. The interface definitions are imported by referencing the *.Jar file in the Process Configuration Console and selecting the interfaces to be made available in the Process Designer.
- A Web-service (available shortly) – the component manager invokes a Web Service. The Web Service definitions are imported by the Process Configuration Console by referencing either a WSDL file or UDDI registry.

Process Sub-maps

Each process can be developed using a collection of nested sub-processes, or sub-maps (Figure 5). Sub-maps can be specially defined in the Process Designer or they can be imported from another process definition acting as a process template.

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

Navigation around a complex process definition, including multiple sub-maps, is simplified in the definition environment. Users can move to the sub-map referenced by a sub-map step by simply double-clicking on the step icon with the mouse or browsing the available sub-maps via a drop down list box.

While viewing a sub-map, the display will show all the places in the process from where that particular sub-map is called; selecting one of the displayed sub-map names will display the associated process sub-map.

In addition to being called directly from another map, sub-maps can also be invoked by 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 sub-maps ensures the consistency of processing and considerably reduces cycle time and costs associated with the deployment of new processes.

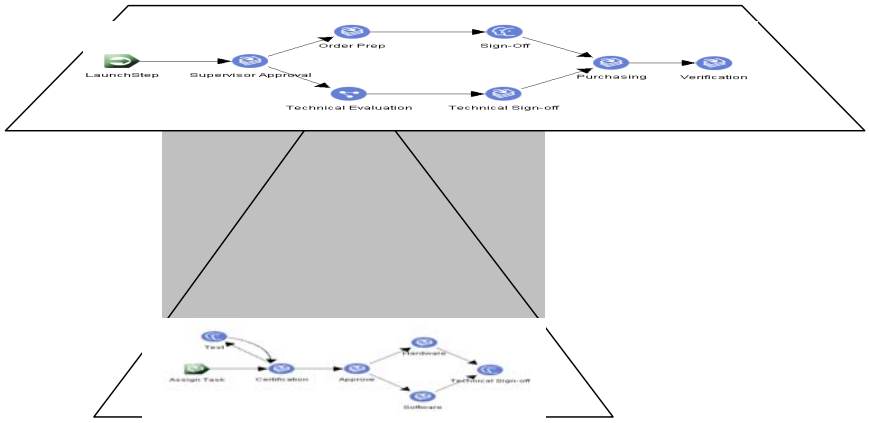


Figure 5 - inside a Sub-map step

Inter-related Processes

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.

EXAMPLE: Process Outsourcing

An example of the benefits of this capability involves process outsourcing, where the responsibility for the execution of a particular function is delegated to an outside service provider or business partner. Consequently, the outsourcing company should only have access 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 processed by the outside company. This process can be invoked and embedded into 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 sub-map, they can be distinguished from the main process. This allows an entirely separate audit log that can be tracked independently from the main process.

Business Agility

In the past, it was common to implement a business process that would remain static and relatively unchanged for considerable periods of time. Today's dynamic market requires deployed solutions have sufficient agility to embrace -- rather than resist -- change.

Process Version Management

Processes are defined as XML documents that are stored in the Content Engine. Storing the definitions in the Content Engine provides a mechanism for managing the lifecycle of the process definition by controlling user access and managing different versions of the same process.

EXAMPLE: Deployment of New Expense Approvals Procedure

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

- Check-out the existing process from the Content Engine,
- Modifying the process using the Process Designer, and
- Check-in the new version of process definition back into the Content Engine (includes the optional 'Transfer" (deployment) of the new process version.

Any new work created will follow the latest approved process map available, while any existing work in the system would remain unaffected.

Additionally, the new process definition may have associated attachments that outline revised guidelines under which any new expense reports must be evaluated. This attachment would reference a specific version of the guidelines (also stored in the Content Engine) revised to reflect to the policy changes. This ensures the correct guidelines are available to the participants in the approvals process for both new and pre-existing expense reports. This enables rapid implementation of changes to business processes, without requiring new versions of processes to be backwards-compatible. The resulting effect is dramatic reductions in time-to-deployment and ability to react quickly to business or transaction events, representing a significant competitive advantage.

On-line Configuration Changes

As existing processes change and new processes are defined, it may also be necessary to make modifications to the system (e.g. create new work queues, add new queue fields etc.).

The Process Configuration Console tool (Figure 6) can make changes to the system while work is being processed without requiring users to log off the system or restart the server software.

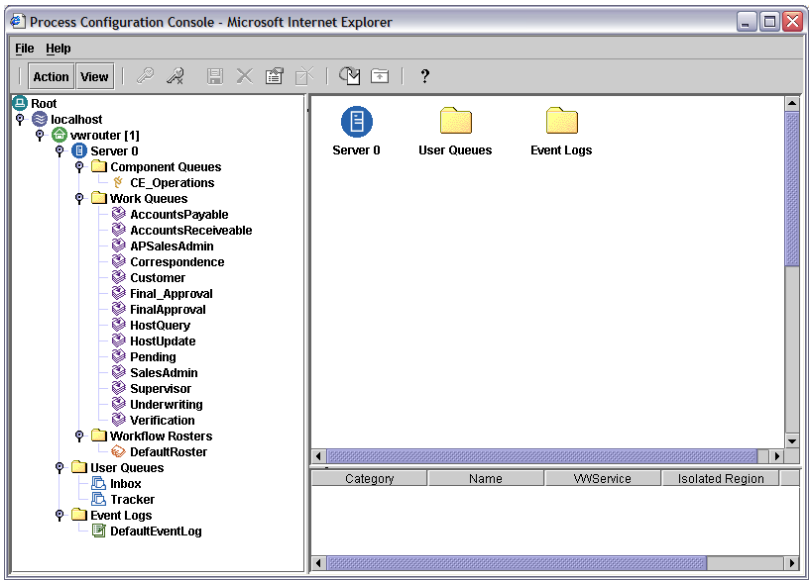


Figure 6 - the Process Configuration Console interface

The Process Configuration Console prompts the administrator with the number of items that will be affected by the modification, thereby allowing an assessment of whether to make the change immediately, or to reschedule it when the performance impact to the system will be minimized.

EXAMPLE: Creating a New Index on a Workqueue

If an improvement is required to how users query the contents of a work queue; this can be achieved through the definition of a new index on the workqueue, thus improving the performance and efficiency of the queries.

The creation of the new index is done via the Configuration Console (though this is also possible via the API).

The field or field names that make up the index are selected from those defined in the queue schema, and the change is then temporarily stored in memory, until the change is committed to the server.

When the change is committed, the user is prompted, as shown in Figure 7, and informed as to the number of items of work that will be affected by the change (in this case, the number of items in the queue); this is a good indicator of the impact of the change.

If, for example, the number was extensive, then the time required to implement the change would be greater, and this modification would best be made when the processing of work from this queue is low. Despite this change, any users' work not affected by the change would be unaffected by the progress of this modification.

It is inevitable that changes such as this will need to be made to optimize the performance of the system under ever-changing conditions, and it is vital that such changes can be made with minimum impact to users, thereby maximizing system availability and overall productivity.

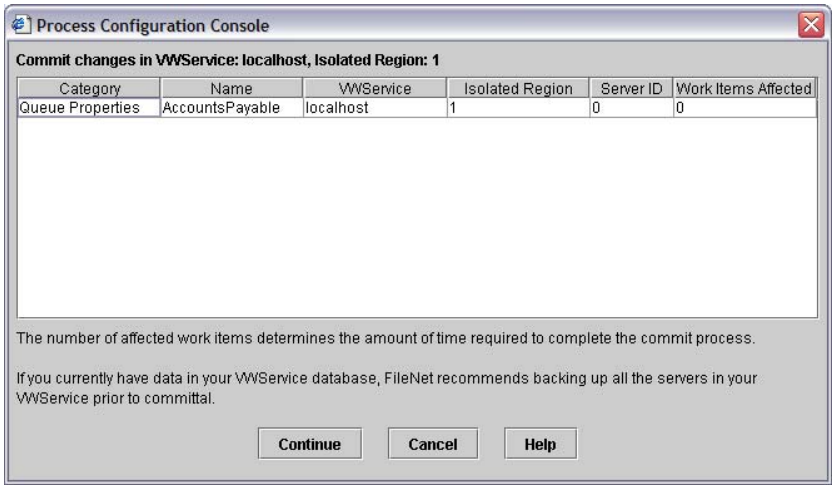


Figure 7 - interface notification of how many live work items affected prior to committing configuration modification

Process Automation - Accelerating how work gets done

Automatic Routing

By automatically routing work through the required sequence of tasks, the Business Process Manager entirely eliminates the time associated with moving work between the various required tasks. This automation also eliminates any delays associated with work being routed incorrectly due to human error. Automated routing can be based on specific rules and events – further accelerating how work gets done.

Resource Pools

The Business Process Manager allows for the deployment of resource pools by defining ***Workqueues***, which are shared among users with appropriate credentials to facilitate dynamic load balancing among users. The work processed from the queue can be further sub-divided so that the work a user sees is matched to their exact skill level.

Users processing work from such a queue may save the work to their ***Personal Inbox*** so they may return to it later, ensuring it will not be processed by another user in the interim.

However, this approach is not always appropriate, and there may be circumstances where a named user or, group of users, is required. This can be achieved via the use of ***Workflow Groups***, where a task is earmarked for processing by a workflow group. When

the work is routed to the task, the users contained in that workflow group at the time of routing receive the work. This capability allows processes to require certain tasks be carried out by specific individuals, thereby ensuring the right resources are properly engaged automatically.

Work Prioritization

The FileNet Business Process Manager can also determine the order in which tasks should appear in a queue. This can be based on any field in the work item. The default is FIFO (First-In-First-Out), but it could also be based on a priority of some other application-based field, such as required service level.

EXAMPLE: Automatic Process Escalation

The use of a queuing mechanism normally relies on FIFO processing. For automated tasks where actual processing time is very low, this will normally suffice. However, in the case of tasks requiring human intervention, a back-log 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.

To address this issue, the BPM suite provides a **Begin Timer** system step as shown in Figure 8. This Timer allows for deadlines to be set, where if the deadline expires before the Timer is deactivated (by passing through the corresponding Stop Timer system step), then the work is automatically routed to an escalation sub-map.

Figure 8 - a BeginTimer system step

To assist with escalation, an associated sub-map could represent a “fast-track” process, or could simply consist of an **Assign System** step that would 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; thereby ensuring the work meets the desired timeline and maintains the required level of service and performance.

The duration of the Timer can be set either at design time or at run time (i.e. based on data contained in the item of work). 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 running simultaneously on each item of work, each with a different escalation process, thereby increasing processing flexibility even further.

Additionally, more passive escalation measures can be implemented. A deadline can be specified for a participant to complete a step. This deadline indicates an amount of time relative to the time when the step was initially routed to the participant. Additionally, the system can be set up so the assigned participant receives a reminder of the pending deadline via e-mail.

If the participant does not complete the step within the specified time, an icon is displayed in front of the step name in the participant's inbox, and an email can be sent to a configurable list of supervisors. The list of associated supervisors can also be specific to each piece of work.

This automatic process escalation has the double 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.

Exception Processing

It is impossible to design a process that will cover all possible outcomes, therefore it is vital that the BPM platform be flexible enough to effectively handle exceptions as they arise.

Task Reassignment

The FileNet Business Process Manager offers users the ability to manually reassign processing to another named user at any step where this privilege has been granted. This is defined individually for each step in the process by a check box in the Process Designer.

When a user reassigns a step, they have two options:

- Delegation – This returns an item of work once the delegated user has completed the required work.
- Abdication – This 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. This allows for necessary process deviations, but retains the same level of audit control over the process, enabling a greater level of flexibility in the manner that work is processed. This reduces the necessity to “over-engineer” process definitions and the effort and time required for deployment.

Process Voting

In many cases, it is necessary for a single task to be carried out by a number of users collectively. These users can be identified explicitly or implicitly via *Workflow Groups.*

Therefore, it is vital that these actions be carried out in parallel rather than in sequence, in order to minimize processing time.

If each user 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.

Therefore the FileNet Business Process Manager provides native *voting* support, whereby 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 (e.g. all users select ***Approve***)
- Any of the responses being of a certain value (e.g. any one user selects ***Reject***)
- None of the responses being of a certain value (e.g. no users select ***Reject***)
- The count of responses of a certain value (e.g. more than one user selects ***Reject***)

These vote based routing criteria can also be combined with more conventional data based decision criteria, (e.g. the document type = "Press Release" and all users selected ***Approve***).

Users carrying out such parallel task can also be made aware of the responses of other users by ensuring that the task they are carrying out has the *View Status* flag set in the Process Designer.

This capability allows for decisions to be made collectively and ensures that, although processing times are dramatically reduced via parallel processing, the appropriate level of control over the process is always maintained.

Fully-integrated Content Management

Eighty percent of business information is stored in an unstructured format (i.e. not in a database), therefore it is important that this information be seamlessly integrated into the process.

This is achieved within the Business Process Manager by the full integration into the Content Engine. Cross-platform user authentication assures instant accessibility to all electronic documents and scanned images stored, while retaining the pre-existing security access controls.

One or more documents can be attached to work flowing through the system, providing users instant access to ALL the information required to process a particular task. In addition to stored documents, other forms of content may also be attached to the work in the form of URL's (Uniform Resource Location) and data files referenced using UNC (Universal Naming Convention), for example, files stored in the shared directory.

This means that from the instant that the FileNet Business Process Manager is installed, any process may reference any document stored without additional coding or user administration.

Seamless Integration

In order for the modeled process to incorporate all aspects of the standard process, it must include those steps carried out by users and those carried out automatically by applications. Therefore, it is crucial that the integration with such applications be seamless.

Application integration takes 3 forms:

- Automated steps – the execution of the task is entirely automated and involves the invocation of a secondary software component.
- Integrated Steps – this is where the execution task involves the presentation of information that resides on other systems at the time the task is executed by the user.
- Rules Engine integration – this where the determination of the nature of the flow through the process is delegated to an external Rules Engine.

Reusable EAI Transactions and Application Functions

Steps on a process map may not involve a user, but may in fact result in the execution of an automated task. Such tasks can be executed via the various technology interfaces previously described in the Component Integrator.

However, a work queue can be defined to be accessed by the third-party application. The definition of the queue includes predefined *queue operation* (including the required parameters), that corresponds to functions within the application. The application can be either a custom developed application or a specially developed EAI Adapter.

The EAI Adapter makes it possible to expose previously defined EAI transactions in the Process Definition environment, allowing them to simply be dragged and dropped into the appropriate location in the process definitions without additional coding. All that is required is that the function parameters be mapped to the appropriate work item data fields. Once a function has been defined, it may be reused across multiple processes without additional coding.

For example, the Process Engine could integrate with a customer database and access the following:

- Create a new customer record
- Check status of a customer record

- Modify status of customer record

A work queue customerDB is created using the Configuration Console (Figure 9). This is where work will be routed when interaction with the customer database is required.

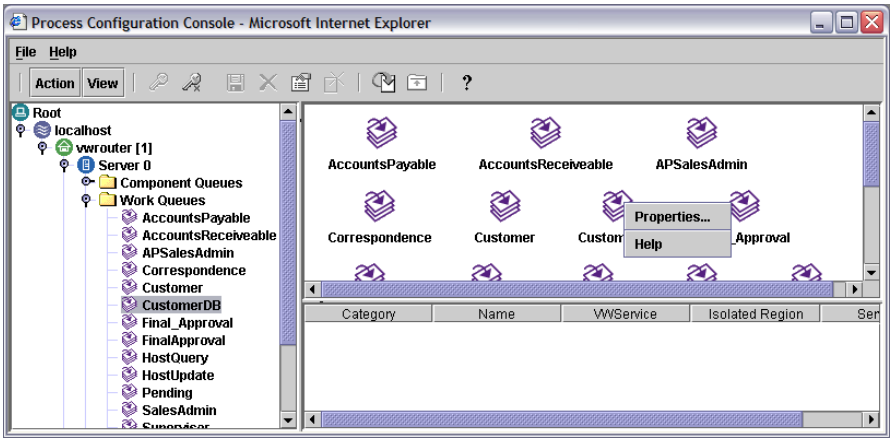


Figure 9 - selecting queue properties in the Configuration Console

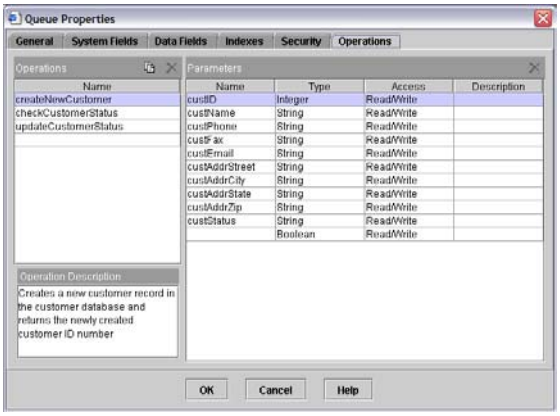


Figure 10 - interface displaying queue operation parameters

Once the work queue is created, operations are then defined to correspond with each of the desired database functions as shown in Figure 10.

This is accomplished by simply creating a step on a process map and selecting *customerDB* work queue At this point, the user would be then be presented with a list of available functions (Figure 11). When one is selected, the user is presented with the pre-configured list of parameters required by the selected function. These are in turn, then mapped to the available field in the process definition.

Once a queue operation has been defined and the associated application functionality has been implemented, it may instantly be reused over and over again, with no additional development, thereby reducing the cost of ownership and the time taken to implement subsequent processes that require the same functionality.

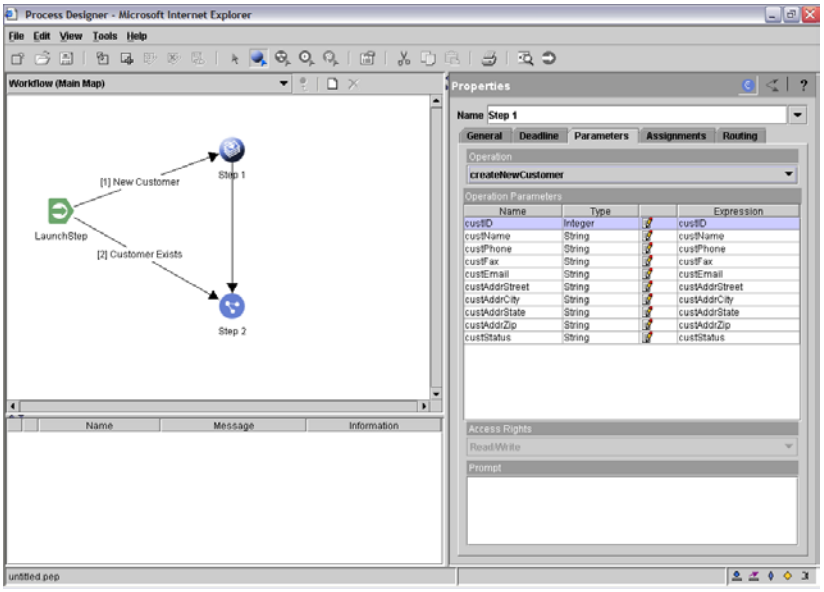


Figure 11 - queue operation parameters displayed in the Process Designer

Unified User Interface

Third-party applications may be accessed from within the *Step Processor* interface. Step Processors can be either a custom-developed application (either created from scratch or customized from those provided with the product), or the native Java step processors can be implemented.

100 Percent Open API

The FileNet P8 Process Engine is developed on an "Open API", meaning that all of the underlying functionality is exposed programmatically, resulting in an unmatched ability to seamlessly integrate all BPM functionality with any application. This approach ensures that the delivered solution provides an easy-to-use interface, providing easy access to both the right information and functionality.

EXAMPLE: Embedded Business Administration Application

If business management functions are centralized within a business management portal, all the business administrative functions are exposed via the API. In this case, it is possible to seamlessly incorporate these into a unified business management application, rather than force business managers to use yet another administration application.

Consequently, managers are able to access a single holistic view of business operations and associated content, enabling them to make better, faster business decisions.

BPM Suite's Foundation Rules Capabilities

The Process Designer includes a sophisticated expression builder which allows the various routes in a business process to be associated with the evaluation of particular business logic.

The seamless integration with an external rules engine enhances that powerful capability and elevates it to a new level of sophistication and flexibility.

Although the Process Designer includes rules capabilities, integrating with a separate rules engine provides a number of important benefits:

1. Business Empowerment
2. In-flight Process Changes
3. Multi-dimensional Processes

Business Empowerment

When the first process automation software evolved in the early 1980's, it was intended to provide a layer of abstraction from traditional software development environments. This was possible due to the fact that the tasks in a process change less frequently than the process in which they reside.

A business process definition prescribes the possible paths that work can take during processing. Because business processes will change less frequently than the rules which govern the flow of work though them, it is possible to create an even higher level of abstraction consisting of those same business rules.

This abstraction serves as a mechanism for providing business managers with direct control over how work is processed, without requiring them to be responsible for the entire process definition.

By empowering business managers with the ability to make rapid business decisions that directly affect the processing of work, this significantly increases an organization's agility, thereby maximizing an organization's ability to be competitive.

In-flight Process Changes

Making modifications to processes is extremely problematic when considering the work already flowing through the processes.

For example: If you wish to delete a step in a process, how do you deal with those items of work at that specific step?

The Process Engine allows for process definitions to be versioned, so that a new version of the process can be deployed without affecting the work in progress, which will continue to follow the process definition in force when it was created.

While this elegant approach simplifies and accelerates the manner in which new processes are incorporated, it does not, however address the processing of work underway.

The integration of a 3rd party rules engine allows for safe modifications of the business rules referenced by a process definition, due to the enforced independence between the process definition and the rules governing its behavior.

This has the benefit of allowing organizations to deploy and manage business processes that exhibit "zero-latency", i.e. where the processing of ***all*** work can be modified instantly and the power to do so is in the hands of business managers and not IT personnel.

Multi-dimensional Business Processes

Traditional BPM doctrine dictates that processes be illustrated as large, often complicated, graphical business definitions. These processes can be simplified through the use of smaller process components or sub maps, which can be called from the main process map. This normally requires that all possible paths are defined in such a manner leading to maps that are overly complex, have a significant total cost of ownership (TCO) and yet often still fail to take into account every eventuality.

Integration to 3rd party rules engines provides the solution to this conundrum, by allowing the relationships between one task and the next to be described in terms of rules rather than a physically described connection.

For example, the next set of tasks in a process could be one of a dozen sub-maps, and the determination of which precise sub-map is to be invoked could be referenced via a single business rule instead of 12 discrete routes in a process definition.

The integration ensures consistency between both the rules and process definition environments, so that when a data field or sub-map is created in the process definition environment, it is available to be referenced in the rule definition environment.

The ability to deploy processes in this fashion, results in processes that are more flexible, easier to understand and have a dramatically lower TCO.

FileNet/ILOG Rules Engine Integration

In addition to the foundation rules capability, the FileNet Business Process Manager allows for integration to 3rd party rules engine vendors such as ILOG. Currently, FileNet provides a connector directly to ILOG JRules Rules Engine. The FileNet P8 Process Engine and the JRules Rules Engine are installed on the same server as part of the implementation. Once installed, the software components will automatically detect the presence of one another and activate the appropriate user interface components.

There are two levels of integration; design-time and runtime.

Design-time Integration - The Process Designer

The design-time integration is concerned with integration between the respective authoring environments; the Process Designer and the Rules Builder.

Business processes are defined using the Process Designer. If the ILOG JRules Engine has been installed, then additional functionality in the Process Designer is activated.

This functionality provides the user with the ability to create *Rules Packages* that will be invoked at selected points in the business process.

When a rules package is invoked, the rules engine will (at runtime) evaluate the rules contained within the rules package until no more rules apply. This invocation of a rules package can be associated will any step in the process.

Exactly when during the completion of a given step the rules package is executed is fully configurable:

- **On Arrival**
  The rules package is evaluated as soon as the work item is queued for processing in the queue of the associated step.

- **On Save**
  The rules package is executed when the work item is saved by the user prior to completing the step. (e.g. the user completes only half the task and wishes to complete the reminder later)

- **On Completion**
  The rules package is invoked once the step has been completed.

Once the process definition has been completed (including referencing where and when the rule packages are to be called, the process definition is checked into the Content Engine and “transferred”.

The transfer of a process definition is analogous to the compilation of source code into an executable program, in that it creates the necessary run-time objects within the Process Engine required to run the process.

ILOG has installed the transfer of process definitions automatically; resulting in the creation of a *Rules Repository* associated with the Process Definition containing all of the rules packages defined in the process definition.

Process & Rules Version Management

In the event that the process definition is modified in the designer, creating a new version of the process, the transfer of this revised version automatically generates a new version of the Rules Package which then is applied against all instances of work undergoing processing for not just the latest version of the process definition, but for all versions.

Design-time Integration – The Rules Definer

The attributes of the process definition, such as data field and sub map names, are automatically created inside the rules repository and are readily accessible for inclusion in rules contained within the rules packages.

Within the rule definer there are a variety of process related attributes that are available for manipulation within the rules:

- **Data fields**
  All of the data fields defined (including system fields) defined in the process definition.
- **Workflow Groups**
  All of the workflow groups defined (including the system groups) defined in the process definition.
- **Attachments**
  All of the attachments defined in the process definition.
- **Sub-maps**
  All of the available sub-maps (including the system sub-maps) contained within the process definition.

In addition to being able to access all of the aforementioned process attributes the Rules Builder has the ability to construct rules that not only manipulate the attribute values, but the may also directly cause a modification to the flow through the process.

EXAMPLE 1: SKIPPING A STEP

During the evaluation of a rules package prior to a step being carried out, it may be determined that the step needs to be skipped and the work should move on to the next step in the process. The rules builder has the ability to invoke just such an action.

This results in the elimination of unnecessary work and maximizes volume of work undergoing Straight Though Processing (STP), while at the same time ensuring work routed to users requires a user to do a task, where the user is adding value, rather than simply manually moving the work on to the next step in the process.

EXAMPLE 2: CALLING A SUB-MAP

During the processing of a rules package it is possible to directly call a named sub-map, even if there is no physical link between the current step and the sub-map concerned.

Inside of the Rules Builder the user has direct access to all of the available sub-maps defined in the process definition and the invocation of any number of these maps can be referenced anywhere in the rules set.

This ability allows for the creation of highly complex, yet flexible processes that would be extremely difficult to map in the event that a predefined physical linkage was required.

Run-time Integration

At run-time, during the execution of a process, if a rules package is associated with a particular step, the entire contents of a work item are passed to the Rules Engine, where the rules contained within the rules package are applied until completion.

Once the rules package has acted on the data contained in the work object, it is passed back to the Process Engine (along with any modified data fields) for subsequent processing.

In addition, to modify the data in the work item, as previously mentioned, the rules package may also affect the flow of work through the process by one of the following actions:

- Skip Step
  This causes the current step to be treated as if it has been completed and causes the work to move on to the next step in the process.

- Call Sub-map
  This causes one of the defined sub-maps in the process to be invoiced. Once the end of this sub-map is reached, the flow returns to the step in the process from where it is called.

This extension of the Business Process Manager suite's ability to encapsulate business rules into the processes being managed greatly enhances the level of business agility that

can be realized. By allowing organizations to build a high degree of flexibility into the processes they deploy, and by empowering line of business managers with the ability to rapidly make and enact key business decisions wherever, and whenever, the need arises

FileNet does not OEM or resell the JRules component; this must be purchased from ILOG or already be installed.

Process Tracking and Notification

Given the importance of the process, it is vital to provide a high level of visibility within the organization to quantifiably assess the current level of performance. Processes must generate a comprehensive audit of all activities that take place as work is undergoing processing. Without a complete record of historical processing, it is impossible to assess operational performance and quantify subsequent improvements.

Comprehensive Audit Log

Business Process Manager's comprehensive auditing function is designed to provide total integrity through the audit log. Log messages are created in the same synchronous database transaction as the event they monitor or track. This is achieved without significantly impacting overall system performance.

The audit log may be extended to include custom data fields, recording the value of a particular data field at the time the event occurred. This extension of the audit schema may result in running out of available columns in the database if large numbers of custom fields are created to accommodate many different types of work. Consequently, it is possible to create multiple audit log tables, so events associated with different types of work can be recorded in different event logs. This allows for the recording of the values of particular data fields at the time when the event occurs. It is this information that provides the valuable addition of context to every event that is recorded.

EXAMPLE: Multi-time Zone Operations

Most organizations have operations that span multiple time zones. The FileNet P8 Process Engine renders all audit data to UTC Greenwich Mean Time, thus maintaining the correct chronology of all recorded events. Additionally, all time-dependent events such as deadlines and process timers are handled correctly. Although the data is stored in UTC, it is displayed in the user's local time zone.

The Tracker

Audit log information can be used to present a complete history of a given piece of work. Rather than merely providing a list of the accumulated log messages, the Business Process Manager assembles the relevant audit log entries and appropriate process definition.

The combination of this information is processed and presented by the *Tracker* (Figure 12), which provides an interactive graphical display of the path taken by the piece of work as it moves through the process, as well as its current status. With this capability, organizations can demonstrate and track how decisions were made – providing a complete audit trail for compliance management purposes.

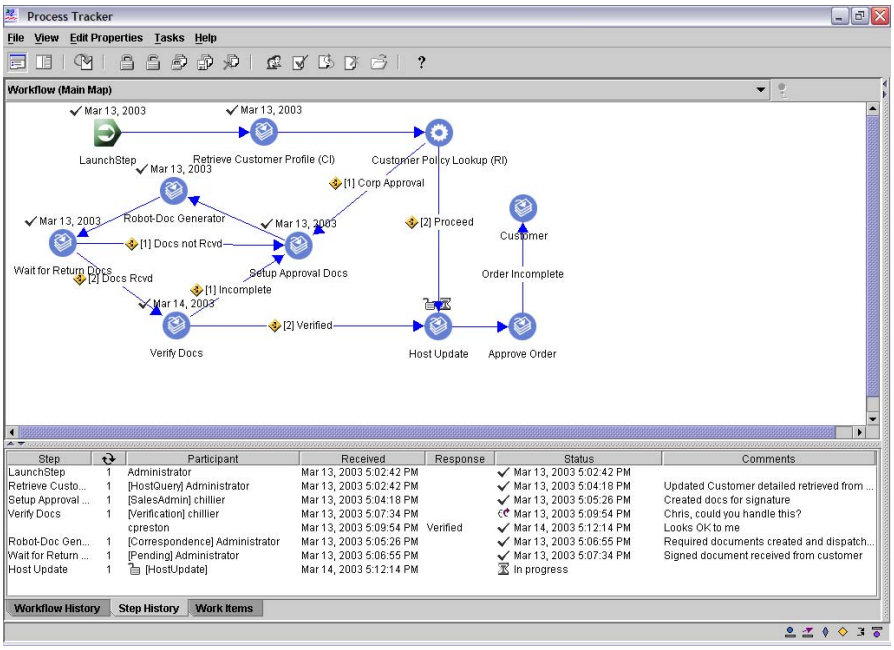


Figure 12 - the Process Tracker interface

The Tracker can be accessed from the Administration Tool, but also can be made available to users via the step processor or e-mail notification, and, of course, can be launched programmatically. This affords users executing specific tasks easy access to the complete history of the processes and a holistic view of how decisions were made around a specific work item.

This access is configurable at each step in the process, as there may be steps where this information is not necessary or, in fact, contrary to the objectives of the process.

EXAMPLE: Engineering Change Order

The Process Tracker provides a live/real-time snapshot of the status of the process. By selecting "refresh", the user may update the display to reflect any changes that have occurred since the original query was made.

The Tracker provides various views of audit trail information including:

- ***Workflow History*** - By selecting the workflow history tab, the user is presented with a high-level view of the process displaying the sequence in which the steps

in the process were executed. More detailed information can be viewed by selecting specific steps to drill down into the events that occurred within the step's execution.

- ***Step History*** - When a step is selected, the audit log for that step is presented, showing each individual action and the chronological sequence of events, including items which have looped back and passed through the step on more than one occasion.

- ***Work Status*** - The tracker interface also shows the location of any outstanding work items, as there may be more than one due to parallel routes and/or multiple assigned participants. The interface also indicates whether the item is currently undergoing processing.

- ***Email Notification*** - Email notification as events occur is critical for certain off-line users. In the user's in-box, there is a section devoted to user preferences where the user may define an e-mail address to which notifications can be sent. This address can be on any SMTP-compliant mail server, including third party-servers (e.g. Hotmail, Yahoo, etc.) or wireless servers associated with pagers or cell phones. Different users may use different servers and may modify addresses at any time.

Task Notification Settings

Users may subscribe to a variety of events. The message does not merely notify users via e-mail of the event, but embeds a URL link that takes the user to either an item of work or to the work's current status as displayed in the Tracker interface.

The Task Settings interface (Figure 13) in Workplace allows the configuration of the e-mail address to which the notifications are sent.

These notifications may be turned on or off for each individual user. These notification events include:

- ***New Assignment, Reminder and Expired Step Deadlines***
  Messages are sent when the user is assigned a new item of work (New Assignment), within a defined interval before the expiration of a step deadline (Reminder), and when the step deadline expires (Expired Step Deadline). The message contains a URL, and when the link is opened, the user is presented with the appropriate interface required to process the work.

  Messages can also be configured to notify users that are not necessarily participants in the process by have a requirement to be able track a particular instance of a process and be notified when particular event occurs. These users are referred to as "Trackers" and are defined as members of the F_Trackers workgroup.

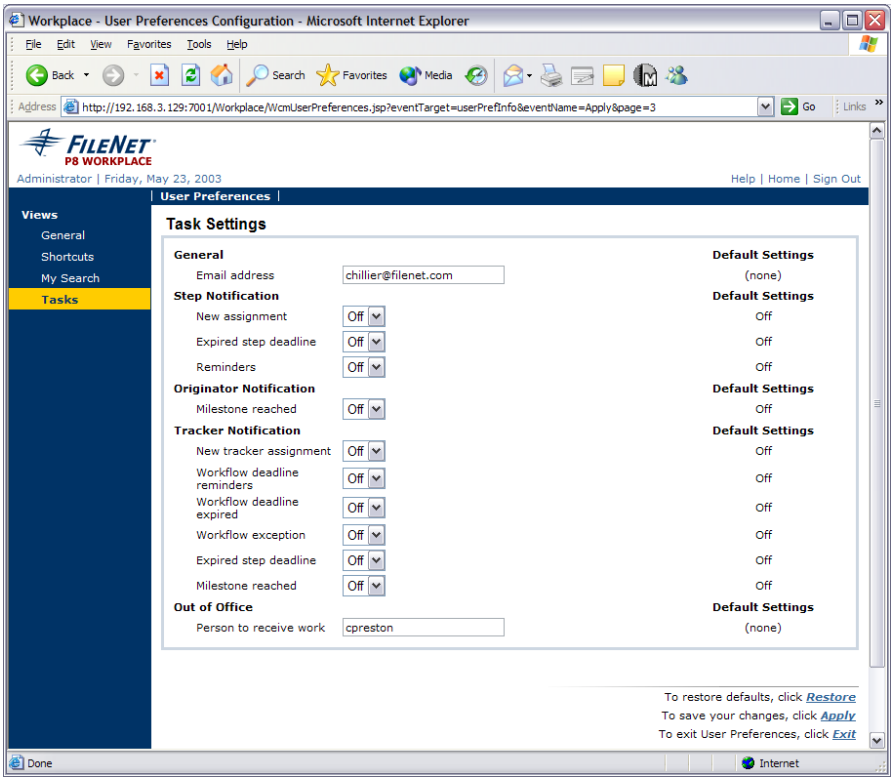


Figure 13 – the Task Settings user interface

- ***New Tracker Assignment***
  This message is sent to inform the user a workflow, to which they have been assigned as a "tracker," has been launched. This message contains a URL, which, when opened presents the user with the status of the workflow presented in the tracker interface.

- ***Workflow Deadline Reminder***
  This message is sent to all designated supervisors associated with a particular workflow at a pre-defined interval before the workflow deadline expires.

- ***Deadline Expired***
  This message is sent to all designated supervisors associated with a particular workflow when the workflow deadline expires.

- ***Workflow Exception***
  This message is sent to all designated supervisors associated with a particular workflow if an error occurs.

- ***Milestone Reached***
  This message is sent to all designated supervisors associated with particular workflow when a process milestone is reached.

These notifications ensure that even if a user does check the system regularly, they are regularly updated via email, or when appropriate, prompted/remained to carry out a particular task. This ensures that users continuously are aware of the work to be done so tasks do not go unprocessed.

Customer Self Management

Putting customers in charge of their interactions with an organization has significant benefits, including:

- Reduced costs and increased responsiveness
- Improved customer experience and service
- Increased customer loyalty

Indeed, empowering customers to take charge of their interactions with any organization has benefits for both parties. For the service provider, there are significant associated cost savings. For example, it typically costs 10-20 ***dollars*** to answer a query over the telephone, yet it costs only 10-20 ***cents*** to answer the same query via the Web. The customer benefits through improved access to information via the Web 24-hours-a-day. Additionally, information entered is automatically recorded so its accuracy is guaranteed. For example, a customer wishing to increase a credit card limit can fill out a Web-based form, this information is then fed directly into the process, eliminating the need for manual data entry, saving money and reducing the potential for errors.

Process Milestones

The process of making such information available via the Web is accomplished through process *Milestones.* Process Milestones (Figure 14) are defined points in a process definition that when reached, generate both a special type of event and an e-mail notification. Each milestone has both an associated message and a security level. Milestones can be set to trigger either before or after a step has been completed.

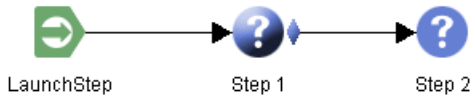


Figure 14 - a process completion milestone set at Step 1

The message associated with a milestone is defined in the Process Designer (Figure 15). It can be constructed in such a way as to reference the current values in the work item.

The type of milestone dictates the security levels on milestone messages stored in the audit log. For example, a Level 1 milestone may be viewed by customers, a Level 2 milestone by employees and partners, and Level 3 by employees only.

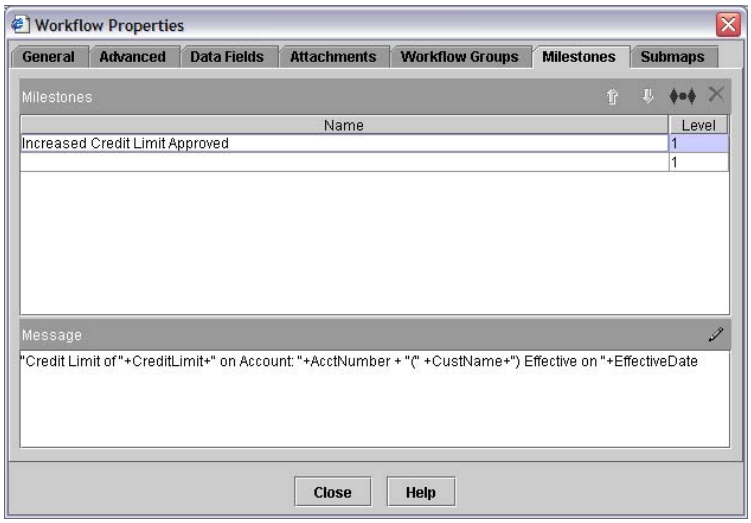


Figure 15 - defining a milestone in the Process Designer

EXAMPLE: Applying for Increased Credit Limit

In making a request to increase the credit limit associated with a credit card, a customer may wish to verify if the action has been completed and what, if any change, has been made to the credit limit. Normally this would be accomplished by calling a customer service representative.

Using the FileNet Business Process Manager, the customer may not only initiate the request by filling out a form on a Web site, which ensures accurate and automatic recording of the necessary information for the transaction to be processed, but they are also able to check the status of the request in real time.

The customer is given a request number which can then be used to provide real-time status of the request, so as the milestones associated with the approval of the change are completed, the information is made available to the customer via the Web site. Messages associated with milestones contain information that provide a clear context related to the items in process and/or completed actions.

10 May 2003 11:02 – Credit Limit of $10,000.00 on Account 2345-256373-31662 (Carl Hillier) effective on 12 May 2003

This provides a mechanism to provide customer self-service, where customers are presented with up-to-the-minute information regarding their interactions with the organization. In this example, the customer would be able to check the status of the request and receive the latest updates as the request is processed.

Process Performance Analysis – Awareness, Insight and Intelligence

The exponential growth of corporate data has evolved the need for faster, more granular, and flexible information processing. Business analysts and managers demand a mechanism for sifting through this data in order to visualize business strengths and weaknesses in a rapidly changing market place.

The delivery of such information enhances the visibility of business operations and the processes that underpin it.

Reporting and analytics is a vital part of the Business Process Manager. This is accomplished through the Process Analyzer, a reporting and analytics tool designed to help enterprises optimize business operations and increase returns from their Business Process Management investments.

The Process Analyzer resides on a separate server with its own separate database (Figure 16). This separation is vital in order to ensure that overall performance of the Process Engine is not affected by the activities of those users accessing the database for reporting and analytics purposes.

The data is transported from the Process Engine to the Process Analyzer via a continuous XML data stream, containing a variety of information including:

- System Configuration Information
- Process Definitions
- Process Event Data

The Process Analyzer consists of 5 OLAP data cubes:

1. Work In Progress (Real Time)
2. Work Item Cycle Time
3. Workflow Cycle Time
4. Queue Load
5. Work Load

The Process Analyzer provides visibility into business processes through comprehensive tracking metrics and reports.

The Process Analyzer collates and leverages both historical information stored in the audit and the current status information held in the system, and increases the level of process visibility by continuously updating a series of OLAP (On-Line Analytical Processing) data cubes with metrics related to the processes being managed.

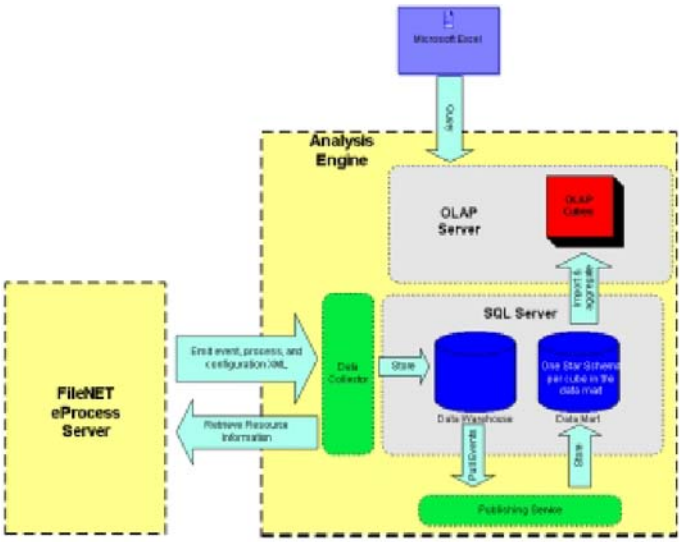
The use of OLAP technology ensures information can easily be reoriented to provide the required representation right when it matters the most. The data cubes are extendable to ensure information is presented in the appropriate business context resulting in more rapid, accurate business decisions.

This extensibility includes the addition of associated cost information, facilitating true quantitative cost-based analysis, a crucial component in any drive to reduce operational expenditure.

The OLAP cubes use an open data model, and therefore can be accessed by any third-party reporting tool capable of connecting to an OLAP data source. This enables invaluable information to be seamlessly integrated into an organization's pre-existing business reporting framework.

The data can also be presented and manipulated using Microsoft Excel, for which the Business Process Manager provides over 20 report templates.

Figure 16 - the Process Analyzer configuration



EXAMPLE: Mortgage Processing

The Process Analyzer can dramatically increase the insight into the way in which a mortgage vendor processes applications. The Process Analyzer provides instant access to vital business information such as:

Breakdown of mortgages by type currently being processed

The ***Work In Progress*** cube is a real-time cube that provides up to the minute information as to the different types of mortgage applications currently undergoing processing (Figure 17).

By publishing the Mortgage Type data field as a user defined field using the Process Analyzer configuration console, this then allows the volume of work in progress to be automatically categorized and presented in this way.

By providing such information in "real-time", organizations can react more quickly than ever before to the increasing speed of change in the business and ensure that available resources are optimally aligned with the current workload with which they are faced.

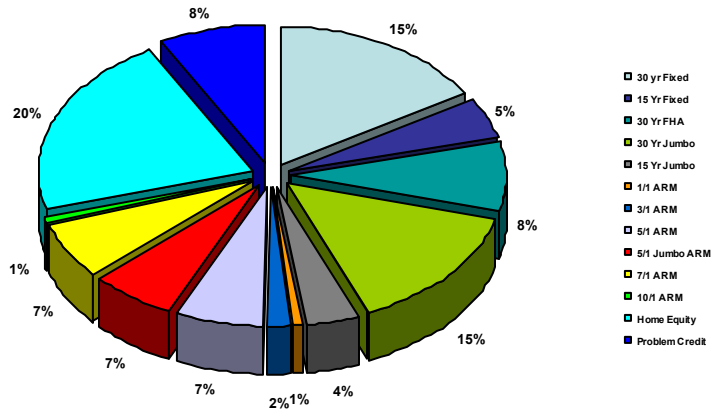


Figure 17 - a Process Analyzer chart displaying a "live" breakdown of mortgage types undergoing processing

Quantitative Cost-based Analysis

One of the key objectives in today's increasingly competitive marketplace is to minimize operational costs in order to maximize profits. Therefore, it is crucial that any insight afforded into processes must include a financial perspective.

The ***Work Item Cycle Time*** cube provides information regarding the various tasks that make up the process being managed. The addition of a calculated measure based on the

time taken to completed each task and the cost of the resources responsible for carrying it out, results in a complete picture as to where and when operational expenditure is being spent (Figure 18). This allows organization to easily and rapidly identify areas to focus on when striving to reduce operational expenditure.

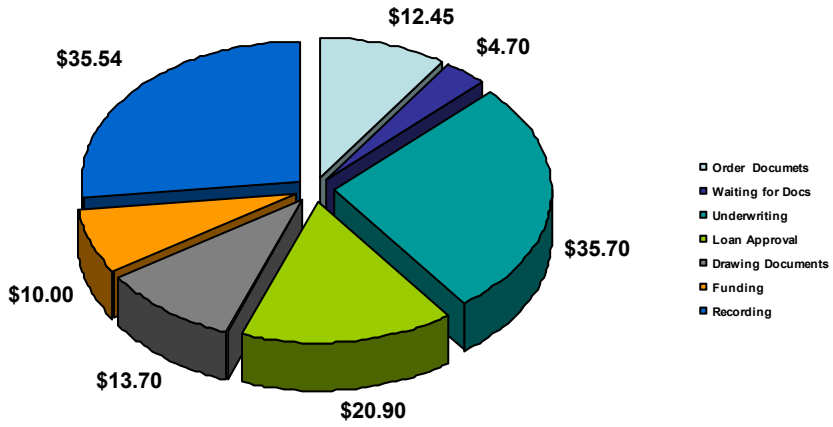


Figure 18 - a Process Analyzer chart showing a functional breakdown of mortgage processing operational costs

Quantitive Performance Assessment

One of the key differentiators in providing home loans is the ability to process the loan applications both accurately and rapidly. There is little point in providing an excellent loan with the best terms and least overhead, if it takes over 2 months to process the application!

With consumers increasingly embracing the Internet as a vehicle for enacting all manner of commercial transactions, a dramatic increase has been seen in the level of expectation consumers have for the speed at which transactions are completed. In short, consumers demand that transactions, which previously took days are now completed in hours or even minutes.

Therefore, the time taken to complete a business transaction with have a significant impact on an organizations ability to compete and the level of customer satisfaction it achieves.

As a consequence, a mortgage vendor will need to continually assess the time it takes to process each of the various home loan products it offers.

The ***Workflow Cycle Time*** cube tracks the performance of the processes, i.e. how long they take to complete. As before, this information could be broken down still further by

extending the cube to include other user defined parameters such as loan type (Figure 19), regional variations, loan amount and credit rating.

This information is tracked historically, making it possible for organizations to easily assess whether they are improving their performance in this regard and/or meeting specific performance goals as defined in their service level agreements or by regulatory authorities.

This information can also be used to help gauge the success of specific process improvements following their deployment.

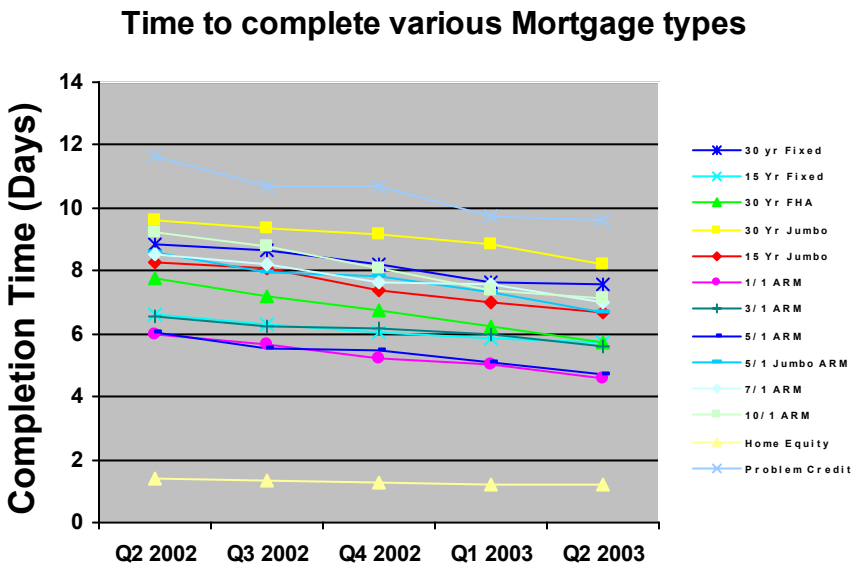


Figure 19 - a Process Analyzer chart plotting the decreasing completion time for various mortgage types

Resource Planning

In order to ensure that the available resources are deployed in the most efficient manner possible, it is important that there is an accurate assessment of the workload that will be placed upon them.

Most organizations will experience some level of fluctuation in workload due to time of year or other factors (such as reduction in interest rates in the case of mortgage processing).

The ***Queue Load*** cube tracks the volume of work moving in and out of the various task under the control of the Business Process Manager, specifically:

- Queue Load – the amount of work residing in a work queue

- Incoming – the amount of work arriving for processing
- Outgoing – the amount of work leaving the queue after being completed

This information can be presented as a historical plot to help predict the future queue loads placed upon specified resources in the future. Though effective resource planning, this can ensure that a consistently high level of performance is maintained, even though the queue loads placed on specific job functions may fluctuate.

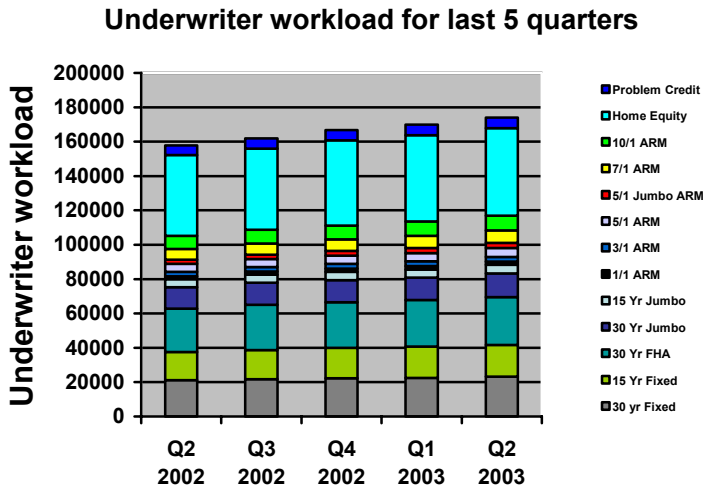


Figure 20 - a Process Analyzer chart showing the increasing underwriter workload broken down by mortgage type

The mortgage provider can use the analysis of the workload placed on the underwriters to view how it has changed over the last 12 months. The workload can be broken down further by mortgage type (Figure 20) or another sub-categorization such as state of origin.

This information is vital in ensuring that critical business functions are appropriately resourced. Only through an in-depth understanding of the demands placed on its resources can organizations ensure that resources are optimally deployed. In addition, such information is an invaluable asset when looking at optimizing the process beyond merely the deployment of resources.

Strategic Business Planning

Understanding the nature of the work being processed is of fundamental importance in effective strategic planning.

The Work Load cube tracks all work from the moment it is created until it leaves the system following its completion. This provides a historical context for this information which is presented, real-time, in the Work In Progress cube.

This information also provides both the necessarily high level view, and historical context, required to give a holistic view of work being processed.

The OLAP nature of the collected data allows for the information to be presented from a variety of different perspectives, such as breakdown of the work by mortgage type (Figure 21), amount borrowed, or even an applicant's credit score.

Such flexibility aids in the rapid identification of trends. Identifying trends sooner allows organizations to react first to changes in a fast moving market place, thus gaining a competitive advantage.

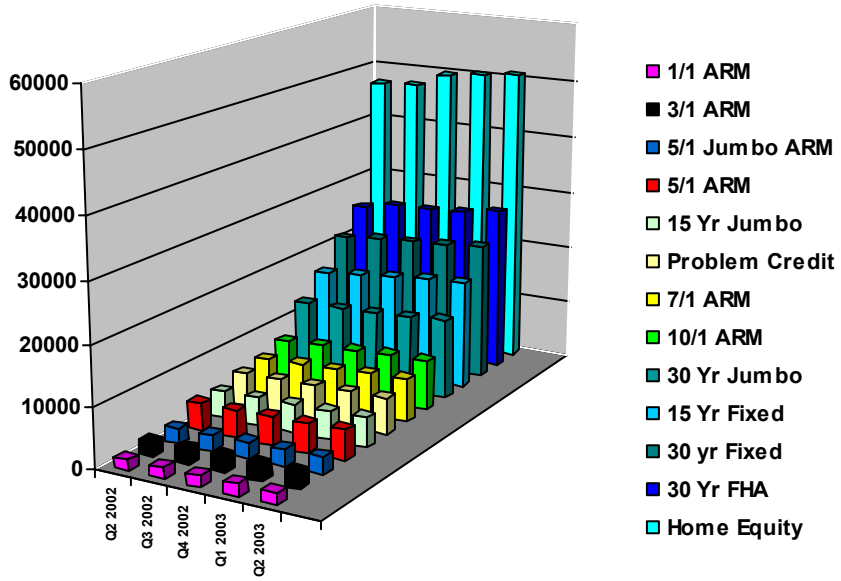


Figure 21 – Breakdown of Work by Mortgage Type

Process Optimization Services – Driving Continuous Process Improvement, Value and Performance

In the past, process-related projects enhanced the way that organizations worked, but once the project was over, the processes remained largely unchanged until the next project.

In today's fast moving marketplace, BPM projects never really end. Like a skipper trimming the sails of his yacht, organizations must continually make adjustments to ensure they are making the most of the prevailing conditions. (If they don't, they can be sure that someone will eventually steal the wind from their sails!)

To this end, organizations must view the enhancement of existing processes as a mission-critical business function. The ability to simulate a new process, or process enhancement, dramatically reduces the risk associated with the deployment of new processes, or refinements to existing ones, by validating every change before it is ever implemented.

The focus of an optimization effort typically revolves around:

- ❑ Reducing process cycle time
- ❑ Reducing operational costs
- ❑ Providing more efficient utilization of resources

Such continuous optimization directly contributes to the ROI and underlying value objectives of BPM. FileNet's Process Simulator is a fully integrated component of the Business Process Manager. It allows processes to be simulated prior to their deployment to determine their effectiveness. This ensures the deployment of the optimal process to deliver maximum value back to the organization.

The simulation environment (Figure 22) is consistent with that used to define business processes, ensuring the process to be simulated is as close a representation of the actual process as possible. Here organizations can use historical data or do "what-if" analysis around specific business scenarios to develop the most effective business processes.

If the simulation is being conducted before an initial deployment, the parameters added to the process definition will be based on the best information available at the time. Obviously, the accuracy of the resulting simulation will be in large part dependent on the quality of the data making up the aforementioned parameters.

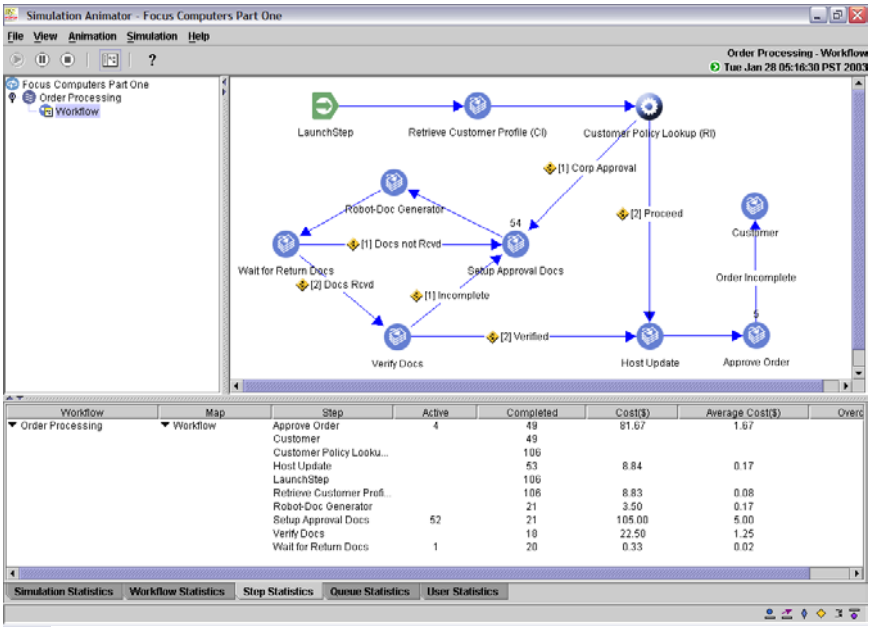


Figure 22 - the Process Simulation environment

Simulation Scenario Parameters

The Simulation Scenario Designer allows users to take one or more defined processes (as processes do not exist in isolation), and add additional data parameters concerning their execution, including:

- Resources
- Job duties
- Shifts
- Process route distributions
- Process "arrivals"
- Task cycle times

Resources

The process of creating a simulation scenario begins by defining the resources in relation to the actions that are to be modeled. These resources are representative of the participants in the processes -- be they human or systems responsible for executing various automated transactions.

The system assigns a cost to each resource to facilitate cost-based analysis. Each resource is categorized as one of the following: Participant, Workgroup, or System.

Job Duties

Each resource has associated with it a collection of skills that are represented by a set of work queues. Each job duty requires various roles performed by the resources during the normal operation of the processes being simulated. Each defined job duty is then allocated resources that are responsible for carrying out the role.

Shifts

Shifts are then simulated as the specified resources are carrying out their assigned job duties. Multiple shifts are defined; the resources and roles making up each shift can be individually tailored to imitate those available during actual processing.

Process Route Distributions

In order to simulate the various activities of users and systems as work flows through the processes being automated, it is necessary to define the behavior of work flowing through each of the processes.

This is achieved by associating percentage weightings with each decision path within the process definition (each time there is a point in the process where there are multiple pathways following a step). This ensures that work flows through the process as it would during a real-world scenario.

Process "Arrivals"

Once the process behavior is defined, the volume of work being created must also be modeled. This includes the type of work and when, during the simulation, this work is created.

Thought should be given to over what period the work will be created. For example, will work be created evenly throughout the day or will there be a peak in the morning associated with work initiated by the arrival of mail? Just as with shifts, multiple arrival patterns can be defined to accurately reflect a complex business-operating environment.

End-to-End Process Analysis and Optimization

If the simulation is to be an enhanced version of an existing process already in production, then the information collected regarding this process and stored in the Process Analyzer can be fed directly into the simulation scenario. This ability to directly leverage the acquired insight into the existing process greatly improves the level of accuracy of the simulation data. Conversely, for more detailed analysis, simulation data can be fed back to the Process Analyzer for greater scrutiny.

Task Cycle Times

Perhaps the most crucial aspect of the simulation is the definition of the time required to complete the various component tasks that comprise the simulated business processes. This will have a significant impact on the projected process cycle times and the associated costs. The average time taken for each task must be defined in order for the simulation to be meaningful.

Once again, if the simulation represents a modified version of an existing process, the definition of each task's cycle time can be leveraged from data already collected by the Process Analyzer. This ensures that the results of the simulation are as accurate as possible.

Consistent Simulation Environment

Process simulations are only worthwhile if they are both accurate and provide the information necessary to assess whether a given process will provide the desired results upon implementation.

Unlike many process management tools that provide mere simulation capabilities, FileNet's P8 architecture maximizes the accuracy of the simulated processes by ensuring

that the representation of the processes in the simulation environment is identical to that which is to be used in any subsequent implementation.

This differs significantly from other vendor's workflow products that rely on 3rd party analysis tools. These tools represent the process maps and capabilities differently from the workflow vendor's tool, creating a fundamental disconnect between the two environments.

With the FileNet Business Process Manager, the Process Simulator uses the same process design maps from the Process Designer. In this way, as changes are made to the original Process Map, changes can be directly re-instituted back into the business process.

However, it is not enough to represent processes as accurately as possible. It is also vital to represent the behavior of the work flowing through them and the actions of the resources that execute them. As data relating to work volumes and task cycle times collected by the Process Analyzer is fed directly into the simulation environment, it adds an additional level of realism to the simulation, further enhancing accuracy.

The ability to simulate a process does not lose its value once the process has been implemented. On the contrary, the accuracy of the simulation can be assessed as the simulation generates exactly the same analytics data as the real-life implementation. This ensures the business intelligence upon which critical business decisions are made is continually refined and improved; assuring decisions are made with the best possible information.

Unmatched Scalability and Process Performance

Scalability

The Process Engine that underpins the Business Process Manager represents the 6th generation of FileNet's process technology, and is the result of many years of expertise and millions of dollars of R & D investment.

The Process Engine has been architected from the ground up to manage millions of business transactions and support thousands of simultaneous users.

Failover and Recovery

In today's global economy organizations must be able to deliver around-the-clock service and therefore must process work when required at any time and any place necessary. Factors such as different time zones must be incorporated into processes to ensure that each transaction is viewed from a global perspective. The systems responsible for the processing of work are robustly designed to deliver 24/7 service

Process Recovery

Occasionally, a process itself may fail. This may be due to erroneous data contained in the process – or incorrect data from a user or business application or network stoppage.

The FileNet Business Process Manager addresses this important issue by allowing the definition of exception process sub-maps, which are automatically invoked in the event that the work cannot be routed to the next destination in the process map.

This ensures that such items of work do not remain unprocessed, but are automatically routed either to a manual or, where possible, automatic repair process.

The FileNet P8 Process Engine extensively leverages the capabilities of the support database environments (i.e. Oracle and Microsoft SQL) and each transaction is fully journaled by the database, with no information residing in memory. This highly robust architecture ensures that in the event of hardware or software failure, the current state of a transaction is known and is automatically completed once the system goes back online.

This capability also facilitates the deployment of the Process Engine in a high availability scenario where the database tables reside on a shared device, which could be split between two or more locations (Figure 23). When the primary server fails, the standby server takes over, and the storage device and transactions continue operating upon resumption of service, without loss of data and the need for users to log back on to the system.

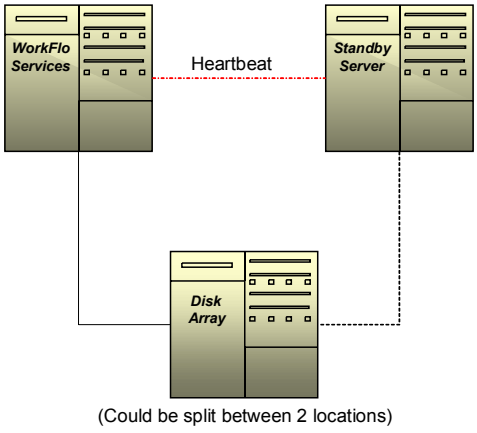


Figure 23 – the Process Engine deployed in a High Availability configuration

This ensures that in the event of a server hardware failure, the system recovers both automatically and, most importantly rapidly, ensuring the required level of server availability.

The Web servers can also be configured in a Web farm to avoid a single point of failure in the configuration, thus delivering the same high level of availability for the Web servers as the back-end servers.

Summary

The FileNet Business Process Manager is designed to provide a robust framework to automate, integrate and optimize business processes to speed critical business decision-making, increase organizational responsiveness and efficiency, and lower the overall cost of doing business.

Dramatic costs savings can be realized by the seamless integration of existing LOB applications, automating process steps where appropriate, and leveraging existing EAI infrastructures or integrating via standard technology interfaces.

Those steps that require user intervention are accelerated by the aggregation of all the information required to carry out a particular task, including scanned images, electronic documents and structured data residing on other information system.

The rich process definition environment combined with the Business Process Manager's event based architecture result in the ability to create powerful, yet agile business processes deployed in an environment that embraces change rather than resists it.

The FileNet Business Process Manager provides a unified process environment to maximize opportunities for automation and simplifies the user experience, resulting in dramatic productivity gains. The ability to reuse processes allows for drastic reductions in the time and costs associated with implanting either new processes or process modifications.

The FileNet Business Process Manager can facilitate the delivery of information both to and from business partners and customers. The ability to interact directly with all constituents by providing up to the minute status of work items and other process related details via the Web drastically reduces costs - while maintaining high levels of customer intimacy and satisfaction, 24-hours-a-day.

In today's fast moving marketplaces, it is vital that organizations assess their performance not merely periodically, but as a continuous activity, allowing process modifications to be made without interrupting the processing of work. The fully integrated analytics and simulation within the Business Process Manager increases the visibility of the performance of critical business processes and provides the means for continuous process improvement.

The Business Process Manager provides an enterprise BPM solution that is able to handle highly complex interrelated processes, managing millions of transactions and thousands of users, with a robust infrastructure that can be incorporated into high availability configurations, supporting 24/7 business operations across multiple time zones. As a result, organizations can rely on the FileNet Business Process Manager to increase process performance, reduce cycle times, and improve the overall productivity by optimizing the flow of work throughout their entire enterprise.

About FileNet

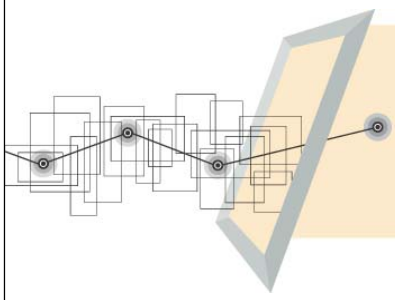
FileNet Corporation (NASDAQ: FILE) helps organizations make better decisions by managing the content and processes that drive their business. FileNet's ECM solutions allow customers to build and sustain competitive advantage by managing content throughout their organization, automating and streamlining their business processes, and providing the full-spectrum of connectivity needed to simplify their critical and everyday decision-making.

Since our founding in 1982, more than 3,900 organizations, including 80 of the Fortune 100, have come to depend on FileNet solutions for help in managing their mission-critical content and processes.

Headquartered in Costa Mesa, California, FileNet markets its innovative solutions in more than 90 countries through its own global sales, professional services and support organizations, as well as through its ValueNet® Partner network of system integrators, value-added resellers, and application developers.

APPENDIX F

The Event Drive Architecture of FileNet P8



The Event Driven Architecture of FileNet P8

A Technical White Paper
January 2003

Introduction

The FileNet P8 Architecture provides the industry's broadest set of integrated ECM capabilities by uniquely combining content, process and connectivity to solve real-world business problems. It is based on an open, standards based architecture that leverages industry standards like J2EE and XML for an open enterprise architecture that protects investment and provides the flexibility to integrate with other infrastructures and application technologies as needed.

The concept of FileNet P8 Business Objects supports an 'event driven system architecture' that allows companies to directly react to customer demands and system events to increase organizational and customer responsiveness and make structured and unstructured content an active, 'intelligent' part of business processes. It introduces enhanced process modeling constructs for increased reuse and improved modeling.

The FileNet P8 Business Object provides three primary integration models for incorporating FileNet P8 content and process management into companies existing processes. External applications can initiate and synchronize with FileNet P8 workflows via the direct integration model or through the event driven model. These models also enable external applications to access FileNet P8 content. The third integration model, the Content-driven integration model, enables the triggering of both FileNet P8 processes and external processes on key content events – for instance, a customer update to a proposal may trigger FileNet P8 processes to review the customer changes and record this interaction to the CRM application.

Table of Contents

BUSINESS OBJECTS

This section discusses some high level capabilities of the Business Object. The Business Object provides an event driven model to support seamless integration of FileNet P8 process and content capabilities with external applications.

Event driven model

The business object provides an abstraction layer and an event driven model for centralizing management of processes within a single system. Without the business object, applications that leverage FileNet P8 workflows must directly call FileNet P8 workflows. They will have to have detailed knowledge of the workflows (workflow name, parameters) that they wish to initiate. These applications will have to be updated when changes in workflows require different parameters. As the number of applications that utilize FileNet P8 workflows grows, code management becomes complex.

The Business Object can be the mechanism through which external applications interface. External applications can trigger key events on Business objects (update, add, delete), and the FileNet P8 business object can manage the workflows and automations that are triggered by these events. This abstraction transfers the management of FileNet P8 specific process information from the calling application into FileNet P8 objects. This centralized management in conjunction with the triggering subsystem enables support for multiple triggering applications (for example, from the Web and from Clarify) through a single business object.

By centralizing management of workflows and automations within the business object, calling applications have simplified interfaces to the FileNet P8 BPM architecture and do not have to maintain a high level of code-level knowledge of FileNet P8 constructs. Overall system complexity is reduced, along with the cost and time for managing this system.

Enhanced Process Modeling

Business Objects introduce enhanced process modeling constructs, such as Object Oriented process design and advanced state management. Business objects are object oriented, so system designers will be able to achieve a high level of reuse from both a data and a behavioral perspective. Advanced state management enables system designers to model complex processes that have lifecycles where each phase is distinct and invokes process flows that are associated with that phase. For instance, a loan, which may have a lifecycle of 30 years has multiple phases, each distinct with it's own unique processes. These phases may be Loan Request (Application/Origination), Servicing and Archival. Advanced state management enables designers to easily model these types of processes without significant coding.

FileNet P8 Integration Models

The FileNet P8 architecture was designed to unobtrusively support a company's existing systems and processes, so the ECM and BPM capabilities could be leveraged with minimal disruption. To accommodate a broad set of integration scenarios, FileNet P8 provides three integration models.

- Direct-Integration model
- Event-driven Integration model
- Content-driven Integration model

The following provides a conceptual view of the various integration models supported by the FileNet P8 architecture.

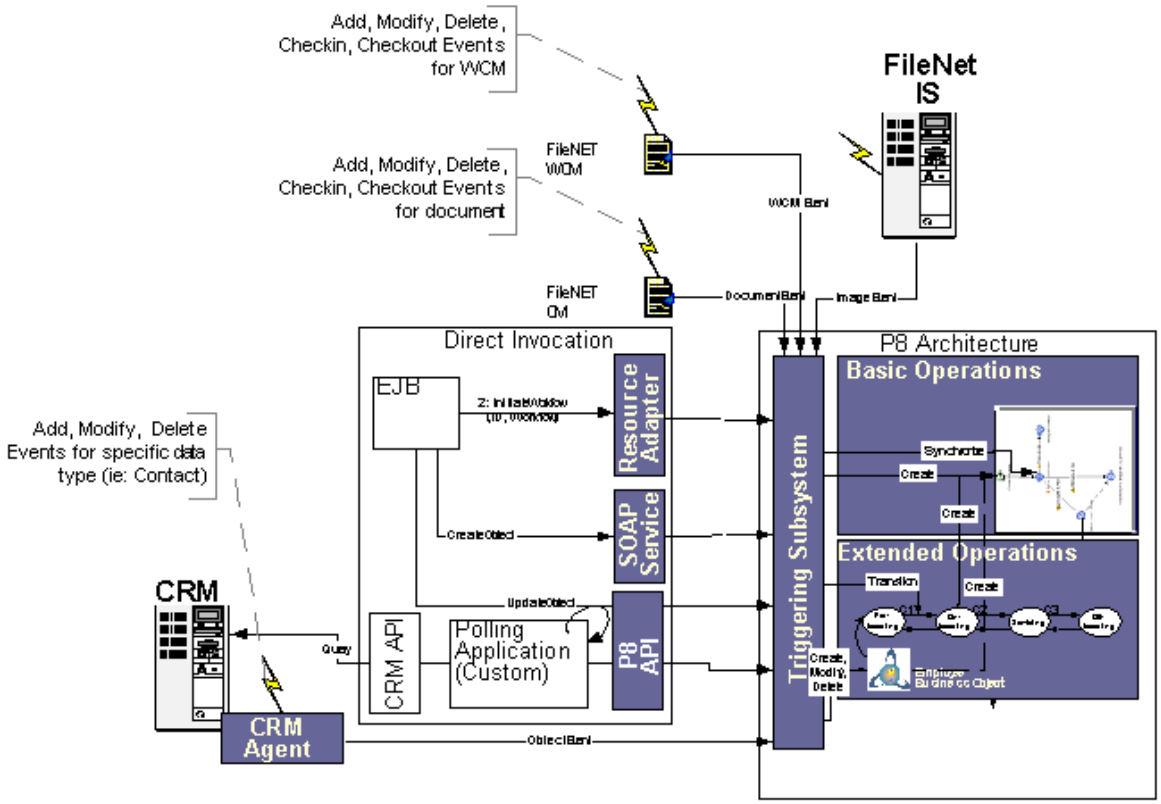


Figure 1: FileNet P8 integration model

External applications can initiate and synchronize with FileNet P8 workflows via the direct integration model or through the event driven model. These models also enable external applications to access FileNet P8 content. The third integration model, the Content-driven integration model, enables the triggering of both FileNet P8 processes and external processes on key content events.

Support for these integration models revolve around capabilities introduced with the business object. The Business Object provides extensive and flexible object/process triggering and invocation mechanisms.

Triggering describes the mechanism through which systems can initiate and interact with FileNet P8 processes and objects (call model, event triggered). ***Invocation*** enables the system designer to map both external system events and intrinsic content **events** (modify, delete, add, checkin/checkout) to workflows and automations.

FileNet P8 processes and objects can be created/triggered by external system events, document events, image events and web content (WCM) events.

External system integration can be broken down into two models: the ***Direct Invocation (call-based) model*** and the ***Event Triggered (non-intrusive) model.***

Model 1: Direct Invocation Model

The ***Direct-Invocation model*** supports applications where the programmer has full access to the triggering application source code or a mechanism for hooking into the triggering application (either polling application using application API or stub code). In this model, the triggering application will communicate with FileNet P8 to create an object or workflow.

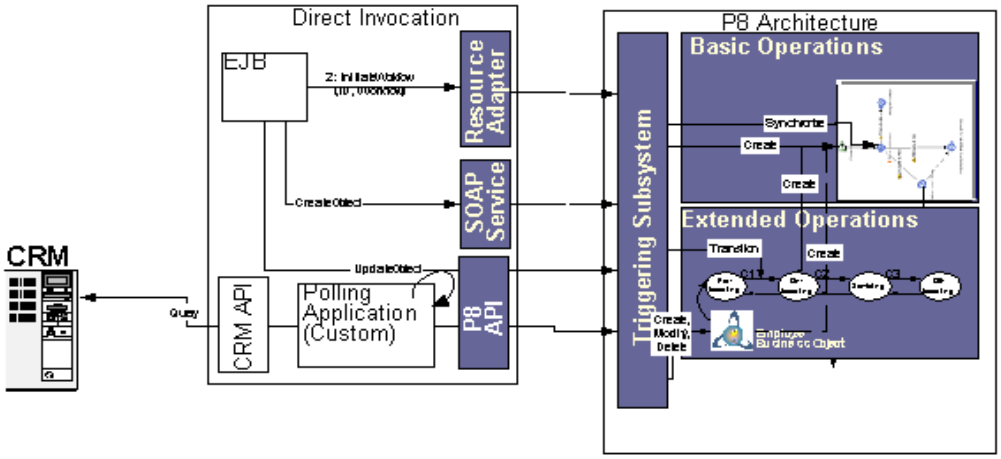


Figure 2: External Application integration via Direct-invocation model

The triggering application can utilize any of the following mechanisms for directly communicating with FileNet P8:

- FileNet JAVA API
- SOAP
- Java Resource Adapter

- Java Message Service (JMS)
- Batch operations (file based)

Model 2: Event-Triggered Model

The ***Event Triggered (Non-intrusive) model*** provides a mechanism to seamlessly plug into applications where source code is unavailable. In this model, agents can plug into these applications and trigger the creation of objects and workflows upon key system events (ie: update of customer information). For example, if the calling system does not easily support the Direct Invocation model (through Java), the Event Triggered model is a low risk, rapidly deployed integration mechanism, with minimal coding requirements.

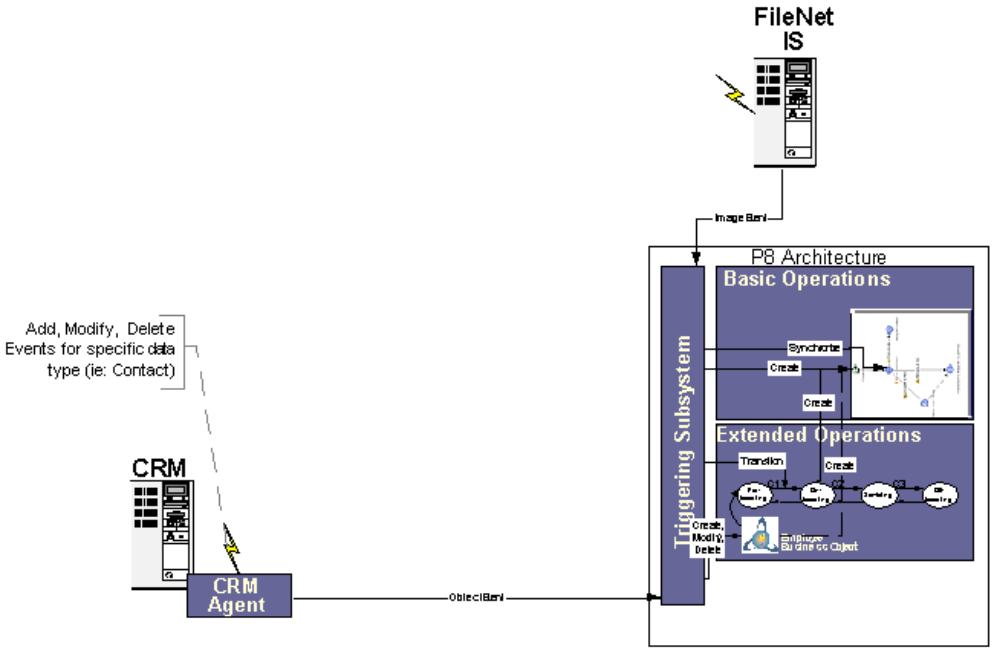


Figure 3: External Application integration via Event-triggered integration model

The event-triggered model enables rapid integration with the following systems (this is a sample list – please check for availability)

- Enterprise Resource Planning Applications (for instance, SAP)
- Human Resources Management Application (for instance, Peoplesoft)
- Customer Relationship Management Applications (for instance, Clarify)
- eCommerce Applications (for instance, Trilogy and Broadvision)

Model 3: Content Driven Integration Model

FileNet P8 provides tight coupling of content and processes, so events related to FileNet P8 based content (Standard content and WCM content) can intrinsically initiate processes. This capability is important when you consider that content, such as the creation of a loan origination document can initiate processes to review and approve the document. An update to the same document can initiate a different process, which may update an external portal with the changed document.

Another scenario that is commonly seen with business processes is one in which a process is awaiting the publication of a set of documents from a different team members. A process can be synchronized to await the receipt of a set of FileNet P8 content events (one for each document being published) before it continues to the next step.

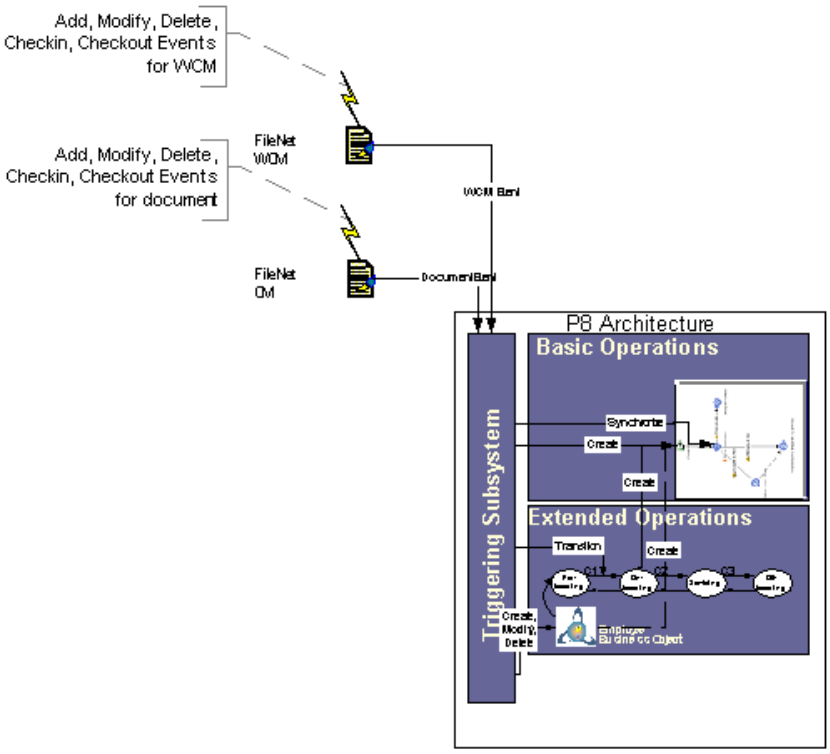


Figure 4: Content Events trigger processes

Process Invocation

Business Objects provide enhanced invocation of processes. Invocation describes the action that occurs due to a key event on an object. The business object can subscribe to workflows and automations (as stated in the previous section). Workflows and automations can also be configured to execute based on key **object events** (Creation, Modification, Deletion, Checkin, Checkout.) This expands the previous eProcess capability of allowing workflows to be executed based on two **document events** (Creation, Checkin). FileNet P8 subscriptions enable the business analyst to map key object events to workflows and automations. This capability centralizes the management of workflows and associates object events to underlying workflows. It reduces code dependencies within calling applications, leading to a reduction of maintenance cost and development time.

Enhanced Process modeling

FileNet P8 Business Object provides advanced process modeling capabilities; including support for object oriented process development and advanced state management.

Object Oriented Process Development

Business Objects extend process design to support an object-oriented development methodology. Business objects support inheritance including both data inheritance and process inheritance.

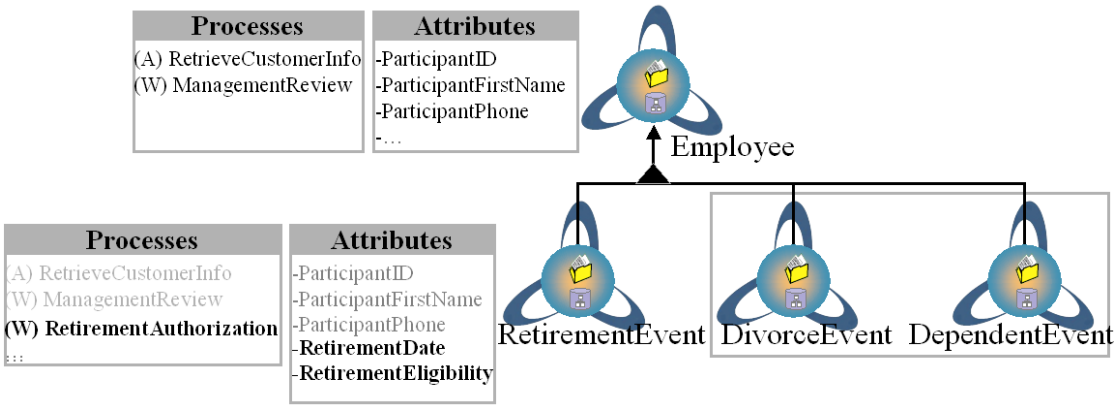


Figure 5: Object Inheritance model for Human Resources Processes

This drawing highlights the LifeEvent object hierarchy, and shows the relationship between the RetirementEvent object type and the LifeEvent object type.

Object subscriptions enable ***objects to subscribe to workflows and flow-through automations***. These ***subscriptions can be mapped to key object events***, such as

creation, update, delete, checkin, checkout. Thus, the creation of a LifeEvent object (through creation of any child object types) would trigger the automation, RetrieveCustomerInfo and the workflow, ManagementReview.

Without the business object, the workflow designer would be responsible for coding this type of behavior inside the workflow through complex condition statements and often requiring duplicate code. As the size of the system grows, this development approach becomes very unmanageable.

The Business Object's support for object oriented process development results in a higher level of reuse, higher design scalability and simplified maintenance. The key benefit is reduced development time that translates directly to reduced time to market.

Advanced State Management

Business Objects provide ***advanced state management*** for complex object types that sequence through various states throughout the object's life. The Business Object ***lifecycle*** capability allows states to be defined, definition of conditions (triggering events) that cause state changes, and actions (workflow launch) that occur upon transition into a new state.

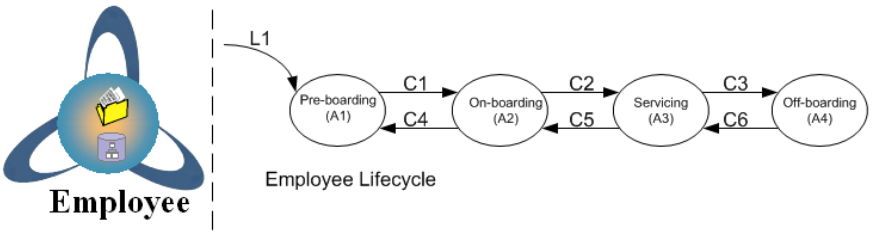


Figure 6: Business Object Lifecycle

This diagram shows the Employee Business Object and it's associated LifeCycle. The employee progresses through various states while employed, and these can be modeled as the states pre-boarding, onboarding, servicing and off-boarding. A request by the hiring manager to initiate the hiring of this person (L1) could be initiated through a web site. This would enter the first state of the employee lifecycle (Pre-boarding). Upon entry to this state, a series of actions may be triggered (A1), such as initiating automations to update an HR management system like Peoplesoft with employee information and launching of a workflow to collaborate on an offer letter. The object would remain in this state for an indefinite period of time, until condition C1 is reached. C1 could be an event triggered with the setting of a flag within the employee record in Peoplesoft, transitioning the state to Onboarding, initiating all of the workflows and automations associated with entry into this state.

SUMMARY

Business Objects are a core concept in FileNet P8's ECM architecture. They provide an effective mechanism to rapidly overlay FileNet P8's Process and Content capabilities over a company's existing business processes without disruption of these processes and systems. FileNet P8's Process Management capabilities can then be used to automate, manage and streamline business processes. FileNet P8's Content can actively participate in processes in a scalable, enterprise-wide paradigm.

The Business Object's support for Object Oriented Process Design and the Advanced State Management reduce the complexity of building, modeling and deploying business processes. It provides enhanced reuse of both processes and content, which will enable processes to be deployed quicker and with reduced cost.

These capabilities will enable corporations to rapidly respond to changing business conditions and allow companies to achieve a higher level of business agility.

About FileNet

FileNet Corporation (NASDAQ: FILE) helps organizations make better decisions by managing the content and processes that drive their business. FileNet's ECM solutions allow customers to build and sustain competitive advantage by managing content throughout their organization, automating and streamlining their business processes, and providing the full-spectrum of connectivity needed to simplify their critical and everyday decision-making.

Since our founding in 1982, more than 3,700 organizations, including 80 of the Fortune 100, have come to depend on FileNet solutions for help in managing their mission-critical content and processes.

Headquartered in Costa Mesa, California, FileNet markets its innovative solutions in more than 90 countries through its own global sales, professional services and support organizations, as well as through its ValueNet® Partner network of system integrators, value-added resellers, and application developers.

APPENDIX G

Panagon Forms White Paper

White Paper

PANAGON™ eFORMS WHITE PAPER

April 2002

FileNET®
The Substance Behind eBusiness™

...achieve a faster return on investment (ROI).

The Key to Faster ROI

Imagine having hundreds of dissimilar, often paper-based forms, hundreds of manual processes, and hundreds or thousands of employees, but no single Business Process Management (BPM) solution integrating these forms into the dynamic enterprise environment. Often compounded by mergers and acquisitions, this scenario represents one of the largest opportunities for companies with key business requirements of lowered costs, improved efficiencies, and better customer service to achieve a faster return on investment (ROI).

To the participant in a business process, there is no distinction between the user interface and the entire business process behind the user interface. Companies recognizing this significance to customer, partner, and employee participants are turning to out-of-the-box electronic forms applications, or eForms, for an intuitive and visually rich user experience into business processes. The integration of eForms into the dynamic enterprise BPM environment represents the most recent breakthrough in eForms technology, following the rise in popularity of Internet-enabled eForms used for "fill and print" or basic workflow routing implementations that were not integrated into business processes. These implementations provided faster access to the form, but failed to significantly reduce the remainder of the form processing time.

This white paper details the business value proposition and architectural implementation of how Panagon eForms develops and deploys enterprise BPM solutions for a faster ROI.

Gartner, Inc. estimates that businesses spend more than $360 billion per year to process forms.

Business Challenge

For the past five years, IT budgets have been earmarked for eBusiness initiatives. However in today's challenging economy, there is urgency for companies to prove eBusiness value through fast, successful implementations.

A successful eBusiness initiative can be measured in a number of ways. Was the project completed? Was it completed on time? Does it function the way it was intended? Did it realize the predicted ROI in the expected timeframe? When the answers to these questions are unsatisfactory, an identified bottleneck to the eBusiness implementation often occurs in the development of user-facing applications. Precious IT and Web development resources are frequently necessary to build and maintain the front-end applications, our user interface to business process management solutions.

Industry analysts and government officials alike have recognized the pressing need to solve the paperwork problem. Gartner estimates that businesses spend more than $360 billion per year to process forms. The U.S. federal government has mandated that paperwork be eliminated from public service by 2003 in its Government Paperwork Elimination Act. As this time approaches, eForms are more important than ever in solving the paperwork challenge.

The key criteria that have emerged from eForms deployment are:

- eForms must integrate seamlessly with a company's extended enterprise business processes
- eForms must incorporate digital signatures in a flexible and secure manner
- eForms must integrate seamlessly with a company's IT infrastructure using a robust toolset
- The data displayed in eForms must be stored separately from the eForm presentation to support easy access by partners or other information systems
- Knowledge workers must be able to access eForms on the Web
- Administration of eForms must be centralized
- eForms must be easy-to-use for increased user acceptance
- eForms must employ form-filling features that improve user productivity and increase accuracy

Panagon eForms brings together the tools required to quickly and successfully deploy integrated eForms solutions across the enterprise.

Panagon eForms leverages 17 years of successful software development to address these needs. Together with Panagon eProcess and Content Services, Panagon eForms represents a best-of-class eBusiness solution. Since its inception, this offering was designed to help companies leverage large quantities of corporate data to improve operations.

Solution Description

Panagon eForms brings together the tools required to quickly and successfully deploy the integrated eForms solution across the enterprise.

Created as a fundamental product component, Panagon eForms extends the functionality of eProcess Services. eForms become the front end, or user interface, for business processes automated with Panagon eProcess Services.

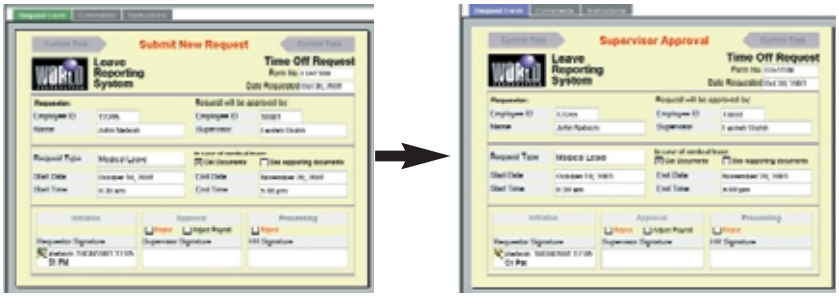


FIGURE 1 — Panagon eForms is the user interface to this Leave Request process. Panagon eProcess routes the Leave Request eForm from the employee to his/her supervisor. After the supervisor has digitally signed the form, eProcess routes it to an HR sub process along with the associated attachments.

Panagon eForms includes key features such as:

- Easy-to-use designer tool – allows anyone, including business users, to design an eForm
- Robust drawing tool set enables forms designers to create eForms that can be designed to appear identical to paper-based forms
- eForm automation of calculations, database lookups, error checking, and auto-entry of validated data improve efficiencies in form filling
- XML-based templates and data allow information to be shared with back-office systems
- Multiple digital signature support (Entrust/PKI™, Microsoft® CryptoAPI, and I-Sign™ Panagon) increases eForm security and authentication
- Same look and feel as Panagon eProcess Services to provide a unified user experience
- Leverages existing Enterprise Content Management repository, eliminating the need for a dedicated eForms server

- Content and property exchange with Panagon eProcess allows form content to drive conditional routing and automatically launch business processes
- Seamless integration with eProcess features include document life cycle management, process control and standardization, process timing, reusable process submaps, complex workflow support, and event triggers (Wait for Condition) for inter-enterprise process communication

Use Cases

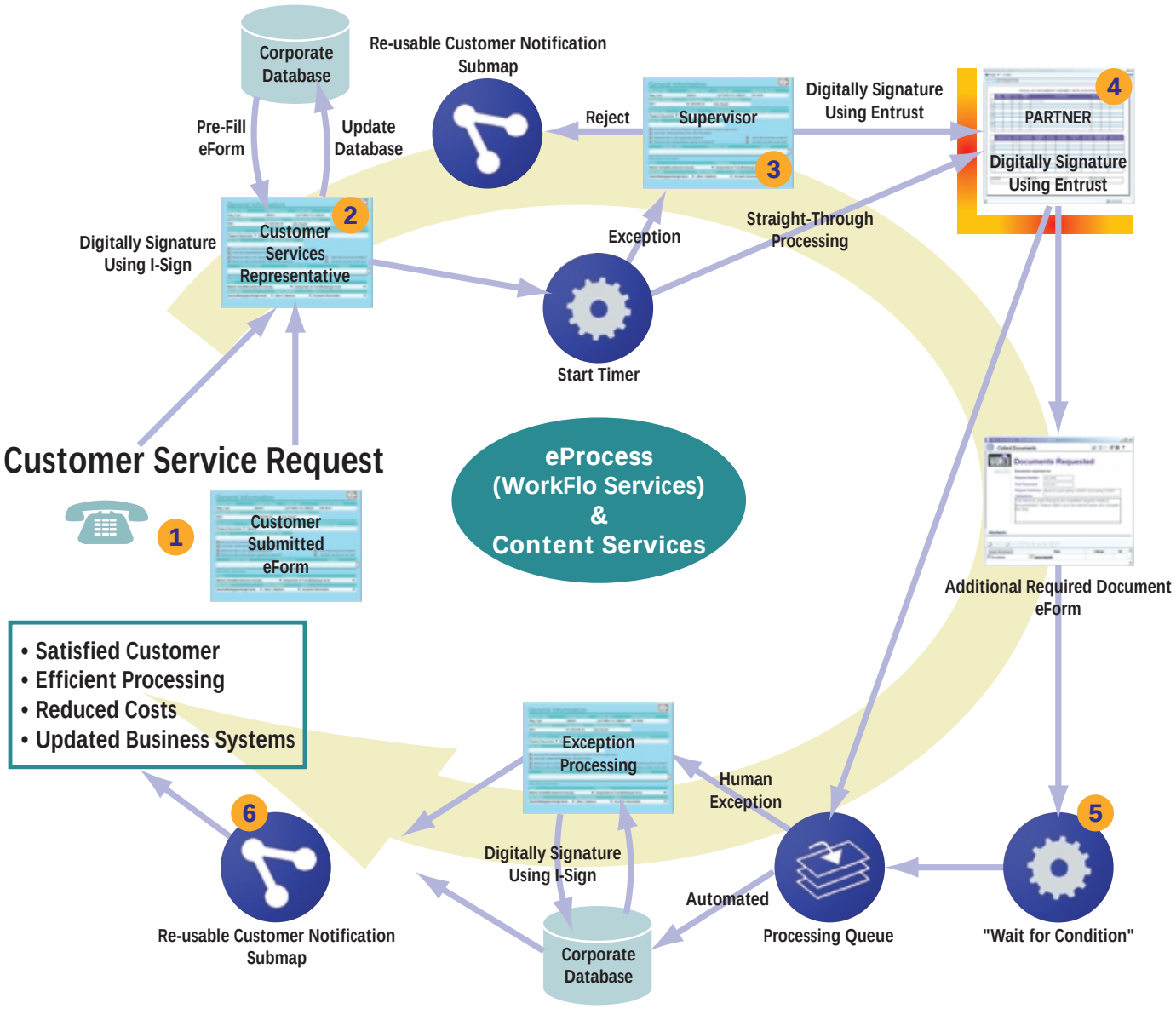


FIGURE 2 — Customer Service Request

CUSTOMER SERVICE REQUEST

As seen in Figure 2, a customer service request may start with a phone call or by filling out an eForm on the Web (Step 1). The Customer Service Representative (CSR) receives an SMTP-based email notification from eProcess that serves up the XML eForm as well as the XML pre-filled data with the employee's information, employee ID as well as the customer's relevant background information, such as whether or not a Premium Customer Service contract exists. After filling out the first eForm, the CSR digitally signs the eForm using the Panagon eForms I-Sign digital signature technology (Step 2). The eForm then updates the corporate database using the integrated ODBC functionality. Next, a Timer is started to enable escalation if a process step is not completed within a specified timeframe. If required, a supervisor reviews the eForm (Step 3) and signs the approval cell of the form, but this time uses a 3rd party digital signature technology, such as Entrust, for high security, high value communication outside the firewall. After the partner completes the eForm (Step 4), an additional process may be automatically launched by eProcess to wait for additional documentation to arrive from the partner. This unique "Wait for Condition" technology (Step 5) enables the main process to synchronize with another process and continue when the launched process data matches a specified condition. Once complete, a reusable sub map is called to notify the customer or assign the notification to a CSR for phone reply (Step 6). The end result is a satisfied customer through fast response time and efficient processing, resulting in lower costs and automatically updated business systems.

CONSTITUENT-CENTRIC SERVICE

As citizens become more and more computer-savvy, expectations have risen for "anytime anywhere" service from government. However, the many paper-based forms that drive processes hinder governments from providing constituents with timely responses.

Panagon eForms and eProcess provide a single, consistent approach for forms and the associated processes to be automated and offered to citizens as a digital government service. Rather than having multiple departments, such as utilities regulations, health services, motor vehicles, etc. with their own independent forms and processing technologies, governments can provide a single, scalable approach for all government processing. For example, from a single government Web portal, a constituent can fill in a utilities complaint form online when a telecommunications company begins "slamming" (switching to a new carrier without consumer authorization).

Panagon eForms provides companies with a simple way to digitize paper-based forms

Once submitted, the request is automatically forwarded to the appropriate government office for review and automatically routed to legal using the built-in digital signature authentication. Furthermore, the process might "wait" if the constituent wishes to mail in supporting documentation, leveraging the "Wait for Condition" capability within eProcess. Once this process is complete, a sub map (reused for every government department) would update internal records and notify the constituent of the resolution. Significant amounts of time and money are saved, and the constituent receives superior service.

TOTAL QUALITY MANAGEMENT

Companies undertaking initiatives to gain and maintain ISO compliance require documented quality assurance processes. On the manufacturing shop floor, properly completing quality control forms, such as work instructions, quality checklists, and non-conformance reports, are critical to ISO standards adherence.

Panagon eForms provides companies with a simple way to digitize paper-based forms. With Panagon eForms, shop floor workers can access non-conformance forms already populated with key information. Since the form cannot be digitally signed and sent until all required fields are filled, the incidence of keying error or incomplete forms is virtually eliminated. The completed non-conformance form can automatically be forwarded to the next step in the workflow with Panagon eProcess Services based on how the eForm was filled in. Not only is the transfer time from one step in the process to the next virtually eliminated, but the fact that digitally signed eForms are stored in the Panagon Content Services repository ensures simple and clear proof of conformance support.

CORPORATE ADMINISTRATION/HUMAN RESOURCES

Effective use of forms makes corporate administration run smoothly. Issues arise when employees cannot locate current revisions when forms are incomplete or when processes are not clear. Bottlenecks in Human Resources forms processing prevents employees from spending more time on critical line of business applications and customer service.

Panagon eForms increases the efficiency and effectiveness of business processes. In the case of expense claims, an employee can access the eForm at an Intranet site; fill and sign it; and have it electronically forwarded to his supervisor. Once the supervisor has digitally signed the eForm, it is automatically forwarded to the accounting department. Processes that could normally take weeks to complete are reduced to days or even hours.

From end-users to process owners to network administrators, each stakeholder has different needs. Panagon eForms was developed with these needs in mind

Stakeholders

"Buy-in" from many stakeholders within a company is critical for a successful implementation of technology. From end users to forms designers to network administrators, each stakeholder has different needs. Panagon eForms was developed with these needs in mind.

EFORMS USERS

Built on familiar Windows® and Web technologies, the Panagon eForms user interface is intuitive and easy to learn. In many organizations, eForms are designed to have the identical appearance of high-fidelity paper-based forms, encouraging user acceptance. Panagon eForms provides unsurpassed usability and integrates transparently with Panagon eProcess Services.

EFORMS DESIGNERS AND BUSINESS PROCESS OWNERS

Traditionally, developing eForms for the Web diverted IT focus from critical issues to non-reusable, form development tools such as DHTML, InterDEV and others. The time and high costs associated with allocating Web development resources can limit the number of processes that are considered for automation. The Panagon eForms Designer contains a simple user interface that enables business analysts to design eForms themselves, allowing developers to concentrate on advanced integration requirements.

NETWORK ADMINISTRATORS

Panagon eForms is managed centrally through a single Web portal based on Panagon eContent Services. As eForms are designed, they are "checked in" or deployed to the repository as eForm templates. If the eForms change or are phased out, only one instance of the eForm requires revision since the eForm template and its associated content and digital signatures are stored in a single repository. This repository, Panagon Content Services, routes single files to complete the steps in a process. Additionally, Panagon eForms is built on Internet standards, making it easy to function within virtually any enterprise infrastructure.

EPROCESS CHAMPIONS

Together, Panagon eProcess and Panagon eForms offer enterprise-class scalability, security, and reliability. From the costs saved by reducing data entry errors to increasing response time to form submission, Panagon eForms provides a compelling opportunity for fast ROI. Its open architecture also ensures interoperability with current and future technologies.

Components

The Panagon eForms product adopts the eForms life cycle: Design, followed by Deployment throughout the entire user community, and finally Processing (and future archiving and disposal). The following world-class components support the Panagon eForms life cycle.

Panagon eForm Components	Panagon eProcess and Content Services Components
Design	
Designer Creation of the eForm layout including user-interface labels, data fields, digital signatures, and "intelligence" features . Forms and data are XML	eProcess Designer Used to create a workflow definition, a process map that specifies the steps, resources, and routing logic needed to complete a business process.
Deploy	
Designer Deploys eForm templates from the Designer to Panagon Content Services in two easy steps.	Content Services Library Acts as the repository for eForm templates, partially completed forms, and completed forms to which the eForms user has access.
Process	
Step Processor The eForm becomes an intuitive user interface for each step of the process, also called the Step Processor. eRouting Slip eRouting Slips differ from eForms in that they are not stored in Content Services and only display information from the eProcess work item. They are useful in applications using images in workflows not requiring Digital Signature authentication and when Content Services is not required.	eProcess Services The Java user interface that ties eForms to the workflow engine. It includes the eProcess Designer, configuration console, administrator, tracker, personal work manager, and more. WorkFlo Services The workflow engine performing processing that lies behind the eProcess Services Java user interface.

Architecture

Panagon eForms integrates seamlessly into the existing Panagon Content Services and eProcess Services architecture. As shown in Figure 3 below, the eForm template created using the Panagon eForms Designer resides on the content server. To allow for sharing of eForm content with external partners, the eForm template is stored separately from the XML eForm data. These templates contain the eForm presentation as well as the underlying logic for such operations as database lookups and validation. When a participant accesses work, typically through email notification, the relevant eForm template is opened within the client Web browser. Panagon eForms automatically prefills the appropriate fields and displays the eForm instance to the user. This is done using the XML parser and renderer components of Panagon eForms, installed on the eProcess Web server. The user inputs additional data and may select from drop down lists, set radio buttons, etc. During this process, the eForm logic notifies the user of errors or performs advanced calculations based on input from previous cells or from external data access. Since the logic resides on the client Web browser, minimal client-server communication is required that would otherwise cause delays during calculation and validation of the eForm.

The Panagon eForms solution is highly flexible to accommodate input from many sources. Throughout the lifecycle of the eForm, the eProcess work item is one source of data exchange to and from the eForm. Any additions or modifications to the eForm by the user will update the eProcess work item upon closing of the eForm. Furthermore, data not contained in the work item, such as digital signature information, is accessed from the content server and is also displayed to the eForm at each step in the process. A ten-step workflow process, for example, has the flexibility to access and modify information from the work item, the content server, or an external database, as well as utilizing a different eForm presentation at each step in the process. Some steps in the process may require a presentation including digital signatures whereas the next step may not contain digital signatures, but could have confidential information not displayed at other steps in the process.

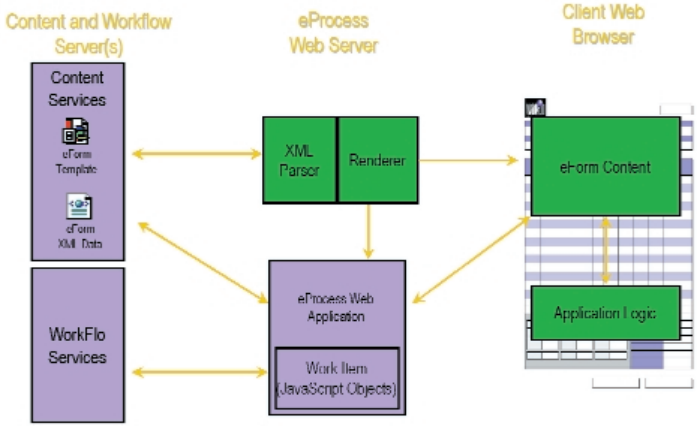


FIGURE 3 — For a rapid eForm display, the Renderer converts only the necessary XML information into the eForm with dynamically generated client-side scripting. Note the eForm's ability to access information from both the Content Services repository as well as the eProcess/WorkFlo Services work item.

Digital Signature Authentication

Businesses wishing to stay globally competitive are either implementing digital signature technology or ensuring that current BPM investments have flexible support for near-term digital signature requirements. In June of 2000, the "Electronic Signatures in Global and National Commerce Act" also known as E-SIGN, took effect in the U.S., enabling digital signatures to be as legally binding as traditional signatures. Businesses throughout the world are also following this trend to eliminate the transfer time of sending and receiving documents in the mail for signatures, saving days or weeks in the financial, insurance, manufacturing and government services processes.

Signing electronically reduces the need to print forms and offers more security than paper signatures in that the authentication remains inside a secure electronic content repository, as opposed to a paper file cabinet.

Digital signature technology employs sophisticated encryption algorithms to provide reliable signer identification and fail-safe tamper detection. This means that once someone has signed data electronically, the resulting digital signature can be used to:

- Verify the identity of the person who signed the data
- Determine whether or not the data has changed since it was signed

While the digital signature is stored with the signed data, the data itself is not altered in any way. Once a digital signature is created, you can easily verify its validity. The verification process involves re-creating parts of the digital signature using current data then comparing the results with the original signature. If they are not equal either the signed data or the digital signature itself has been changed or tampered with.

Panagon eForms includes Digital Signature authentication in the out-of-the-box product as well as the flexibility of the design to support different requirements, as well as the flexibility to support high-end security needs. There are several options depending on the requirements shown on the next page:

Method	Description	Implementation
I-Sign	The most commonly used method, I-Sign uses an MD5 message digest algorithm recommended by the W3C for encryption and the existing Content Services user ID and password for authentication.	Included as part of Panagon eForms. No extra cost. Easy to set-up. No additional administration.
Entrust, VeriSign	Purchased for high security environments.	Integration hooks included with Panagon eForms. Additional licensing and software is purchased from the respective vendor.
Microsoft CryptoAPI	To authorize, a digital signature certificate issued by a Microsoft Cryptographic Service Provider that supports the CryptoAPI standard (e.g. VeriSign), is required. The certificate is installed in the browser.	The Microsoft CryptoAPI is a complete set of tools. Microsoft distributes the Cryptographic Service Provider Developer's Kit (CSPDK) to authorized recipients located in the U.S. and Canada, as well as licensed entities outside the U.S.

Typically, before users can sign electronically, it is necessary that they obtain special files that act as their "electronic identity" for signing purposes. These files require additional administration overhead by an organization's security or administrative manager. With I-Sign, no additional overhead is required since it leverages the Content Services authentication that may be used for other unstructured content.

This approach is commonly used for intranet applications since it has the benefit of no additional cost or implementation effort. For high value (such as expensive approvals) or high security applications, the Public Key Infrastructure (PKI) integration in Panagon eForms is used in parallel. PKI involves the use of two cryptographic keys, one private and one public. eForms encrypted with one key can only be decrypted with the other key. The publicly available key, distributed through the Web browser, is embedded in a certificate with personal details about the user. These certificates are issued by certificate authorities, such as VeriSign and Entrust, and provide the validation function by linking a public key to a user in order to identify individuals and organizations. These PKI solutions are wells suited for extranet and Internet applications, allowing Panagon eForms to suit the specific needs of intranet, extranet, and Internet requirements.

To create Digital Signatures on a form, the user simply places one or more signature cells on the form using Panagon eForms Designer. Each signature cell can sign the entire form or specified portions of the form. eForms users can easily sign completed forms and check the validity of signatures.

Panagon eForms Designer supports the use of digital signature technology both for signing completed forms and for authorizing templates. Authorizing a template adds an additional level of security to forms that are completed and signed electronically.

The Panagon eForms Advantage

Panagon eForms, combined with FileNET's eProcess technology, is a compelling solution that offers an easily justified purchase. It leverages existing FileNET investments and makes your life easier by reducing the complexity of disparate forms and custom-developed process user interfaces.

Speeds Time-to-Deployment

Panagon eForms allows eProcess applications to be deployed faster for an immediate ROI. Months of time have been removed from enterprise implementations that required integration of dissimilar content repositories, workflows, and paper or eForm implementations. This integration comes out the box with the combined solution.

Increases Operational Efficiencies Across the Extended Enterprise

Enterprise-class scalability allows for the automation and management of thousands of repeatable processes.

Increases User Productivity and Accuracy

The user interface for integrating eForms into eProcess applications is highly customizable enabling an extremely intuitive experience for workflow participants. User productivity and accuracy increase while training costs are reduced.

Flexible eForm Presentation

Satisfy the unique needs of every process type. Some eForms display workflow information only, other eForms contain attached content and digital signatures and in many instances, both types of data are displayed in the eForm.

Encourages User Acceptance

eForm and workflow designers can replicate existing paper processes for a familiar user experience. Panagon eForms' seamless integration with Panagon eProcess and Content Services eliminate the need for users to learn two separate systems.

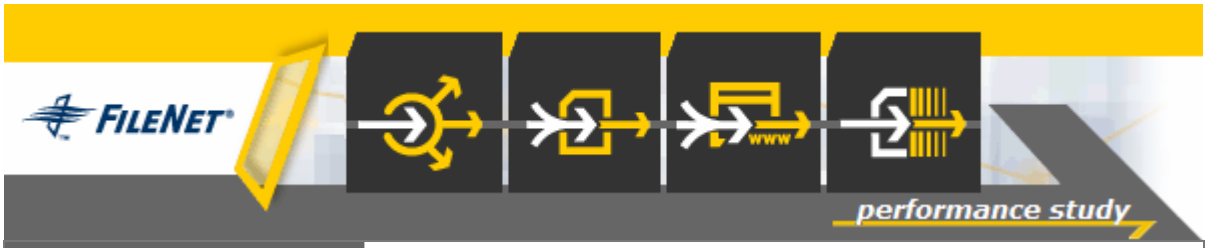
UNIQUE CAPABILITIES

Why is FileNET unique in it's ability to deliver these benefits?

1. No other solution offers a comparable level of eForms integrated into an enterprise scalable Business Process Management solution
   - Most eForm solutions are effectively "standalone" eForm offerings with limited routing (person to person), but lacking workflow depth and breadth. Critical workflow features, such as inter-process communication (Wait for Condition), re-usable submaps, milestones, and high volume throughput to name a few, must be a part of any eForms process management solution.
2. The only FileNET supported enterprise eForms/eProcess environment
   - eForms and their associated enterprise processes must be very tightly linked and therefore require a tightly linked development and customer support solution. Panagon eForms and eProcess are combined solutions with a single point of support contact.
3. No additional eForm Server (repository)
   - For customers using Content Services as a part of their Enterprise Content Management solution, there is no need for an extra cost "eForm Server" license or the associated hardware and maintenance costs.
4. Support of Internet, extranet, and intranet Digital Signature Requirements
   - In addition to the integrated VeriSign and Entrust solutions targeting extranet and Internet applications, Panagon eForms customers see significant benefit in the I-Sign Digital Signature solution, which is particularly well suited for intranet applications. Frequently, customers need an alternative to the higher cost Digital Signature solutions for their own employees in conjunction with VeriSign and Entrust for high security, high value solutions. I-Sign comes at no additional cost with Panagon eForms.

APPENDIX H

FileNet ECM Performance Study



FileNet Content Manager Performance Study

Document Search and Retrieval

Summary

This document details performance tests of Enterprise Content Management (ECM) functions of the FileNet Content Manager product. The tests consisted of CPU and response time characterizations of logon, browse, search, and document retrieval operations. All characterizations were done for test runs each of which comprised one type of transaction only.

As expected, response times for most operations were less than one second, an exception being the browse of a folder with 100 contained documents; that operation generated response times of just over one second.

A FileNet Engineering Performance Study

Draft, March 27, 2003

Table of Contents

Table of Figures

1 Introduction

This document discusses the results of performance tests of the FileNet CONTENT MANAGER product, with particular focus on the Enterprise Content Management (ECM) functionality. End-to-end response times for various operations are given, and, together with CPU utilization statistics for different components, are used to characterize the system.

The intended audience for this document, and for the remainder of the FileNet Content Manager performance white paper series, includes FileNet Engineering and Marketing, and, following appropriate review, other readers interested in the performance characteristics of the FileNet Content Manager products.

2 Test Methodology

FileNet conducted this study in order to evaluate the performance of the FileNet Content Manager product. This part of the study is not an attempt to completely characterize the scalability of the product; rather, the idea was for it to be an initial set of tests detailing response times and CPU costs for the major content management operations. Future studies will address issues of large-scale performance under high loads, as well as authoring and BPM functionality.

The terms "throughput" and "applied workload" are both seen throughout the discussions here, and both have precise meaning. As mentioned above, throughput refers to the number of transactions completed per unit time, and is a measured quantity of a running system. Applied workload refers to the throughput that would be measured if the response time to the client were zero; in other words, it is the amount of work theoretically being applied to the system. In general, as response times rise, throughput drops, even if the applied workload remains the same.

Throughput measures are reported in terms of successfully-completed compound transactions per minute. Workload was generated using virtual clients, each of which performed, on average, one such transaction per minute. Specifically, each client drew request delay times from a Poisson distribution with a mean of one minute.

A third-party testing and simulation tool, RadView's WebLoad, was used to drive all performance activities. Scripts, which were coded in JavaScript and executed by the tool, controlled the virtual clients and determined the

execution of operations in the test. For more details on the test process, see the appendix.

2.1 Description of the Test Data

A database of synthetic test documents was created. This was done using a custom population tool which added the documents via the Content Engine API. That tool created 250,000 simple text files each of sizes 3K, 10K, 30K, 100K, and 2,000 files of size 1M bytes. Those documents were filed into folders in two different trees, one with folders containing 10 documents and one with folders containing 100 documents. A total of 1,002,000 documents were created. See the appendix for more information.

2.2 Description of the Workload

The workloads for the tests described in this study were designed to test the content management functions available in the FileNet Content Manager product, in such a way as to characterize the individual transaction cost and response times. Because of this, for this part of the study sequence, no "mixed" tests (tests including a variety of transaction types) were performed. Rather, each test consisted of a set of identical transactions in which each virtual client varied only in the amount of time it waited before commencing with the next operation. Each test thus consisted entirely of logon/logoff pairs, browses, document retrievals, or searches. Such an approach allows for the estimation of CPU costs for various mixes of operations by the explicit use of the calculated values from the individual numbers determined here.

2.2.1 Logon

Using a randomly-selected username from the set of names known to be in the system, a logoff operation was performed, followed by a logon. No other activity was performed. The operations were executed in this order because all users were logged on at the beginning, and their sessions were kept active for the duration of the test.

2.2.2 Browse

All folders in the system were populated with a known number of documents. For this study, they contained either 10 or 100 documents, and the corresponding tests will be referred to as "browse 10" and "browse 100", respectively. These tests opened a bookmarked folder containing a

particular number of documents, and the timing statistics were recorded. This folder was chosen at random for each browse from a set of folders with the given number of contained documents.

2.2.3 Search

Every document in the system included a user-defined property used for a unique document ID. The search test consisted of starting at a randomly-chosen document ID within the range of known IDs and performing a query resulting in a returned list of the next ten document IDs. Other performance studies in this series will test the search functionality more extensively, including characterizing the performance of returning different-sized result sets.

2.2.4 Document Retrieval

Documents for retrieval were chosen by starting with the set of IDs corresponding to appropriately-sized documents. Within that set, document IDs were chosen randomly, and the corresponding documents were retrieved.

3 System Configuration

The system under test comprised separate machines for the different FileNet Content Manager components, including a Domain Controller (DC), Content Engine (CE), two Application Engines (AEs), and SQL Server. In addition, there were machines to facilitate the test simulations, including a WebLoad Console to control the running of the tests, and a set of Load Generators, controlled by the Console, to simulate user load on the system. These components are described as follows:

- **Domain Controller**
  DELL PowerEdge 2450
  Dual Pentium III 1 GHz processors
  512 MB RAM
  Windows 2000 Advanced Server SP3
- **Content Engine**
  DELL PowerEdge 2650
  Dual Pentium 4 Xeon 2.4 GHz processors
  2 GB RAM

Windows 2000 Advanced Server SP3
FileNet CONTENT MANAGER 2.0 build kl140.117

- **Application Engines (2)**
  DELL PowerEdge 2650
  Dual Pentium 4 Xeon 2.4 GHz processors
  2 GB RAM
  Windows 2000 Advanced Server SP3
  BEA WebLogic Server 7.0
  FileNet CONTENT MANAGER 2.0 build per 100.178
- **SQL Server**
  DELL PowerEdge 2450
  Dual Pentium III 1 GHz processors
  1 GB RAM
  Windows 2000 Advanced Server SP3
  Microsoft SQLServer 2000
- **WebLoad Console**
  DELL PowerEdge 2450
  Dual Pentium III 1 GHz processors
  512 MB RAM
  Windows 2000 Advanced Server SP3
  RadView WebLoad 5.0
- **WebLoad Load Generators**
  DELL PowerEdge 2450
  Dual Pentium III 1 GHz processors
  512 MB RAM
  Windows 2000 Advanced Server SP3

All machines were connected through a physically-isolated 1-gigabit Ethernet switch.

4 Results

This section presents selected results from the tests. CPU utilization and associated throughput are used to calculate CPU cost, and those results are presented for a representative sample of operations. Response times for a sample of operations are also presented.

4.1 CPU Utilization

CPU utilization was low on all components at all load levels in the test. In terms of raw utilization, the most expensive operation was the browse with 100 contained items, in which the CE was utilized at 44% under a 200-user workload. A representative sample of utilization is given in the following table. For the entire data set, see the appendix. It should be noted that throughout the discussion that follows, the CPU utilization and cost statistics are presented for the machines that were actually used in the tests, rather than for a normalized CPU. In particular, this means that the numbers for the SQL server should be seen in light of the fact that the machine used for that component in this study was a 1 GHz Pentium III, compared the other components, which were 2.4 GHz Pentium 4s. This means that comparison of the loading and cost characteristics of those components requires translating between the different processor types.

CPU Utilization for a Representative Set of Operations					
FILENET CONTENT MANAGER component	**Browse 10 200 users**	**Browse 100 200 users**	**30K DB Retrieval 200 users**	**100K DB Retrieval 200 users**	**Search 200 users**
Application Engine	8.1%	15.0%	2.7%	2.9%	8.3%
Content Engine	7.5%	44.1%	3.8%	4.0%	12.1%
SQL Server (1GHz CPU)	2.6%	12.2%	1.5%	1.9%	4.6%

Table 1 CPU utilization for a representative set of operations. The values shown for the AE are the averages of the two AEs in the system.

The following table shows the calculated CPU costs for the same set of operations. These costs represent the number of CPU-milliseconds each component takes for each transaction. The costs are calculated from the observed throughput (in transactions per minute) and the measured CPU utilization for each type of transaction.

CPU Costs for a Representative Set of Operations					
FILENET CONTENT MANAGER component	**Browse 10 200 users**	**Browse 100 200 users**	**30K DB Retrieval 200 users**	**100K DB Retrieval 200 users**	**Search 200 users**
Application Engine	98	196	33	36	109
Content Engine	46	288	23	24	80
SQL Server (1GHz CPU)	16	80	9	12	30

Table 2 CPU costs for a sample of operations, in CPU-ms per transaction

This table shows that the browse operation was the most expensive to the AE and SQL CPUs (only logon was more expensive on the CE). From additional data, provided in the appendix, we also know that the cost grew quickly with the number of contained items. Document retrievals, on the other hand, remained relatively inexpensive, and did not show a strong dependence on the size of the returned document.

Several operations, most notably document retrievals, show a marked and consistent decrease in AE and SQL server CPU cost with increasing throughput. For example, at 50 users, the CPU cost on the AE for 30K database retrievals is 53 CPU-ms/transaction. At 100 users, this drops to 41.7, and by 200 users it is down to 32.6 CPU-ms/transaction. This pattern is repeated for all document sizes for both database and file store retrievals. While such a sequence of times seems counterintuitive, the result is likely due to caching; direct evidence of this has not yet been obtained, however, and the behavior warrants further investigation.

4.2 Response Times

The following charts show a selection of response times from individual tests. A representative sample was chosen for presentation here; for the complete set of results please refer to the appendix. The median times are reported in the charts in order to reflect the experience of a typical user of the system. The response times are those for what is referred to as the "primary transaction", which is the part of the compound transaction most

typically noticeable to a user, and which often occurs at the end of a compound transaction. See the appendix for a discussion of the transaction types.

The chart below shows the response times for the logon operation, as a function of throughput. At the high end of the load, a throughput of about 90 transactions per minute was obtained, with logon times of about 6 seconds. At lower loads, the response times were sub-second. The high-load times may appear problematic, with the system becoming bound by large logon times. However, this test consisted of logon operations alone, something not seen in real-world operation mixes. In such actual use cases, users tend to log on and perform a variety of other operations, such as browses, searches, retrievals, content creation, workflow operations, etc, before terminating the session. The logon/logoff-dominated test was used here mainly to characterize the CPU cost of the operation.

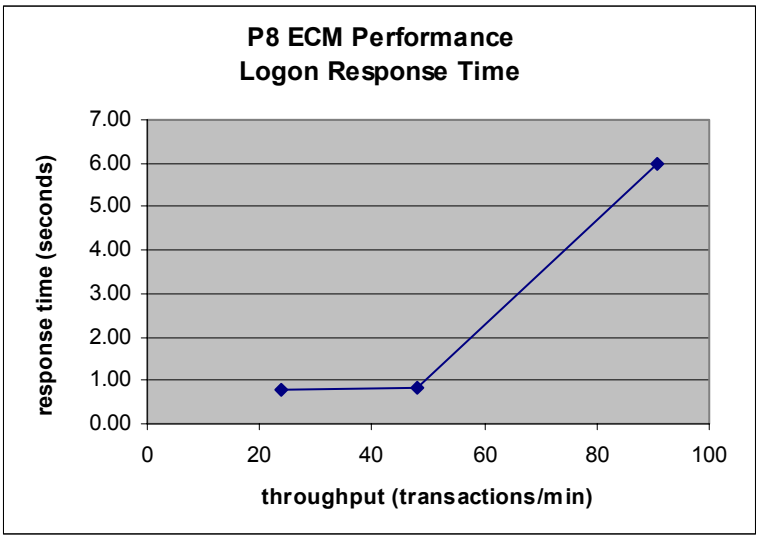


Figure 1 Response times for the logon operation

Below is a chart showing the response times for the browse operation. Times are shown for both folders containing 10 items and folders containing 100 items. The response times were noticeably dependent on the number of contained items; all load levels tested (up to 200 transactions per minute) produced sub-second response times with 10-item folders, while the times both were higher and grew faster when using folders with 100 items. Some increase in response time with increasing number of contained

items is to be expected, since an HTML representation of a subset of the items is returned with each browse call. The exact nature of this dependence is not revealed in these data, however, and further tests are warranted. Of particular interest is the question of whether the response time levels off after the maximum subset of returned items is reached.

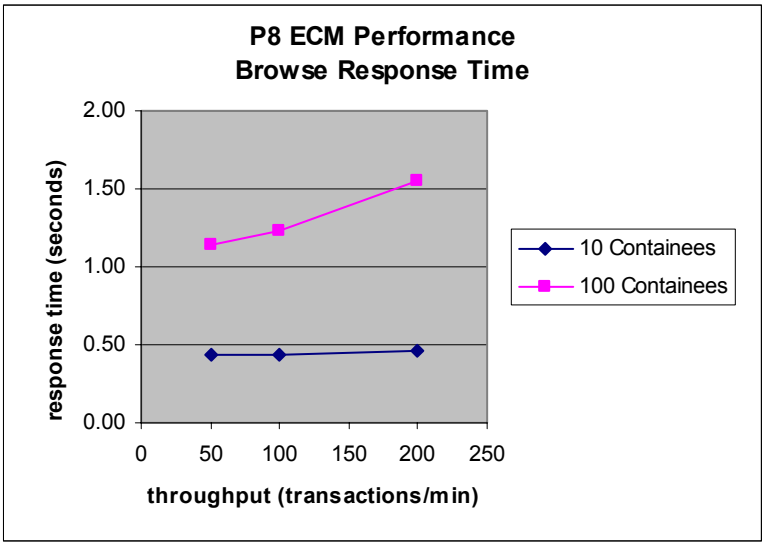


Figure 2 Response times for the browse operation

The next chart shows the response times for a typical document retrieval operation, the 30K retrievals. Tests were done using documents of sizes 3K, 10K, 30K, 100K, and 1M bytes, and all but the tests involving the largest documents showed the behavior exhibited in the chart: the response time dropped as the load increased. This was true both for documents being retrieved from the database and for those being retrieved from the file store.

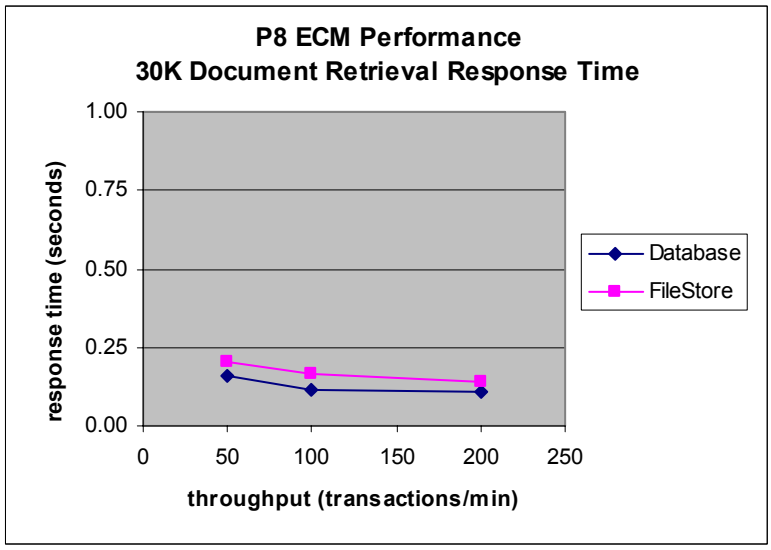


Figure 3 Response times for 30K document retrievals from the database and file stores. Note the decreasing response time with increasing load.

Finally, we present a chart showing the response times for the search operation. These times remained sub-second for all loads in the test range.

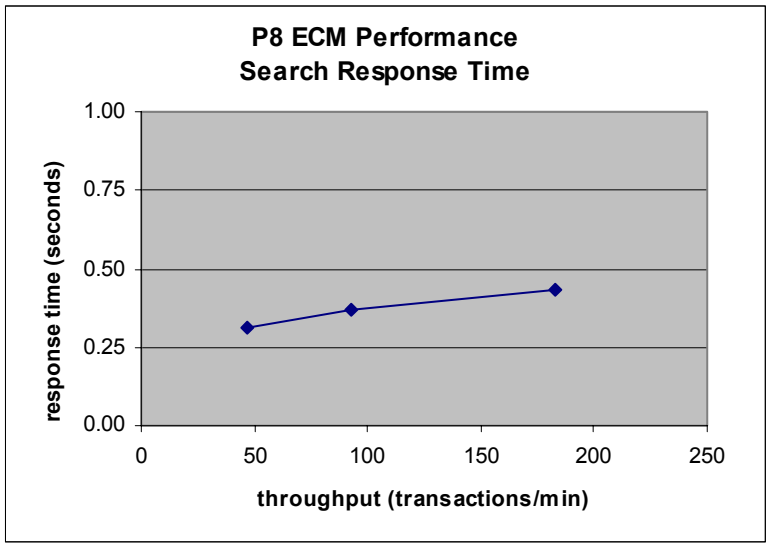


Figure 4 Response times for the search operation

5 Conclusion

This study presents a brief characterization of the performance of the FileNet Content Manager functionality. That characterization is chiefly in the form or CPU cost and response time data, and gives an initial data point for further comparison. All tests run were individual transactions, allowing for accurate CPU cost appraisals to be performed at low and moderate load levels. Later studies will include higher load levels and mixtures of transactions, better to simulate typical use cases.

The results from this study show a relatively high cost for the browse and logon operations, and a relatively low cost for document retrievals. The utilization and CPU cost dropped slightly with increasing transaction load, and, with the exception of the 1MB documents, did not vary appreciably with the size of document retrieved. Future scalability studies will be able to explore this relationship in detail.

Appendix A Database Population

As is described briefly in Section 2.1 above, a database of synthetic documents was created for these tests. The documents themselves were simple text files with the string "1234567890" repeated until the desired file size was reached. To create the data population, an object store was created using the Enterprise Manager. A custom population tool then synthesized the documents and added them to the object store using the Content Engine API. 250,000 documents were created in each of four sizes: 3K, 10K, 30K, and 100K bytes, and 2,000 documents were created of size 1M bytes, for a total of 1,002,000 documents. Of the non-1M documents, 125,000 were placed in the file store, and 125,000 were placed in the database store. All 2,000 1M documents were placed in the file store.

A hierarchy of folders was then created, and the documents filed in them. The hierarchy consisted of three trees of folders. Each tree consisted of a single "root" folder containing some number of sub-folders. Each sub-folder contained the same number of sub-folders in turn (this is referred to as the "width" of the tree). This pattern continued for a number of repetitions referred to as the "depth" of the tree. The final set of sub-folders, which did not themselves contain any sub-folders, are referred to as "leaf" folders. The root folder and all non-leaf folders contained the same number of documents.

The trees were organized as follows:

- 1 tree of width 10 and depth 3, for a total of 111 non-leaf folders, each of which contained 100 documents
- 1 tree of width 10 and depth 4, for a total of 1,111 non-leaf folders, each of which contained 100 documents
- 1 tree of with 10 and depth 5, for a total of 11,111 non-leaf folders, each of which contained 10 documents

All leaf folders in the above-described trees were empty; they contained no sub-folders and no documents.

Appendix B Description of the Test Process

All activities were authenticated with the system using a user account chosen randomly from a pool of 2,000 user accounts. Individual user names could be used multiple times, as no attempt was made to ensure that the chosen names were unique. In other words, the entire pool of names was

used every time a name was randomly selected, so some names may have been used more than others.

To avoid overwhelming the system with initial logons, each virtual client performed, as its first operation, a logon following a predetermined wait time. The wait time was different for each client, and was chosen so as to spread out the impact on the server of the flow of initial transactions. This approach was chosen in order better to simulate real-world customer use cases.

Following the initial logon, each client waited a random amount of time, and then executed the following steps:

1. Call the test script function that performs the sequence of HTTP calls required for the desired operation. (In this case the operation is always the same, since these tests are testing individual transactions.)
2. Verify the success or failure of the operation and its transactions. Failed transactions are not included in the captured statistics.
3. Call a function to determine a random amount of time to wait, and wait that amount of time.
4. Go back to step 1.

In step 2 above, the success of the transaction was verified before the result was counted. It should be noted that the tests for this paper contained no failures. In step 3 above, the "think times" were chosen from a Poisson distribution with a mean of 60 seconds. The Poisson distribution was chosen because of its ease of use and its similarity to the Gaussian distribution when the mean is large.

After each test, results for system statistics were gathered from an hour of "steady state" behavior in the middle portion of the test. For testing purposes, the system was regarded as being in a steady state once

- all virtual users had logged on,
- all virtual users had started performing activity,
- memory and processor utilization remained relatively constant, and
- the failure rate was sufficiently low (typically less than one half of one percent).

Appendix C CPU Utilization and Cost Data

What follows are three tables showing the CPU utilization for both AEs, the CE, and the SQL Server for 50, 100, and 200 transaction-per-minute loads. There were two AEs in the system, and each carried half the load, meaning that at 100 users, for example, each AE was carrying the transactions of 50 users. Note that for document retrieval tests, data were not collected for the 1 MB documents at the 200-transactions-per-minute load point.

CPU Utilization for a 50-Transaction-Per-Minute Load (%)					
operation	AE 1	AE 2	Average AE	CE	SQL (1GHz)
Logon	1.6	1.9	1.8	4.9	1.0
Browse 10	2.4	2.7	2.6	1.9	1.4
Browse 100	4.2	4.5	4.4	8.7	3.8
Search	2.5	2.7	2.6	2.2	1.5
Ret 3K DB	1.1	1.3	1.2	0.9	1.0
Ret 3K FS	1.1	1.1	1.1	1.1	1.0
Ret 10K DB	1.1	1.3	1.2	0.9	1.0
Ret 10K FS	1.1	1.3	1.2	1.1	1.0
Ret 30K DB	1.1	1.1	1.1	1.0	1.1
Ret 30K FS	1.1	1.3	1.2	1.1	1.0
Ret 100K DB	1.1	1.3	1.2	1.0	1.2
Ret 100K FS	1.1	1.3	1.2	1.2	1.0
Ret 1000K DB	1.5	1.7	1.6	2.1	2.1
Ret 1000K FS	1.5	1.8	1.7	2.0	1.0

Table 3 CPU utilization for a 50-transaction-per-minute load

CPU Utilization for a 100-Transaction-per-Minute Load (%)

operation	AE 1	AE 2	Average AE	CE	SQL (1GHz)
Logon	2.6	3.0	2.8	10.6	1.0
Browse 10	4.7	4.2	4.5	3.9	1.8
Browse 100	7.7	8.3	8.0	19.1	6.6
Search	4.3	4.6	4.5	4.8	1.8
Ret 3K DB	1.6	1.8	1.7	1.8	1.1
Ret 3K FS	1.7	2.0	1.9	1.9	1.1
Ret 10K DB	1.6	1.8	1.7	1.8	1.2
Ret 10K FS	1.6	1.9	1.8	1.9	1.0
Ret 30K DB	1.6	1.9	1.8	1.9	1.2
Ret 30K FS	1.7	1.8	1.8	2.2	1.1
Ret 100K DB	1.7	2.0	1.9	2.0	1.4
Ret 100K FS	1.6	2.0	1.8	2.3	1.1
Ret 1000K DB	2.5	2.9	2.7	3.9	3.2
Ret 1000K FS	2.6	3.0	2.8	4.2	1.1

Table 4 CPU utilization for a 100-transaction-per-minute load

CPU Utilization for a 200-Transaction-per-Minute Load (%)					
operation	AE 1	AE 2	Average AE	CE	SQL (1GHz)
Logon	3.8	4.3	4.1	42.6	1.0
Browse 10	7.7	8.4	8.1	7.5	2.6
Browse 100	14.7	15.3	15.0	44.1	12.2
Search	7.9	8.6	8.3	12.1	4.6
Ret 3K DB	2.3	2.6	2.5	3.6	1.3

CPU Utilization for a 200-Transaction-per-Minute Load (%)					
operation	AE 1	AE 2	Average AE	CE	SQL (1GHz)
Ret 3K FS	2.4	2.9	2.7	4.2	1.2
Ret 10K DB	2.4	2.7	2.6	3.6	1.4
Ret 10K FS	2.4	2.8	2.6	4.2	1.3
Ret 30K DB	2.5	2.9	2.7	3.8	1.5
Ret 30K FS	2.4	2.9	2.7	4.3	1.2
Ret 100K DB	2.7	3.1	2.9	4.0	1.9
Ret 100K FS	2.7	3.1	2.9	4.5	1.2

Table 5 CPU utilization for a 200-transaction-per-minute load

What follows are tables giving the calculated CPU costs for the main system components.

CPU Cost for the AE			
operation	50 users	100 users	200 users
Logon	174.4	139.7	107.0
Browse 10	122.6	109.6	98.3
Browse 100	212.9	197.8	195.5
Search	133.7	114.7	108.6
Ret 3K DB	57.9	40.9	29.9
Ret 3K FS	52.3	44.0	32.2
Ret 10K DB	57.5	41.4	30.8
Ret 10K FS	57.3	42.9	31.9
Ret 30K DB	53.0	41.7	32.6
Ret 30K FS	56.7	42.2	32.7

CPU Cost for the AE			
operation	50 users	100 users	200 users
Ret 100K DB	60.0	45.2	35.5
Ret 100K FS	56.7	44.1	34.9
Ret 1000K DB	81.5	69.2	N/A
Ret 1000K FS	82.0	70.5	N/A

Table 6 CPU cost for AE, in CPU-ms per transaction

CPU Cost for the CE			
operation	50 users	100 users	200 users
Logon	247.6	264.0	563.2
Browse 10	45.3	47.6	45.8
Browse 100	213.5	236.6	287.9
Search	57.0	62.2	79.9
Ret 3K DB	22.9	22.0	22.2
Ret 3K FS	26.0	22.8	25.3
Ret 10K DB	22.2	21.9	21.9
Ret 10K FS	26.3	22.9	25.4
Ret 30K DB	23.9	22.9	23.0
Ret 30K FS	26.6	26.5	26.0
Ret 100K DB	25.2	24.3	24.2
Ret 100K FS	28.7	27.9	27.4
Ret 1000K DB	53.0	50.8	N/A
Ret 1000K FS	50.6	53.2	N/A

Table 7 CPU cost for the CE, in CPU-ms per transaction

CPU Cost for the SQL Server (1GHz CPU)			
operation	50 users	100 users	200 users
Logon	48.1	23.9	13.1
Browse 10	33.9	22.3	15.8
Browse 100	92.4	82.2	79.7
Search	37.3	22.8	30.4
Ret 3K DB	25.0	13.6	8.0
Ret 3K FS	24.6	13.1	7.5
Ret 10K DB	24.8	14.1	8.4
Ret 10K FS	24.7	12.5	7.6
Ret 30K DB	26.0	14.6	9.2
Ret 30K FS	24.2	13.0	7.3
Ret 100K DB	28.6	17.3	11.8
Ret 100K FS	24.5	12.8	7.4
Ret 1000K DB	52.3	41.3	N/A
Ret 1000K FS	25.3	13.8	N/A

Table 8 CPU cost for the SQL Server, in CPU-ms per transaction

Appendix D Response Time Data

What follows is a chart giving the response times for the primary transactions discussed in this paper. All times are in milliseconds. Note that times were not collected for the 1MB documents at the 200-transactions-per-minute load point.

Response Times for Primary Transactions			
operation	50 users	100 users	200 users
Logon	771	850	6002
Logoff	118	57	165
Browse 10	435	441	457
Browse 100	1142	1226	1550
Search	315	368	431
Ret 3K DB	162	91	86
Ret 3K FS	175	157	115
Ret 10K DB	166	103	97
Ret 10K FS	198	150	132
Ret 30K DB	161	114	107
Ret 30K FS	203	166	143
Ret 100K DB	217	213	213
Ret 100K FS	212	191	169
Ret 1000K DB	1238	2145	N/A
Ret 1000K FS	1152	1975	N/A

Table 9 Response times in milliseconds for primary transactions

Appendix E Glossary

AE	Application Engine; the component of a FILENET CONTENT MANAGER system that handles specialized application and web services
applied workload	the throughput that the system would demonstrate if all response times were zero
CE	Content Engine; the component of a FileNet Content Manager system that handles document management

CPU cost	the number of milliseconds a CPU spends on each transaction
DC	Domain Controller; the component of a FileNet Content Manager system that handles account authentication
median	a statistical measure of a data set. The median is the middle item in a data set when all the values are in sorted order. As such, it measures the "typical" data point, since it is not influenced by the values of the data points at the ends of the set.
millisecond	one one-thousandth of a second
script	a list of commands interpreted by a tool specialized for driving test sequences
throughput	the amount of actual work a system is doing, measured, for the purposes of this paper, in transactions per minute
virtual client	a simulated user of the system, performing the same set of transactions typically performed by a user, but simulated in software
workload	the amount of work the FileNet Content Manager system is being asked to perform, usually measured in number of virtual users of the system

Appendix F Transactions

Throughout this document, references are made to "transactions" and "operations", often with qualifying terms such as "primary" and "compound". A brief definition of these terms is in order.

Because many different activities are simultaneously happening during a performance test of this nature, it is important to define carefully what it is we are going to measure, and decide what significance we are going to attach to those measurements. The overall goal to is simulate, at least broadly, the actions of actual users of a system, so that the performance characteristics observed during the tests can have a reasonably direct application to real-world situations, and can be used to help predict the sizes of systems necessary to reach certain stated performance requirements. With this in mind, a set of terms has been adopted for use in our technical white papers:

compound transaction	this is the entire transaction, from beginning to end, and consists often of many smaller transactions between a client and server. These may include sequences of page transitions initiated by user actions, HTTP page requests of all types, database queries, etc.
primary transaction	this is the "main" transaction in a compound transaction. There is one and only one primary transaction in each compound transaction, and it is typically the last one. This is the transaction that is timed for the purposes of response time characterizations, as

	it is the one most often associated by actual users to the activity in question (browsing, searching, etc). The primary transaction time is usually the time from the user clicking the "search" (or similar) button after filling out a form, for example, until the requested page returns. The compound transaction includes all the other activities of getting the previous pages.
operation	a general term used to indicate any activity, including a small transaction discussed above, or a primary or even compound transaction

Because there is a one-to-one relationship between compound and primary transactions, the count used for throughput calculations can be based on either. However, the times reported in the response time tables are for primary transactions, and so necessarily omit a significant fraction of time the spent on the server and client with related activities. Thus, if one were to use the reported response times to do calculate the throughput (assuming that all the time was accounted for), an incorrect result would be obtained.

Appendix G Calculation of CPU Cost

The CPU cost of various operations is mentioned throughout this document. It is a useful way to characterize the performance of a system regarding particular operations or sets of operations. It is reported here in terms of CPU-ms per transaction, or the number of milliseconds a CPU spends on a given transaction, on average. This value is calculated as follows:

$$c = \frac{u}{t} \times 600p,$$

where c is the CPU cost, u is the utilization in percent, t is the measured actual throughput in transactions per minute, and p is the number of processors on the machine. The factor 600 normalizes the units. This number gives the cost for the CPU in question; if a standardized measure is needed, such as CPU-ms on a 1 GHz Pentium III, then an additional factor will have to be employed. The costs in this paper are costs for the systems used, and are not further converted.

White Paper

PERFORMANCE – PROCESS ENGINE

Presented by FileNET Product Management

March 27, 2002

Table of Contents

INTRODUCTION

This document discusses the results of tests of the system, with particular focus on the Process Engine. End-to-end response times for various operations are given, and, together with CPU usage data from the different components, are used to characterize the scalability of the system.

The audience for this document, and for the remainder of the performance white paper series, includes FileNET personnel from Engineering or Marketing and, following appropriate review, other readers interested in the performance characteristics of the product.

TEST METHODOLOGY AND ENVIRONMENT

TEST PHILOSOPHY

FileNET conducted this study to evaluate the performance and scalability of the product. Tests were designed to provide information that would be useful for sizing systems, calculating CPU costs, and measuring response times. This paper does not provide a large-scale proof of concept of the system regarding performance under high loads; rather, the chief aims of the testing process were to provide sizing guidelines and system characterization.

This document is one of a larger sequence of white papers. Cumulatively, these papers are intended to document the performance and scalability of the product in its various uses and configurations. This paper, individually, documents a subset of those uses in a single configuration. The tests that are documented here were designed to exercise the Process Engine

TEST METHODOLOGY

Throughput measures are reported in terms of successfully completed transactions per minute. Workload is generated using virtual clients, each of which performs a single transaction, on average, per minute. Specifically, clients draw from a Poisson distribution with a mean of one minute.

For each set of tests we evaluated, experiments were performed to determine the applied workload required to exercise the system without causing the least scalable component (i.e., the component with the least throughput capacity) to become bottlenecked. Once the relationship between workload and CPU cost was sufficiently well understood, applied workloads were chosen for the tests reported in this paper which resulted in CPU loads of approximately 10%, 40%, and 70% on that least scalable component in each of the tests.

The system was populated with work items for users in defined work queues. The number of work items and the number of users determines the overall queue depth and size of a system. Unless stated otherwise, testing targeted one workflow and one process queue.

Process Engin was designed to allow the use of multiple Application Engines (AE). The test harness used for these tests included a single AE; readers who need sizing projections for larger systems can extrapolate using the CPU cost data provided for the Content Engine, Process Engine, and Domain Controller. Later white papers will directly address the performance and scalability of larger systems.

DESCRIPTION OF TESTS

This section comprises of a brief discussion of the composition and execution of the test set.

JAVA CLASSES

The foundation of the performance tests is a set of custom Java classes. Each method of these classes provides the complete features for one of the activities

included in the tests. All activities remain stateless except for re-use of the logon (e.g., the session object).

INTERFACE

Both HTML and JSP pages are used by the testing framework. HTML pages provide the ability to supply data input, while JSP pages provide the basic logic flow and invoke methods from the Java classes to complete each activity.

TEST SCRIPTS

A third-party web testing and simulation tool drives all performance activities. Script logic uses the HTML pages mentioned above to execute work in the system, and is integrated with the existing infrastructure for providing data input, reporting transaction times, etc.

DATABASE POPULATION

A set of simple HTML and JSP pages constitutes a set of known data. Data was created in the system by using the testing tool to drive the population pages.

The system was populated with work items for users in defined work queues. The number of work items and the number of users determine the overall queue depth and size of a system.

A system with 100 data fields, 25 fields exposed in the corresponding work queue, and 50 parameters in each step of the workflow was used as a "large" workflow.

The queue was examined before each run to ensure that it contained at least 20 work items for each user. If a user allocation was below 20, additional work items were added to make up the difference. Work items were created for a maximum of 2000 users, bringing the total number of work items to at least 40,000. In this case, every search by a particular user returned about twenty work items.

OVERALL TEST FLOW

All activities authenticate with the system using a non-unique domain user account. Each transaction (activity) creates the session object, passing the user information, within the first instance of that transaction. At the end of the transaction the session object is saved. It is retrieved for re-use in subsequent transactions.

To avoid overwhelming the system with initial logons, each virtual client waits for a different, pre-determined amount of time before performing its first transaction. For example, if the inter-logon delay were three seconds, then virtual client #1 would log on, followed three seconds later by virtual client #2, followed by another delay, followed by virtual client #3, etc. Such a scheme allows a steady, constant rate of initial logons.

For the remainder of the test duration, each virtual client executes the following steps:

1. Choose a job at random to perform based on the desired distribution of work.
2. Call the Java Script function that performs the sequence of HTTP calls required to perform the desired job. (These functions capture the timers and counters required to produce detailed information on throughput and response times.)
3. Verify the success or failure of the job and its transactions. Failed transactions are not included in the statistics captured.
4. Sleep for a randomly-assigned amount of "think time". This time is chosen from a Poisson distribution with a mean of 60 seconds, meaning

that over a period of one hour, virtual clients will perform an average of 60 transactions each.

5. Go back to step 1 in this sequence.

During the test runs, each test continues for two hours, to ensure that at least 90 minutes of "steady state" behavior can be observed. For testing purposes, the system is considered to be in a steady state once

- all virtual users have logged on,
- all virtual users have started performing activity, and
- memory and processor utilization remain relatively constant.

WORKLOAD DESCRIPTION

Each workload comprises a list of transactions. The following table describes the mixed test workload and relative frequency of each type of operation for the activities discussed in this paper. This workload reflects a "workflow-centric" test scenario with no content management operations.

Each user randomly executes one of the six types of transactions according to the distribution in the following table. (Note that these percentages were calculated to mimic the workload used in previous Panagon eProcess testing efforts.)

Transaction	Percentage of Total
Logoff/Logon	5%
Launch Workflow	28%
QueryQueue	24%
QueryWorkItem	19%
DispatchWorkItem	12%
UpdateWorkItem	12%

The test script was created by using an application written specifically for performance testing.

WORK ACTIVITIES

This section discusses details of the individual types of activity used in the test cases.

LOGON/LOGOFF

The Logon/Logoff transaction consists of a logoff followed by a logon. If the virtual user is already logged on, the transaction will log the user off and then log the user back on again. The session object is maintained appropriately. A non-unique domain user account is used for authentication.

LAUNCH ITEM

The Launch Item activity simulates the simple launch of a workflow (with no attachments). Values for all step parameters are randomly generated.

QUERY QUEUE

Given specific criteria, the Query Queue activity retrieves the data fields for a number of work items which match the criteria. The system is populated with a reasonable

number of work items. Different users and queues can be given different thresholds so queues will contain a varying number of items. Currently, all exposed queue fields are fetched from the system.

Each queue contains approximately 25 work items. One queue is named by each query, and that queue is returned. All exposed fields in the queue are displayed.

If the workload mix includes the Dispatch Work Item transaction, the number of work items assigned to a user could drop during the test run as work items terminate. To ensure a completely stable set of data for retrieval, the Dispatch Work Item transaction is not included in the workload mix.

QUERY WORK ITEM

The Query Work Item activity retrieves the data fields for a single work item, given specific criteria. Currently, all step parameters are fetched from the system, but only the subject field is passed back to the client. Additionally, the item is not locked while it is being fetched.

In the future, these tests could be enhanced to provide control of the result set retrieved (i.e., system properties only, exposed properties, step information, etc).

DISPATCH WORK ITEM

Given specific work item identification, the Dispatch Work Item activity locates the item using a user-defined index and updates a few specific data fields. The updated data fields are not used in any portion of a query, so there is no problem with data modification affecting other tests. The work item is then dispatched to the next state.

UPDATE WORK ITEM

Given specific work item identification, the Update Work Item activity locates the item using a user-defined index. All parameters on the current step are updated (as with Dispatch, the updated parameters are not used in any portion of a query, so there is no problem with data modification affecting other tests) and the work item is saved.

WORKFLOW MAPS

For development of the tests, the workflow map was kept very simple with regards to the number of steps, data fields included, routes, etc. An example of this very simple workflow is as follows:

The work items have different sizes and numbers of data fields to better simulate real world situations. The workflow has at least one data field of each of the allowable types including several that are arrays.

DESCRIPTION OF THE HARDWARE ENVIRONMENT

The hardware consisted of separate machines for the different components as follows:

- **Domain Controller**
  DELL PowerEdge 2450
  Pentium III 2 x 1GHz
  512 MB RAM
  Windows 2000 Advanced Server
- **Content Engine**
  DELL PowerEdge 2450
  Pentium III 2 x 1GHz

512 MB RAM
Windows 2000 Advanced Server

- **Application Engine**
  Sun Ultra-Sparc II
  dual 450 MHz
  2 GB RAM
  Solaris 8
- **Process Engine**
  DELL PowerEdge 2450
  Pentium III 2 x 1GHz
  512 MB RAM
  Windows 2000 Advanced Server
- **File Store Server**
  DELL PowerEdge 2450
  Pentium III 2 x 1GHz
  512 MB RAM
  Windows 2000 Advanced Server
- **SQL Server**
  DELL PowerEdge 2450
  Pentium III 2 x 1GHz
  512 MB RAM
  Windows 2000 Advanced Server
- **Console**
  DELL PowerEdge 2450
  Pentium III 2 x 500 MHz
  512 MB RAM
  Windows 2000 Advanced Server

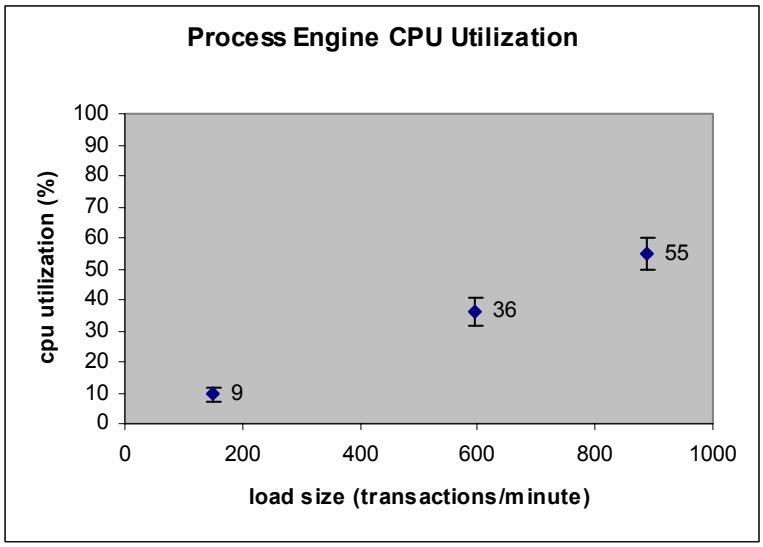
End-to-End Scalability of the System

CARRYING CAPACITY BY COMPONENT

This section discusses the capacity of each major component of the system in terms of processor utilization. Each chart shows the utilization of the processor for a component at three different load values. The three values form a line, the slope of which gives the rate at which new transactions are loading the system. This leads to the cost to the component in terms of CPUms per transaction. See Appendix A for a discussion of that calculation.

PROCESS ENGINE

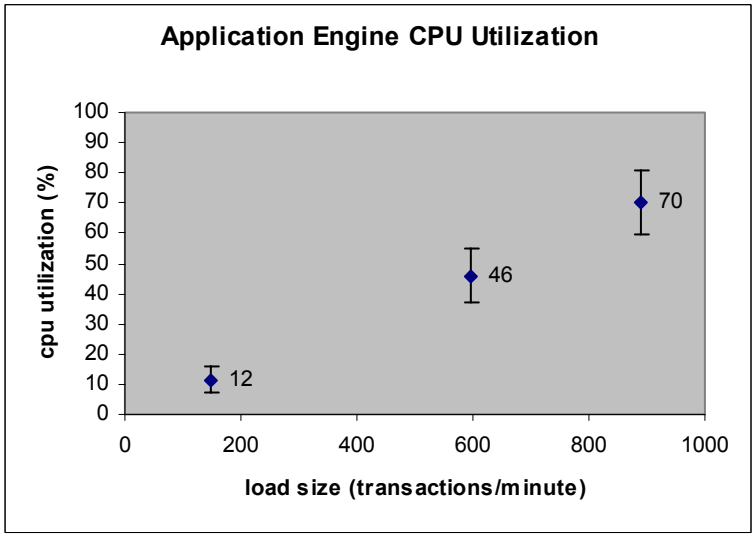
The utilization of the Process Engine is shown in the following graph.



As the load increased to almost 900 transactions per minute, the CPU responded in a linear manner. The variation in processor response remained quite small (only a few percent), and the average transaction cost the CPU approximately 74 ms.

APPLICATION ENGINE

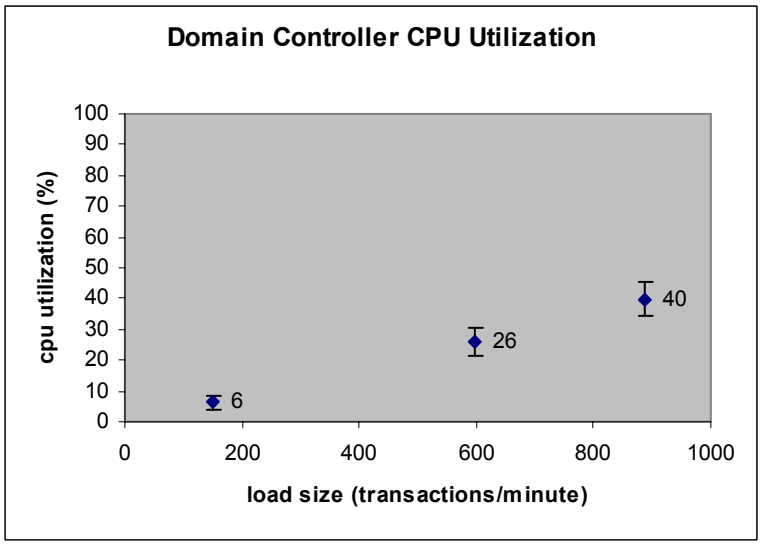
The response of the Application Engine CPU to the applied load is given in the following graph.



The utilization of the Application Engine CPU grew in a linear fashion through the test range, reaching 70% at the peak of 900 transactions per minute. At this rate of growth, the average transaction cost the CPU 44.5 ms (in standardized CPU ms; see Appendix A), making the Application Engine the component with the largest CPU cost (and thus the most CPU-bound).

DOMAIN CONTROLLER

The utilization of the Domain Controller CPU is given in the following graph.



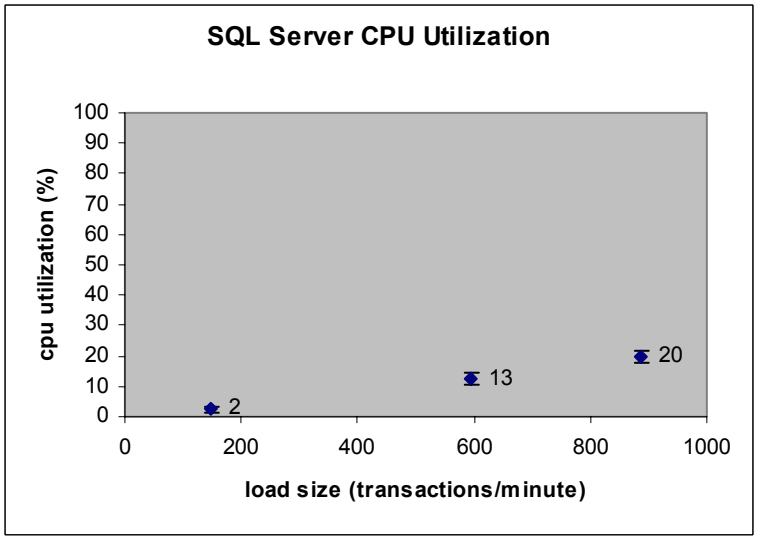
The utilization grew in a linear manner throughout the test range, reaching 40% at almost 900 transactions per minute. The variation remained small for all data points. The CPU cost to the Domain Controller for the average transaction was approximately 54 ms. This utilization is somewhat high, and is the result of known problems. It is expected to be lower in future patch releases.

CONTENT ENGINE

Because the Content Engine was not used in the transactions involved in this test, the CPU utilization of that machine remained at a small fraction of 1% for all data points in the test range.

SQL SERVER

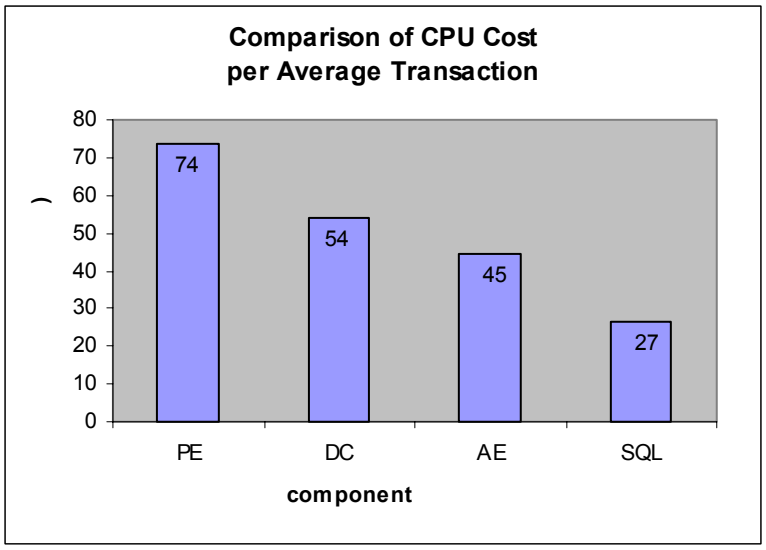
The utilization of the SQL Server CPU is shown in the following chart.



The utilization remained low throughout the range, and grew in a predictable, linear fashion, indicating that the amount of load hitting the SQL server is not sufficient to cause a bottleneck (there was only 20% utilization at 900 transactions/minute). The cost was approximately 27 CPUms/transaction.

SUMMARY

Below is a chart summarizing the CPU cost data discussed in the preceding few sections.



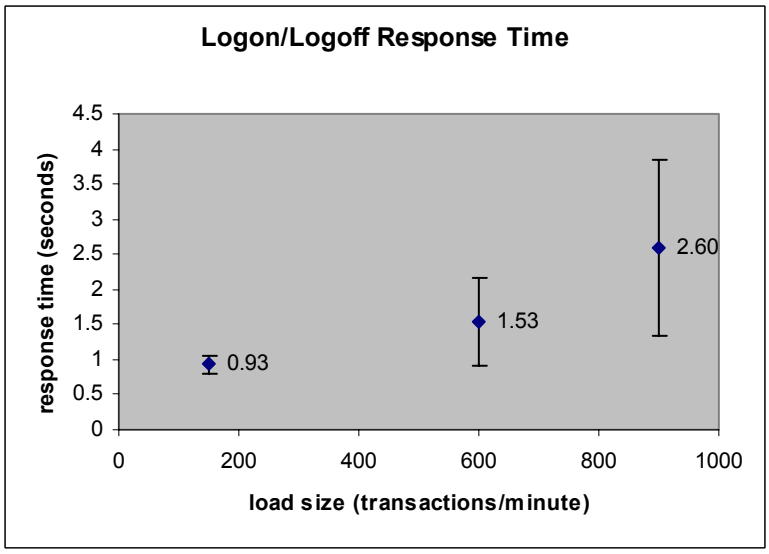
Although the Application Engine is the component most closely CPU-bound, having a CPU utilization of 70% on the high end, the Process Engine has the largest CPU cost (because the Application Engine is a less powerful machine).

End-to-End Performance of Major Functions

In the following sections are brief descriptions of the response times of tested operations. The test scenario used was the mixed set discussed earlier, and the response times are separated in this section by type of activity.

LOGON

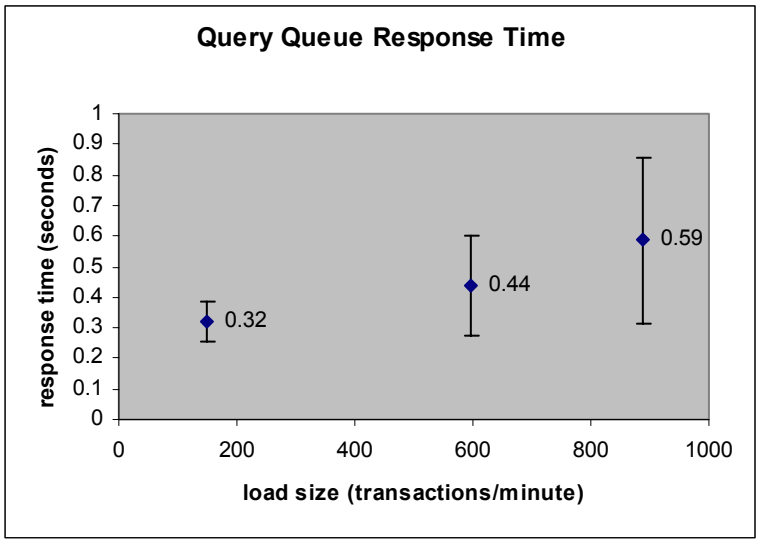
The following chart shows the response times for the Logon/Logoff test.



The response times for the 900 transactions/minute case began to rise non-linearly. This may be attributable to the fairly high CPU load on the Application Engine (70%).

QUERY QUEUE

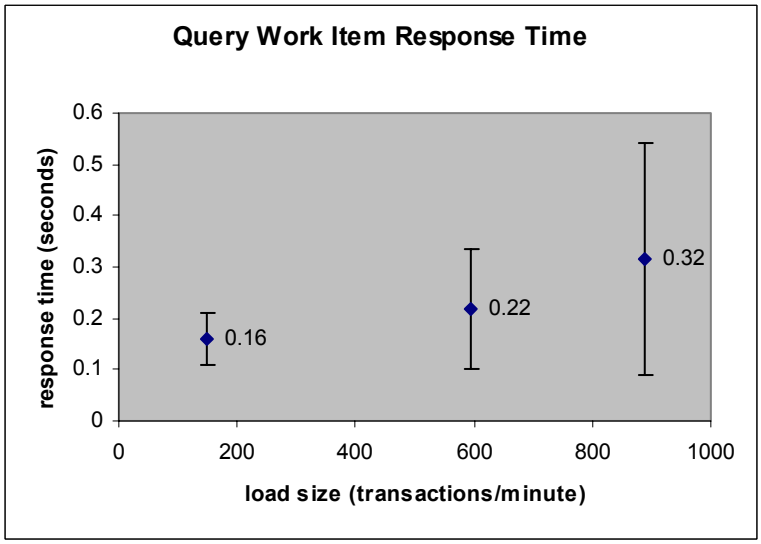
Below is a chart showing the response times for the Query Queue activity.



The response times here showed the same beginning of non-linear growth at the high end of the test range. Even with this growth, the times remained low, at only .59 seconds for 900 transactions/minute.

QUERY WORKFLOW

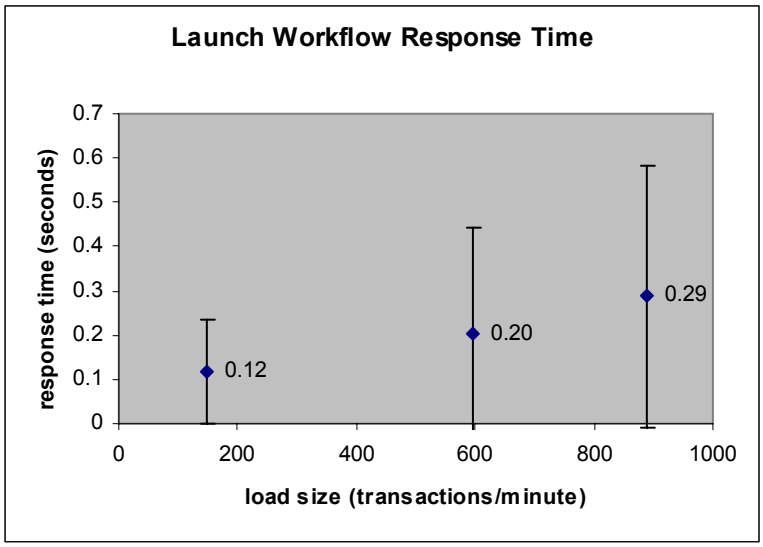
The response times for the Query Workflow activity are presented in the following chart.



As with the Query Queue response times, with the Query Workflow activity we see the beginnings of non-linear growth at the 900 transactions/minute level. The times themselves remained reasonable throughout the range, being sub-second even at the high end.

LAUNCH WORKFLOW

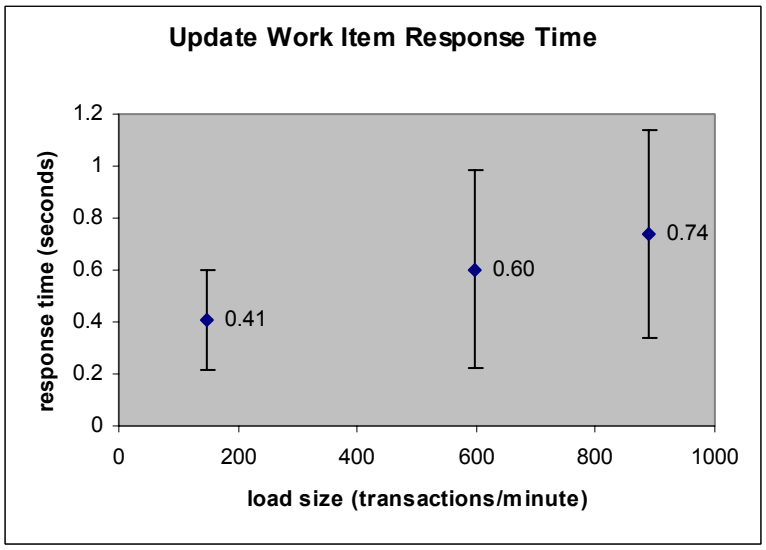
The chart below gives the response times for the Launch Workflow activity.



These response times remained in a linear pattern of growth through the 900 transactions/minute load level, and the top times averaged only 0.29 seconds, with the variation small enough to keep almost all times below one second.

UPDATE WORK ITEM

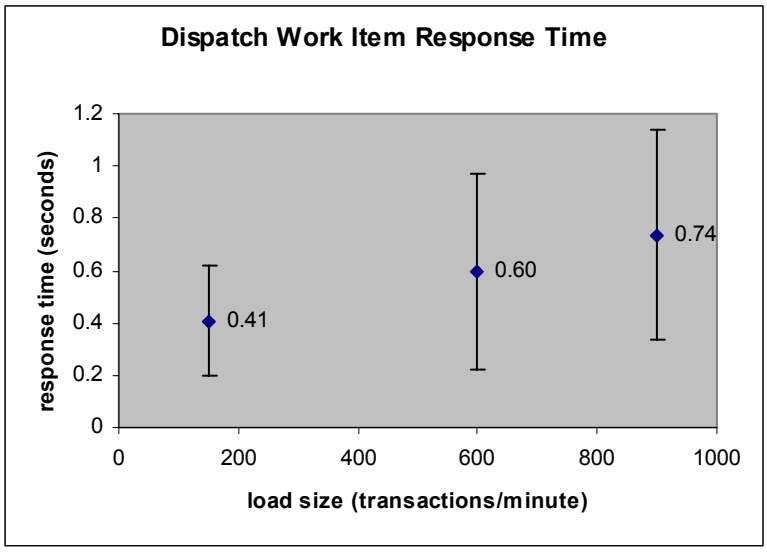
Following is a graph showing the response times for the Update Work Item activity.



The response times here remained linear throughout the testing range, and the most of the high-end times were less than one second.

DISPATCH WORK ITEM

Below is a chart showing the response times for the Dispatch Work Item activity.



The times for this activity also remained linear in growth throughout the tested region, and the values at the high end reached only three quarters of a second with reasonable variation.

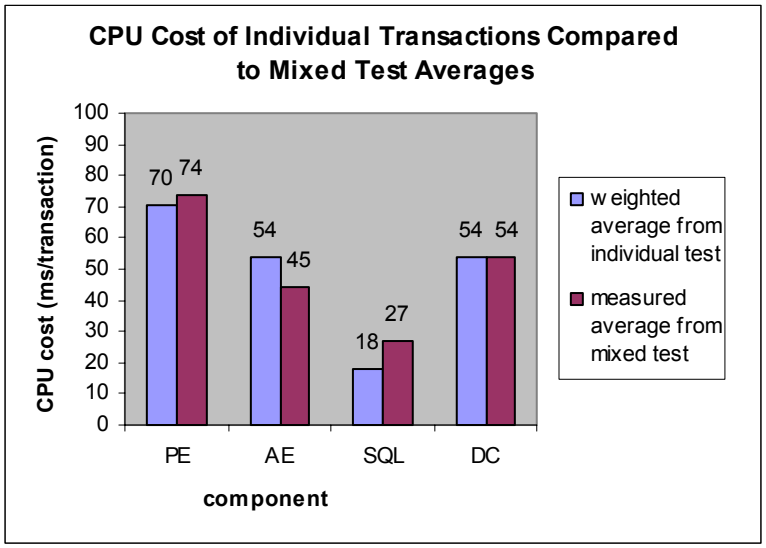
CPU Cost of Individual Transactions

Additional tests were run in which only one transaction type was used for the entire duration of the test. This resulted in CPU usage statistics for each type of transaction on the Process Engine, the Application Engine (the most CPU-bound component of the system), the SQL Server, and the Domain Controller. These were converted into CPU cost in terms of CPUms/transaction (see Appendix A for a discussion). These costs are presented in the table below:

transaction	CPU Cost (CPUms/transaction)			
	PE	AE	SQL	DC
Dispatch	78.7	35.7	43.6	148.8
Launch	47.4	30.9	6.0	4.1
Logon	96.0	297.7	2.0	8.1
Query Work Item	28.9	27.4	8.8	77.8
Query Queue	116.1	67.0	18.3	4.0
Update Work Item	80.1	37.6	40.9	154.0
weighted average	70.5	53.6	18.0	53.6
Measured average	74	44.5	27	54

At the bottom of each column in the above table is the weighted average of the column. These are calculated from the proportion of the total assigned to each type of operation, as given in section ***Workload Description***, and are compared to the measured averages,

determined from the mixed tests discussed in section ***Carrying Capacity by Component***. This comparison is shown in the following chart.



The relatively small differences in the measured versus calculated averages are most likely due to different system loading in the single transaction tests; i.e., since all transactions are the same, the effects of a given transaction's CPU cost are exaggerated. Overall, it can be seen that the values of measured and calculated weighted average are in reasonable agreement.

Discussion of Performance Characteristics

The primary goals of this paper were to characterize the performance of the system with the Process Engine, and to give an indication of the scalability of the system to larger configurations.

Larger systems generally require the response times on all operations to be sub-second. This requirement was met by nearly all operations; only the Logon/Logoff stands out at the upper end of the test range as having unacceptably high response time.

The components performed well, with the Application Engine being the only CPU-bound element. Of all the components, the Application Engine is the one we would generally prefer to become CPU-bound first, as it is the component most readily configured in a farm setup. Configurations for handling larger number of users will use a farm of Application Engines, and future papers will explore such configurations. Future papers will also explore configurations with large databases. With the current configuration, the data suggest that 900 transactions per minute is near the maximum workload the system can handle while maintaining sub-second response times on most operations. Multiple Application Engines (and, later, Content Engines) will help raise this number significantly.

Appendix A: Calculation of CPU Cost

CPU cost numbers are presented at several points in the paper. Those numbers are reported in terms of CPUms/transaction, and constitute a useful comparative measure. The CPU cost is calculated as follows: we start with the CPU utilization percentage, *u*, for a given system, and the throughput rate, *t*, given in transactions per minute on that machine. We take, usually, several data points and plot them, to show scalability of the system. The plot is *u* as a function of *t*. This curve is often linear. If it is not, then we are most interested in the numbers at the high end; the slope of the curve tells us the cost of adding additional throughput. Thus, we are interested in the slope at the high end of the curve,

$$\left.\frac{du}{dt}\right|_{t_{\max}}$$

where $t_{\max}$ is the maximum throughput for which data are available. Typically, the progression throughout the test region is linear, and so the value of this slope is just the same (approximately) for all values of *t* in the range.

The slope given above has the units of %CPU/ (transaction/minute). We would like to end up with a figure in CPUms/transaction, so we must take the slope and change the units. There are 60 seconds in a minute, and 1000 ms in a second. Furthermore, we want to change from % to a fraction, which means dividing by 100%. Thus, the CPU cost, *c*, given in CPUms/transaction, is

$$c = \left.\frac{du}{dt}\right|_{t_{\max}} \times 60\frac{\text{sec}}{\text{min}} \times 1000\frac{ms}{\text{sec}} \times \frac{1}{100\%}$$

$$= 600\left(\left.\frac{du}{dt}\right|_{t_{\max}}\right)$$

Because the curve is usually linear for the region being tested, the derivative above can usually be gotten by just doing a linear fit of the data points.

The above calculation gives the number of milliseconds a given processor spends on each transaction. These numbers can be directly compared when the machines are the same, but when different types of machines are used for different components (as is the case in these tests, as the Application Engine is a Sun box), and then a compensating factor must be included. In particular, from SPEC CPU marks we have that the AE should be factored down by 410/874 (it is a dual 450 MHz UNIX machine). Numbers given for the AE in the paper have had this factor applied when calculating the CPU cost.

Appendix B: Glossary

Application Engine	the component of a system that handles specialized application and web services
Content Engine	the component of a system that handles document management
CPU bound	characterized by a state in which an increase in CPU utilization will result in a marked decrease in performance
CPU cost	the number of milliseconds the CPU spends on each transaction
Domain Controller	the component of a system that handles account authentication
Process Engine	the component of a system that workflow and process management
queue	a list of transactions awaiting processing, which is serviced in a first-in-first-out manner
script	a list of commands interpreted by a tool specialized for driving test sequences
throughput	the amount of actual work a system is doing, measured, for the purposes of this paper, in transactions per minute
transaction	the basic unit of work in a system, consisting of an operation involving workflow processing, client authentication, document management, etc
virtual client	the unit of applied workload, which simulates an actual user of the system by performing the same set of transactions typically performed by a user
workload	the amount of work the system is being asked to perform, usually measured in number of virtual users of the system

Sample Large Installations

Pages/month	Scan Customer Name	Application(s)
13,000,000	New Jersey, State Of, Division Of Taxation	Document Mgt/Store & Retrieval State Tax
9,800,000	Delta Airlines	Accounts Receivable Implementation
7,500,000	Northwest Airlines, Inc	Accounts Receivable
6,000,000	Shaw Industries	
5,000,000	Adminastar, Inc.	Claims Processing Customer processing
4,000,000	McGraw Hill	Educational Testing
4,000,000	The Psychological Testing Corporation	High Volume Storage
4,000,000	Colonia Nordstern Verwaltung Und Finanz Ag	Processing Underwriting
3,000,000	Yellow Freight Systems	Accounts Payable Accounts
3,000,000	American Airlines	Litigation Support
3,000,000	Ft Mortgage	Loan low.
2,750,000	Money Store, The	
2,000,000	Sks - Credit Suisse	Accounting, Customer Service
2,000,000	Banc One Mortgage	Customer Service Includes Escrow, Audit,
1,380,000	Mbna America Bank	Customer Service
6,000,000	First Union Mortgage	Loan servicing - Mortgage.