OPENTEXT

Case360 Product Overview

Release 11.4

Copyright

Table of Contents

Chapter 1

Overview of Case360

Case360 is a Java based application development platform used to add Web-based work management capabilities to applications. These capabilities enable businesses to automate diverse business processes, manage their data, and streamline their business operations.

Case360 is implemented using the Sun Java 2 Enterprise Edition (J2EE) Enterprise Java Beans (EJB). An intuitive user interface includes tools that provide administrative, application design, query, and process functions.

Capabilities

Using the Case360 tools, developers can implement the following core work management capabilities: Case Management, Process Management, and Content Management. Additionally, applications can access data stored in external repositories by using Case360 Enterprise Document Access Connectors.

The Case360 capabilities can be implemented individually or they can be combined to provide a highly integrated system.

Case Management

Case Management capabilities are provided by Casefolder. Casefolder provides a way to collect, store, organize, and manage information associated with a particular case. Examples of cases include insurance claims and loan applications.

In addition to providing methods for managing business data, Casefolder also supports collaborative processing of cases, which are the electronic equivalents of paper-based case files that can contain forms, documents, images, and other information related to a specific case.

The relationship between items in casefolders can be managed through Casefolder rules, as well as through tasks and deadlines. Collaboration features include the Discussions feature, which manages discussion topics among users working on a case; shared concurrent access with conflict detection; task assignment; status; deadlines; history; subscription; and "what's new" functionality.

Process Management

Process Management capabilities are provided by Process. Process is used to distribute work within an organization to different work groups, where specialized processing can take place. Process routes data - referenced by process instances - to designated people or activities, where predefined rules are applied and actions are performed based on specified conditions.

Process instances can reference objects that are stored in different repositories. For example, when casefolders are integrated with the process, they can automatically be routed to the appropriate users for processing. Process manages process instances, providing the services required to route them between activities and to distribute work throughout the organization.

Process Designer, which is based on Microsoft® Visio®, is a graphical tool that designers use to build structured processes. Each process defines the actions that are performed on objects and enforces the rules required to support those actions. Multiple processes can be run on a Case360 Server, and designers can create multiple versions of a process.

Process can be used in conjunction with OpenText managerView, a product suite that provides analytic reporting for Business Process Management (BPM) applications. To utilize managerView, you need to purchase and install managerView and configure managerView support on your server.

Content Management

Content Management capabilities are provided by Filestore. Filestore manages any kind of content used by Case360, including text and image files, word processing documents, spreadsheets, scanned or faxed documents, and other file types. Filestore functions include storage and retrieval, versioning, check-in, and check-out. Extended support is provided for TIFF image files.

Filestore also provides extensive storage management features including mirroring (replication), rollover, cost based retrieval, and an open interface to add new storage devices. Filestore includes support for magnetic, database BLOB, optical storage, EMC® Centera™ storage, and NetApp® SnapLock™. Storage is licensed separately.

Enterprise Document Access Connectors

Enterprise Document Access Connectors are interfaces that provide Case360 users access to objects in other OpenText repositories or in external repositories that are managed by another product. An Enterprise Document Access Connector consists of a Java class installed on the Case360 Server and a Web user interface for presenting the objects contained in the repository. Enterprise Document Access Connector implementations retrieve content data as binary objects that Case360 can present to client applications through its Web service APIs. No access to the external repository's client application is required.

OpenText offers Enterprise Document Access Connectors that provide access to other OpenText and external repositories. A license to use these Enterprise Document Access Connectors must be purchased separately. Software developers can create additional Enterprise Document Access Connectors by using the external repository base class code included with Case360 and following the repository interface definition.

Building Applications

Designers build Case360 applications using a combination of Case360 tools, Hypertext Markup Language (HTML) customization, Java Server Page (JSP) design, and Java programming. Case360 provides the tools for creating database tables and queries as well as for designing casefolders, processes, forms, and documents. An extensive programming interface and a powerful scripting language enable designers to add advanced features and modify the way the core functions work.

Application Development Tools

The primary application development tool in Case360 is the Toolbox, which is enabled for users who have System Administration privileges. The Toolbox provides developers with the capability to define repositories, database fields, Casefolder templates, Form Data templates, Filestore templates, scripts, queries, menus, and Capture packages and to define security. Administrators use the Toolbox to manage users and the system.

The Visio-based Process Designer tool enables developers to create the process maps. For many types of processes, no additional programming is required in order to build sophisticated process applications.

Layout Designer

A Web-based drag and drop tool called Layout Designer can be used to define custom forms to be presented to the user for use with any repository. Layout Designer is intended for use by designers who do not have any coding skills. Further, it does not expect designers to have any knowledge of HTML, CSS, JavaScript, or Ajax.

Layout Designer generates a JSP that is used to render the form. An application designer can edit the generated JSP to add custom behaviors. Once the JSP has been modified manually, the layout designer can no longer be used to edit the layout.

Programming Interface

The Case360 programming interface supports additional application development and integration functions. The toolkit includes both Java and Web Services Application Programming Interfaces (APIs) as well as servlets and Sun EJB technology that can be used to build a customized user interface, supporting functions, and a base set of queries. The Case360 Web Services API adapts very well to Service Oriented Architectures (SOA).

Using the APIs provided, data from existing business applications can easily be integrated into applications without disrupting normal business operations. All of the APIs used to create the user interface servlets are exposed for application development.

All functions are exposed through both Java and Web Services APIs.

- A complete set of Java-based APIs provides the capability to extend the system through event handlers, program activities, robots, and customized servlets.
- The Web Services interface is designed to expose a complete service-oriented API for interacting with case processing and content management capabilities. The Web

Services APIs provide access to Case360 capabilities using industry accepted standards and communication protocols that define Web services.

Application designers can

- Create new applications.
- Extend the capabilities of existing applications by adding Case360 functions to them.
- Access data in legacy applications and integrate the data into Case360 applications.

Licensing

Case360 incorporates a number of components, each of which is licensed separately.

To get information about special considerations for high availability environments, contact your OpenText Representative.

To request a software authorization key, visit the OpenText Knowledge Center.

Casefolder, Process, Forms, and Filestore

OpenText licenses Casefolder, Process, and Filestore separately and on a concurrent user basis. It controls the number of concurrent accesses to the Case360 Server for each Case360 component - Casefolder, Process, and Filestore. Using a Form Data form is the same as using a casefolder.

The Case360 license tool enables the system administrator to define a set of named concurrent user groups. Each group can be assigned any number of Casefolder, Process, and Filestore licenses, as long as the total across all groups does not exceed the total number of purchased licenses (the server’s license management tool provides a software authorization key for this purpose). The user administration tool lists license groups and shows the names of the users whose license usage is accounted for in each group.

One Process license is automatically granted to enable administrators to use Process for life cycle functions.

One Filestore license is automatically granted to enable storage of process and application definitions.

As shipped, the Case360 package includes an interim license key that enables you to install, activate, and test the software. The key allows three concurrent users to access all the components (Casefolder, Process, and Filestore) for a period of 60 days.

Related Products

OpenText also licenses the following products, complementary to Case360:

- Case360 Enterprise Document Access Connectors are licensed on a per instance basis.
- Scan Manager is licensed on a throughput basis.
- Data Managers are licensed on a capacity basis.
- analystView is licensed on a per-copy basis.
- managerView is available with certain Case360 bundles or as an optional per-seat and per-server add-on.

- Records Manager is licensed on a named user basis.
- Recognition Server is licensed on a per server basis.

OpenText offers additional complementary products. For a complete list, contact your OpenText Representative.

Chapter 2

Architecture

With a basic knowledge of HTML and Java and an understanding of the Case360 software design, repositories, and default user interface, an application designer can easily customize or extend the default interfaces and functions.

Case360 is implemented using the Sun Microsystems Enterprise JavaBeans (EJB) architecture, which provides a server-side component model for the Java programming language. EJB technology can be used to create reusable business logic and enterprise applications that can be deployed on any platform and operating system that is Java-compliant.

The Java development environment is extremely popular because it is robust, reliable, and proven. Java-based applications are highly scalable, they are transportable across different platforms, and they can be integrated with a wide range of existing applications and data repositories.

Case360 is installed on a Java 2 Enterprise Edition (J2EE) application server. The J2EE server provides many of the application’s underlying services, such as user authentication, database access, and file management, enabling application designers to focus their efforts on designing and implementing the business solution.

Case360 applications use popular Web browsers as the client interface. Web browsers are easily deployed, cost effective, and provide convenient data access across the Internet to employees, partners, clients, or vendors with the proper user authorization.

The Case360 scripting language provides extensive support for accessing and manipulating XML documents. An XML field type can be used to store an XML document using the database server’s XML column support. This support is based on the W3C standard object model, with a few extensions. The full W3C Document Object Model is supported natively by the expression package.

Software Design

The Case360 architecture follows the thin client model. With this model, the Web browser client is required to perform little, if any, data processing. Therefore, it is not necessary to install anything on hundreds of client machines.

Servers

A Case360 configuration consists of supporting servers that run application, database, and file management software. Server platform support is based on the supported application server and database. For example, servers can run under one of the following operating systems:

- Microsoft® Windows®
- Sun Solaris™
- IBM®AIX®
- HP-UX®
- Linux®

Using the same operating system on all servers simplifies system management and is recommended.

For the most current information about the supported versions of operating systems, servers, and database products, refer to the Product Information Matrix (PIM), which is available in the Support area of the OpenText Web site (www.opentext.com).

Application Server

Case360 software is installed on a J2EE-compliant application server running IBM WebSphere®, Oracle WebLogic®, or JBoss®.

The application server software acts as the Web server, handling the application logic and connectivity, operations between browser clients, and the enterprise’s back-end business applications and databases.

Case360 software manages JSP files, properties files, event handler interfaces, form files, and framesets that provide the default user interface and Case360 functions.

To simplify development, it is recommended that the same application server be used for development and production environments.

Database Server

The database server holds the Case360 database, performs database management functions, and processes requests from the application servers. Browser clients cannot directly access the database server.

Case360 uses the Java Database Connectivity (JDBC™) API to provide database-independent access to its databases. The following database products are currently supported:

- Microsoft SQL Server
- Oracle® Database
- IBM DB2

File Servers

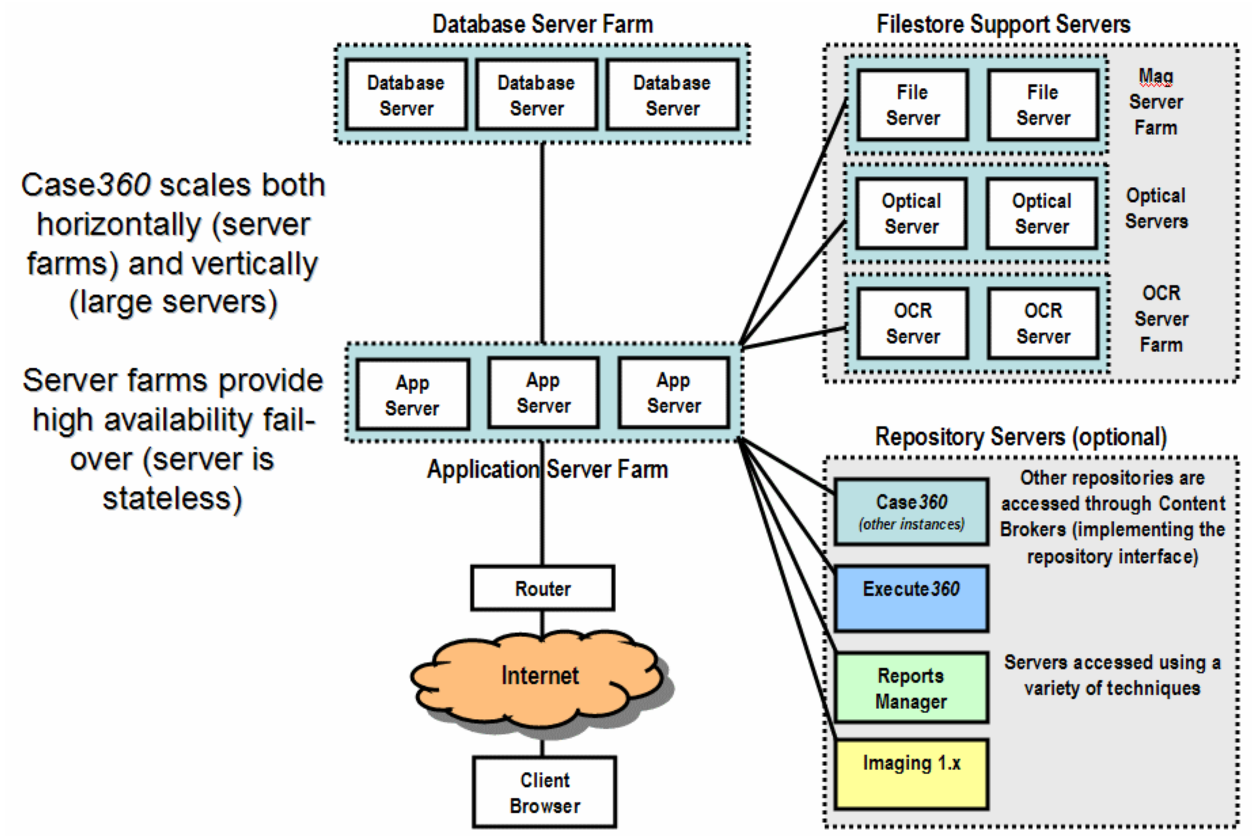
File servers provide file handling and storage functions for multiple users on the network. Filestore uses these servers to store the files being managed. Content can be stored on magnetic media, in database BLOBs, on EMC Centera servers, on NetApp SnapLock servers, or on optical disk. A data manager API is provided, enabling customers or third parties to create custom data managers to support additional physical storage mechanisms.

Configurations

There are many different ways to configure a Case360 system. For the lightest of business applications, or for a developer's system, all three servers can be installed on one computer. In a production environment, the application server is typically installed on one computer and the file and database servers on a different computer. A large installation may require installing each type of server on a different computer and can scale to multiple computers for each function, as illustrated below

Client browsers connect to Case360 through the application server. Since client browsers cannot directly access the database server or the file server, clients cannot "see" the data on these servers. This enables Case360 to maintain a high level of security.

The following diagram illustrates the use of multiple computers to support functions provided by the database, application, and file servers.


Database Server Farm
Database Server
Database Server
Database Server
Filestore Support Servers
File Server
File Server
Mag Server Farm
Optical Server
Optical Server
Optical Servers
OCR Server
OCR Server
OCR Server Farm
Case360 scales both horizontally (server farms) and vertically (large servers)
Server farms provide high availability fail-over (server is stateless)
App Server
App Server
App Server
Application Server Farm
Repository Servers (optional)
Case360 (other instances)
Other repositories are accessed through Content Brokers (implementing the repository interface)
Execute360
Reports Manager
Servers accessed using a variety of techniques
Imaging 1.x
Router
Internet
Client Browser

Expansion Considerations

There is no universal solution to scaling an installation to handle an increased workload. Depending on requirements of the business application, it may be necessary to use multiple application servers.

In smaller installations, you can increase performance by replacing older computers with computers that use faster processors or by replacing them with multi-processor computers, faster disk drives, and more memory.

You typically grow application server and file server capacity by adding more servers and grow database server capacity by using a larger server. You may want to analyze your application to determine which components can be executed on separate servers.

Dynamic Load Balancing

The dynamic load balancing mechanism is used to partition work between servers in a cluster (application server farm). The load balancing mechanism has the following features:

- There is no configuration required - servers self-register.
- Work is automatically redistributed if a server drops out of the cluster (work is not permanently assigned to a single server, and no database updates are needed to reallocate work).
- The processing capacity of each server can be adjusted to allow servers of different capability to work together in a cluster.

Work is shared between servers based on a key associated with each instance. Each server has a specific range of values for which it is responsible and it processes only those instances whose assigned number falls in that range. These ranges are computed automatically from the configured server capacities set in the Servers tool.

Multi-Server Clusters

It is common to configure a server using multiple application servers. This may be done to increase the capacity of the server to handle large loads. It can also be done to provide for fault-tolerance so that a server fault does not interrupt service.

If you are running individually managed servers accessing the same database, for example two WebSphere servers, you need only define the Configuration Tool options on one server. Because the options are written once to the database, other servers will inherit the options.

Database Partitioning

In most cases, a single server cluster is used to run all the applications for an enterprise. As demand grows, additional application servers can be added to the cluster to handle the load.

However, the database server can quickly become a performance bottleneck. It is important to spread the database load across multiple database servers. Rather than managing multiple independent databases, the server accesses a single logical database that is distributed across database servers using the partitioning feature provided by the database server.

Security

Case360 security is based on the separation of authentication from authorization. Authentication is the process by which users identify themselves to the system. Authorization is the process by which users are granted access to content.

Authentication

Vendors of operating systems and security systems offer a wide variety of authentication mechanisms. Their systems may include hardware and software components. In addition, many customers want to reduce costs by consolidating user management into a single directory mechanism.

Customers choose an authentication mechanism based on their application platform. For Java-based Web applications, authentication is provided by the J2EE application server, with which Case360 integrates. Case360 calls an API provided by the application server to get the trusted identity of a user.

Authorization

Case360 provides the mechanisms that authorize access to content created and maintained in its system. It does this because there are few standard authorization mechanisms and the ones that exist do not provide the flexibility required by Case360 customers.

Production process management systems often use variable information associated with specific users to make process decisions and to control access to content. For example, a user’s proficiency rating can determine how many of the user’s cases need to be audited.

Case360 manages this type of information through a user table that stores information such as each user’s logon ID, display name, and custom user properties. When a user logs on, the logon ID is used to access the user’s custom user properties from the database.

The logon ID is also used to determine the user’s roles. Instead of assigning permissions directly to a user, a user is assigned to one or more roles. A role is like a group - users are assigned to roles based on their job responsibilities. Specific permissions are granted to each role, so that every user assigned to the role inherits the corresponding set of permissions. An API is provided to enable an application designer to obtain this user information from an external directory such as a Lightweight Directory Access Protocol (LDAP) server.

The Case360 mechanism uses access control lists (ACLs) to authorize access based on user membership in application-defined roles. This mechanism controls user access to elements - or portions - of an object. Each ACL lists the roles that will have access to content controlled by that ACL. Each role in an ACL lists the permissions that members of that role will have to objects controlled by the ACL.

Every Case360 element, such as repository objects; queries; files; fields used in form definitions; tasks, content, and discussions in Casefolders; and process versions can have an associated ACL. This provides object-level security to the application.

Permissions in an ACL can have associated expressions that specify the conditions under which each permission will be granted. This enables security rules to be defined that can use any of the protected object’s properties as well as the current user’s properties. Rules-based security is especially useful for applications that have very specific security requirements that cannot be fulfilled with simple yes/no permissions.

The following list gives a few examples:

- A user has a custom field that defines the maximum dollar value of the cases the user is allowed to access.
- Users should be allowed to modify a document for only 30 days after it was created.

- A task cannot be modified until another task in the same case is completed.
- Only the user to whom a casefolder is assigned is allowed to modify it.

Java Messaging Service (JMS)

Java Messaging Service is the J2EE standard for supporting reliable messaging for a range of providers (messaging services). JMS provides a standard mechanism for isolating systems running different platforms and distributing components of complex architectures, handling the message delivery and fault tolerance within the provider. JMS supports two main modes of operation: point-to-point and publish-subscribe.

Case360 acts as a consumer and producer of JMS messages. As a producer, messages can be sent containing configurable text, field and XML data, from scripts or a process activity. As a consumer, messages are received through the Message Input driver, enabling the receipt of text and binary/document information.

As a consumer of JMS messages, Capture scripts can perform various operations depending on the application’s requirements. Some examples follow:

- Store the message as a form.
- Attach the form to a casefolder.
- Create a process instance and run it in a single thread.
- Utilize immediate processing to cause the server to evaluate the process rules while the caller waits.
- Return a response to the message producer.

SOA Functionality

Case360 support for Service Oriented Architecture (SOA) and Straight-Through Processing (STP) is useful for advanced human-centric business applications. This functionality enables organizations to meet the growing requirement to host/publish and consume/reuse business services for exposing content and process data across the organization, outside the firewall, and for Business to Business (B2B) transactions.

The following use cases demonstrate how Case360 functionality provides the capability to deliver services that can be exposed across the organization and to trading partners.

Insurance Claims Service

An example insurance company has hosted an insurance claims process on Case360. Insurance claims are generated by claim agents outside the organization who work directly with the customer.

When a customer contacts a claims agent to submit a claim, the agent captures the customer claim information within the agent’s own system and assesses coverage based on the information recorded locally.

XML messaging - When the agent has completely entered all of the claim information, the system sends it to the insurance company. This is performed by preparing an ACORD specification XML document (ACORD is a global standards body that provides messaging standards for the insurance industry).

Service host - The claim XML is sent to a JMS service on a secure extranet that the insurance company makes available to its agents. The JMS service is hosted by Case360 and provides an endpoint for many different agent requests.

Message capture and XML storage - The delivered JMS message containing the claimXML is immediately picked up by Case360 through Capture using the Message Input driver. Conditional expressions parse the XML document in the JMS message to identify the type of transaction being requested and trigger the appropriate script to be run. The appropriate capture script for claims transactions creates a process instance for the claim and stores the claim XML to an XML field in the process instance.

Since the claim process is expected to take an extended period of time to assess, the activity is not set for immediate (real-time) processing. On creation of a process instance, the Capture Script continues, sending an informational response to the claims agent. The Capture Script completes, triggering the JMS response to be sent. The agent receives a response that the claim is pending processing within seconds.

Standard application - The claim runs through the extended human-driven process, managed by Case360. Standard Filestore, Casefolder, and Process capabilities are used.

XML parsing - At every step in the process the stored claim XML is parsed to present claim information to end-users and to provide automated logic for decision activities. At the end of processing, the claim information must be passed to the claims system of record to trigger payment of the claim.

Consume service - The system of record is a mainframe application that has a simple Web service API exposed. The Web service activity in the Case360 process sends the claim details to this Web service before moving on to complete its process.

XML creation - The Case360 process creates a new XML document and fills specific elements with information about the success of the claim process and information returned from the system of record.

XML messaging - Case360 takes the XML response message and sends it through JMS to the agent system, informing the agent of the success of the claim and the pending payment.

Immediate Processing

The insurance claims service in the previous example processes insurance claims from a remote insurance agent. To enable agents and centralized customer services representatives to respond to customer inquiries about the status of their claims, the Case360 claims service is extended to provide real-time inquiries.

A remote agent system or the customer service portal presents a form enabling the agent to enter identifying information about the customer or claim number. This information is used to create an ACORD XML message that represents the query transaction to be performed. The requesting system sends this to the Case360 Insurance Claims Service through a JMS message. In processing the message through its Capture Broker, the Case360 application identifies that this is a real-time inquiry and creates a process instance for immediate processing.

In immediate processing, the process instance is created and runs through the process while the requestor waits for a response. The Capture Script that creates the process instance waits until the process completes.

The inquiry process performs several steps. It queries the Case360 claims process to look for a pending claim; it checks to see whether the claim requires additional information or documents; if necessary, it queries the system of record to see if the claim has been paid.

When all queries are complete according to the process logic, the inquiry process returns with a result that is passed to the original requestor as an XML document. The contents include information about the inquiry made by the requestor.

While the claims process is running, standard rendezvous capabilities can be utilized to reconcile new documents and service requests with the ongoing process. In this way, new documents, information and service requests can be attached to a process to affect its processing through subsequent decisions or to move it from a pending state if more information is required.

Persona-Based Interface

The default Case360 user interface is constructed using a set of Java servlets tied together through HTML framesets. Case360 also comes with a user experience (UX) tag library that provides a simple and efficient method for building a customized user interface for one or more users by adding just a few lines of code with no pure HTML tags inside a page.

Case360 persona-based home pages that use the persona-based repository framesets can help an organization meet the specific needs of the various types of users who participate in a process by providing them with unique views that include only the information they need to get their jobs done. By enabling people to do their work better, applications built using persona-based Case360 framesets deliver increased productivity gains and eliminate inefficiencies and costs from a process.

Persona-based home pages and framesets or applications using persona-based framesets

- **Are intuitive** - which makes it easier to do work - if something is easier to do, users are more likely to do it. Clear communication between the user and the system also increases accuracy. And that ensures that tasks are completed more efficiently.
- **Provide a comfortable and familiar environment** - Understanding how to use something with less training lets users get up to speed faster and work more intuitively. And if users willingly do something more often, they have a greater chance of liking it. If they like doing it they will be more comfortable doing it.
- **Make work more efficient** - Persona-based BPM looks at the repetition, order, synergy, value-add and necessity of work done by a process participant. By understanding what tasks are performed most often, those activities can be made easier to perform and more accessible.

Persona-based Home Pages and Framesets

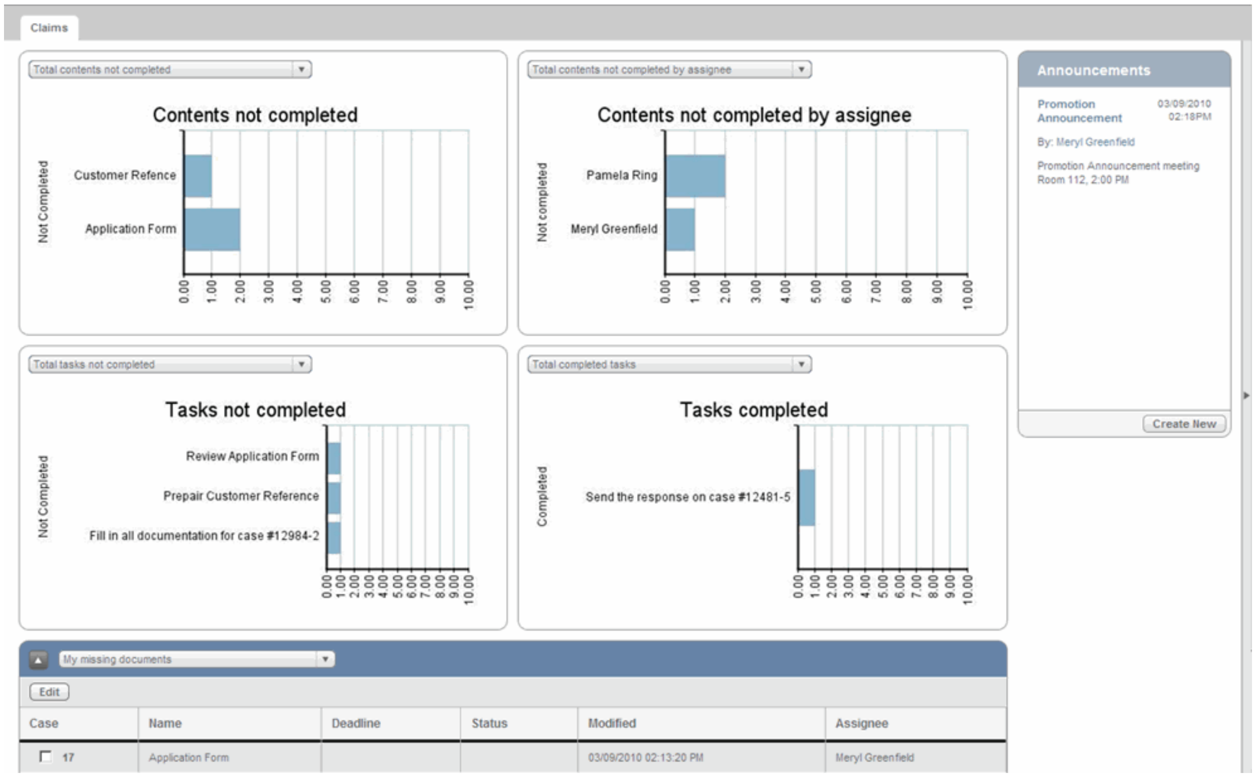
The following persona-based home pages are provided with Case360: Data Entry, Processor, Researcher, Supervisor, and Manager. An organization can use these pages as is or can adapt them to individual preferences. The following list describes some of the features provided on persona-based home pages:

- Breadcrumbs enable users to return to recently visited pages.
- Announcements offer a convenient method for notifying selected users about a subject of interest.
- A slider control provides a simple way for users to update their status for management reporting.

Case360 provides a set of themes that a server administrator can use as the basis for quickly and easily creating a customized home page for a persona. Each theme comes with specific graphics and styles for page decoration. Custom theme creation does not require any code changes in jsp files. All customization is done only by changing the styles in .css files. The organization of styles, images, and client-side javascripts makes the theme customization process fast and easy.

Persona-based home pages and applications built using the persona-based pages boost the efficiency of existing behaviors and empower process participants to adopt new ones - truly changing how work gets done.

The following illustration shows an example of the Manager home page.



Promotion Announcement 03/09/2010 02:18PM
By: Meryl Greenfield
Promotion Announcement meeting Room 112, 2:00 PM

Case	Name	Deadline	Status	Modified	Assignee
17	Application Form			03/09/2010 02:13:20 PM	Meryl Greenfield

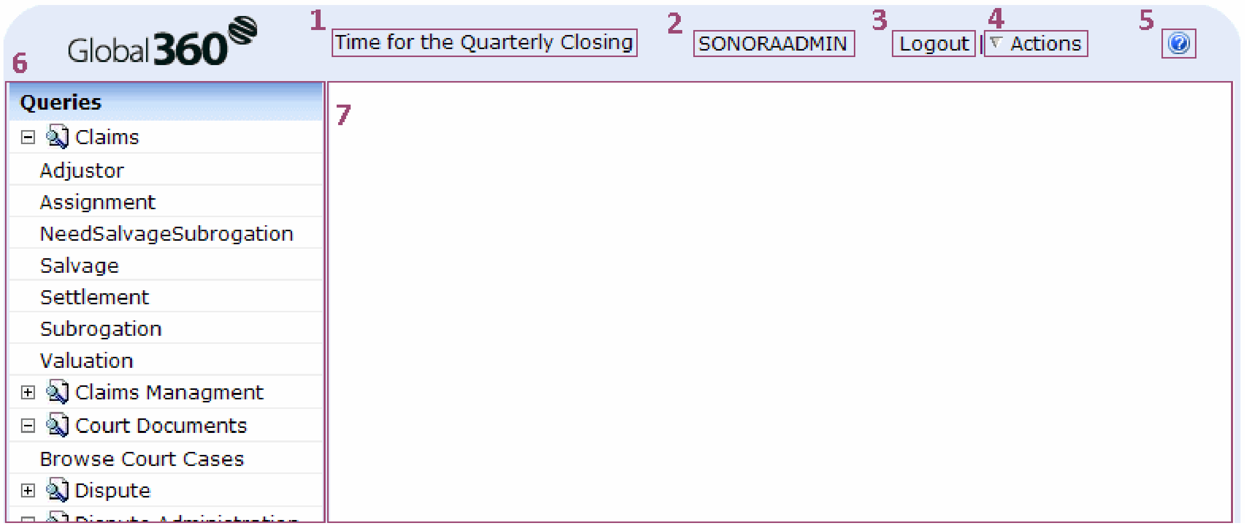
Legacy Interface

A user who is not assigned a persona uses one of the three legacy home pages. The Query servlet lists the queries in the left frame of the legacy Home page. The Actions menu enables authorized users to create and manage objects in supported repositories. The

Toolbox option that appears on the Actions menu for the default legacy page enables authorized users to administer the system and design the application.

Home pages can be customized so, if you use the Legacy home page, it may not look exactly like the following example. However, many of the elements on a customized home page will be the same as those on the default Legacy home page. The following list describes the significant elements on the Legacy home page (the numbers correspond to those in the example).

1. **Home page message** - displays a message that is set by the system administrator using the configuration tool.
2. **Current user** - displays the name of the user who is currently logged on.
3. **Logout** - logs off the current user and releases the corresponding license.
4. **Actions menu** - lists the objects that the user can create and provides a link to the toolbox.
5. **Help** - displays the Case360 help system.
6. **Query list** - lists the queries that are available to the user.
7. **Query results** - presents the results of the queries.



Applets

Case360 includes the following applets:

Upload Applet - Copies a file from a local computer to the file server.

Download Applet - Copies a file from the file server to a local computer.

Imaging for the Web Applet - Enables users to view, edit, and annotate TIFF files. This applet was formerly called the Web Image Editor (WIE) Applet.

Print Applet - Enables printing of documents.

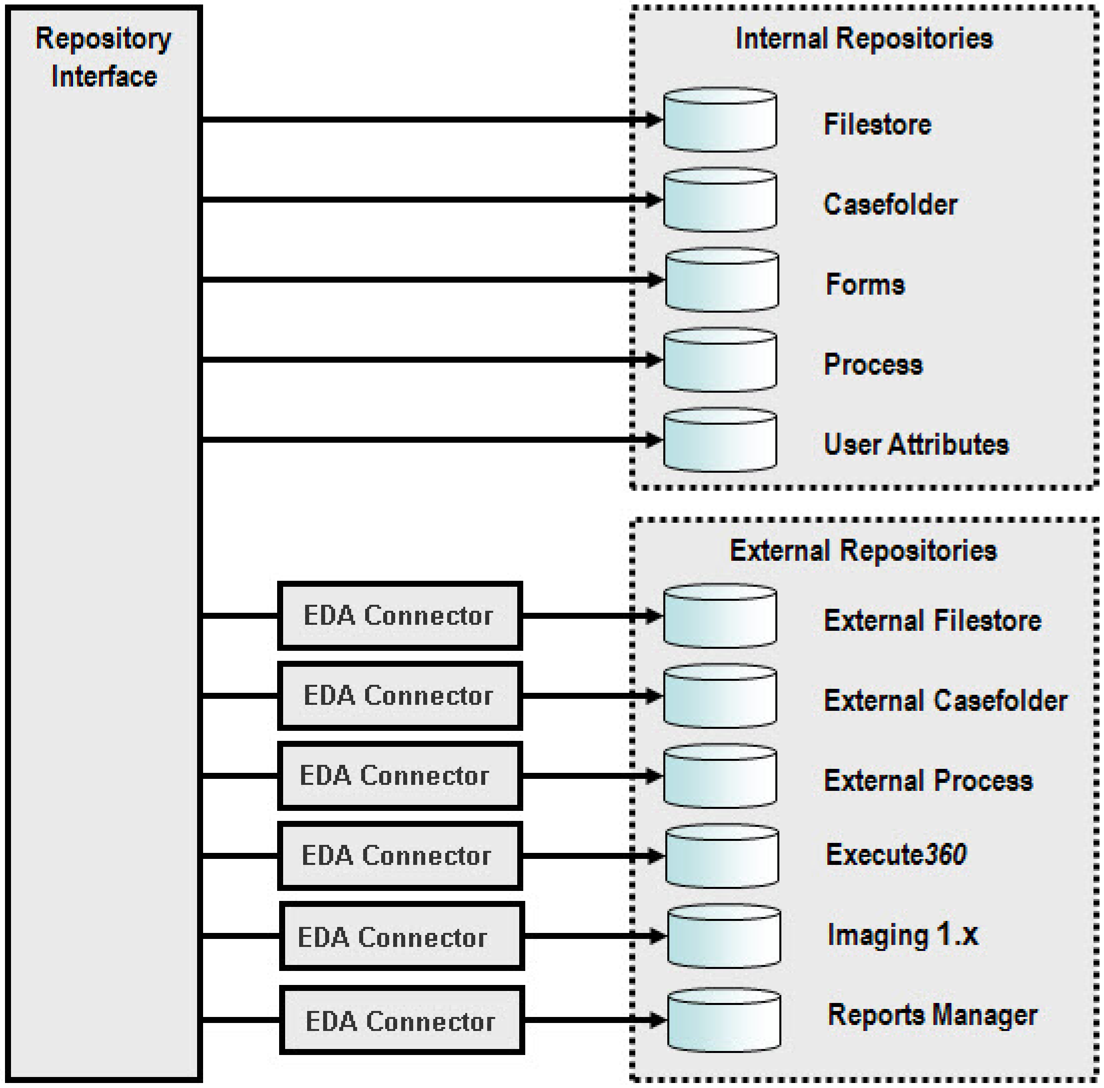
Filestore uses the Upload, Download, and Imaging for the Web applets to manage its objects.

These applets are designed to enhance the standard Web user interface but they can be disabled. If the applets are disabled, Case360 reverts to the default user interfaces.

Chapter 3

Repositories

A repository is a location where objects are stored and it includes the mechanism required to manage those objects. A single repository can manage multiple instances of many object types. Case360 supports both internal and external repositories.



Internal Repositories

Case360 includes five internal repositories - Filestore, Casefolder, Process, Form Data, and User Attributes.

External Repositories

Many customer applications use existing repositories. The information contained within these repositories represents a substantial investment and is a significant asset for any business. To access these repositories, Case360 uses an Enterprise Document Access Connector interface.

An Enterprise Document Access Connector comprises two components - a Java class that supports the Case360 repository interface definition and a Web user interface for presenting the objects stored in the repository. The Java class is installed on the Case360 Server.

OpenText offers the following Enterprise Document Access Connectors:

- Enterprise Document Access Connector for Case360 Casefolder
- Enterprise Document Access Connector for Case360 Filestore
- Enterprise Document Access Connector for Case360 Process
- InfoFusion Enterprise Document Access Connector for EMC Documentum
- Enterprise Document Access Connector for Execute360
- Enterprise Document Access Connector for IBM DB2 Content Manager
- InfoFusion Enterprise Document Access Connector for IBM Filenet P8
- Enterprise Document Access Connector for Imaging Server 1.x
- Enterprise Document Access Connector for Microsoft SharePoint Server
- Enterprise Document Access Connector for OpenText Content Server
- Enterprise Document Access Connector for Reports Manager

Enterprise Document Access Connectors can be written by software developers using the external repository base class code included with Case360. Enterprise Document Access Connectors can also be provided by third party vendors.

Access To Repository Objects

Users typically access data in repositories through the query mechanism (which is common to all the repositories). Queries find objects based on field values associated with each object. You can create queries that run against internal and external repositories. Users select and run queries as needed.

The Repository Model

Case360 defines a simple repository model in order to support as many existing repositories as possible. The basic elements are summarized in the following list:

- Each object in a repository must be identified by a unique ID that contains the information the repository needs to locate the object.

- The repository must be able to provide the list of object types it manages.
- Each object type has a set of fields that can be used to search for objects of that type.
- The repository is responsible for providing the Web user interface that is used to manipulate its objects.
- A Case360 ACL controls access to a repository as a whole. The repository controls any other type of security.

When applications make changes across multiple internal repositories, Case360 provides transactional integrity. It does not provide transactional integrity for external repositories.

Repository Schema

A schema describes the structure and organization of a data storage mechanism. In the case of Case360 repositories, the schema contains all the field information and form definitions used by a Case360 application. The Case360 Field Management System (FMS) tools enable application designers to manage and edit the schema.

A field is the part of a record that holds a certain type of data. Application designers identify the fields their application will require, create the fields (if necessary), and define them. This information becomes part of the schema. When creating forms, they select the fields to be included on the form from the schema.

Fields with the same name must have the same definition across all the internal repositories. This enables designers to reuse the same field on different forms. Reusing fields is important because it enables the query mechanism to identify shared fields when running a query against more than one object type. For example, if a user runs a query against two object types, where both include a field named Amount, there will only be one Amount column in the query results.

If your application uses an external repository, the Enterprise Document Access Connector creates the fields and forms for it.

Case360 supports several types of fields - Boolean, Decimal, Integer, Text, Long Text, Date, XML, and Repository Key. Each type of field has a different set of attributes that define its behavior. Attributes can be items such as name, label, description, length, format, and so on.

Repository Key Fields

A repository key field manages a reference, or pointer, to a repository object.

Repository objects are typically "attached" to process instances through repository key fields. For example, if an application is using Process to route casefolders to activities, a process instance will include a repository key field that contains a pointer to the Casefolder object that is to be routed.

Application designers can also use repository key fields to "attach" repository objects to Filestore, Casefolder, and Process objects.

Related Data

Table definitions can include relations as well as fields. A relation uses a query to retrieve a set of rows from another table that is related to the parent table. This enables representation of much more complex data models without the need to write custom code.

Forms

A form defines the collection of fields associated with an object. In the internal repositories, the form accesses the underlying database table that stores field values. Forms are used to

- Organize fields.
- Present fields to users for modification.
- Provide a framework for building queries.
- Prompt users for parameters when running a query.
- Define a set of custom fields for a user or role.

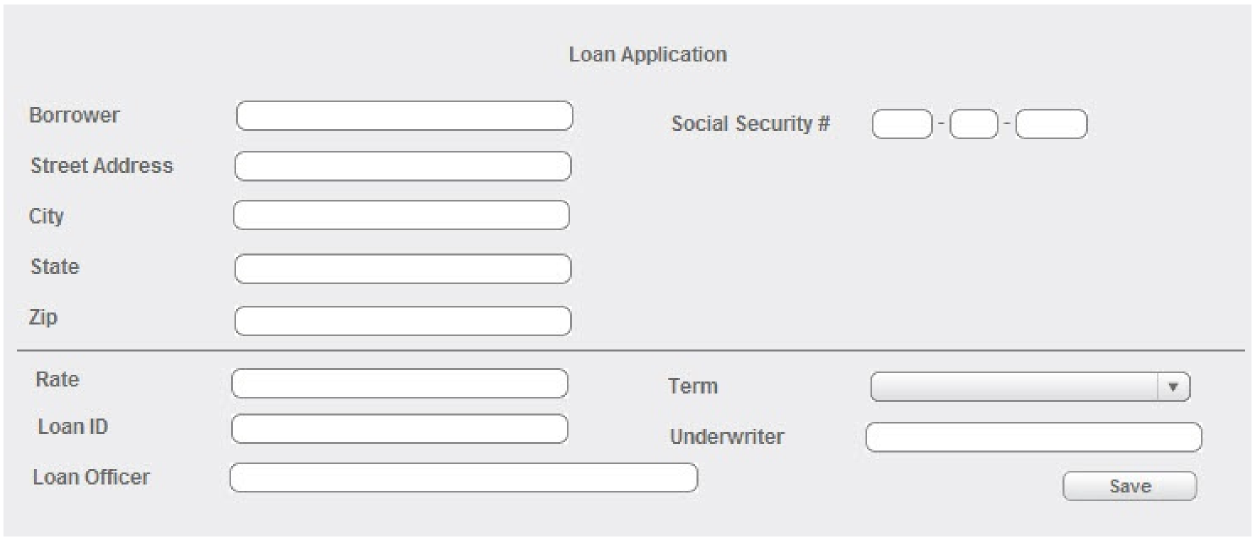
The default forms provided by the server display fields as a single column of input controls that are sorted alphabetically by field name.

Form objects - objects whose entire content consists of structured data - are stored in a form data repository.

Field formatting for display is handled by the Formatter user interface. When defining a field, the application designer specifies how that field should appear in a form by selecting a formatter from a list of formatters for that field’s type. This enables the designer to specify exactly how that field is to appear. Application designers can also create their own field formatters.

Application designers can create custom forms that present the fields in a way best suited to the application and the users. Custom forms can include any desired HTML content, including active elements such as hyperlinks and JavaScript. Forms can be customized using the Layout Designer. When a layout is saved, a JSP is automatically created and is used as part of the table definition.

The following is an example of a custom form created for a Casefolder:



Conflict Management and Resolution

Since Case360 supports collaborative processing, more than one person may be able to modify the same form at any given time. When a form is saved, the server software checks for conflicting changes. If there are no conflicts, both users' changes are merged and saved. However, when a conflict is detected, the second user to save the form may be presented with a conflict resolution dialog box asking how to resolve the conflict, depending on the conflict level assigned to fields on the form.

Case360 provides several levels of conflict detection, including

- Strict
- Conservative
- Relaxed
- No conflict detection

Conflict detection is applied on a field-by-field basis. Each field in a form has a conflict level indicator that determines how conflicts are handled.

Chapter 4

Casefolder

Casefolder is an internal repository that lets users create and manage casefolders - electronic folders, analogous to manila folders - that contain documents and other supporting data. Casefolder provides a way to keep a running history of changes made to the objects in its repository, track the status of tasks, manage and assign work, and use deadlines to keep items on schedule. It can provide an integrated view of data managed by diverse applications.

Casefolders can contain references to objects in any repository, including

- Filestore documents (text, image files, and so forth).
- Other casefolders.
- Process instances.
- Objects in any external repository.

Casefolder uses templates that are defined by application designers to create new casefolders. Users typically access casefolders through the query mechanism (which is common to all the repositories). Other repositories can reference casefolders.

The Casefolder Dashboard is used by both system administrators and application designers to define various types of information about a particular type of casefolder.

The following list provides some examples of the types of information that can be defined for each type of casefolder:

- Custom presentations for the structured field information.
- A set of custom fields.
- Whether the casefolder can be archived after the business activity it supports has been completed.
- A set of default queries.
- Charts that enable the owner of the business process to quickly see how work is progressing.
- Whether activity on the casefolder is to be reported to managerView for sophisticated reporting and, if so, additional information to send along with the events.

It is easy to manage a project from a casefolder, for example by organizing the contents in a logical structure. Members of a work team can share the contents of casefolders, access them to read background documents, review tasks, deadlines, and status, and participate in threaded discussions about items in the case folder. With Casefolder, the designer can structure work to the internal organization of the company rather than having to restructure existing processes, although careful design of applications can help to restructure the process.

Application developers can easily customize the Casefolder interface. They can give different users different views of the same casefolder and integrate casefolders with existing applications.

Casefolder Templates

Application designers can define any number of Casefolder templates. These templates are used to create casefolder instances. These casefolders instances inherit the properties of the template used to create them. The templates can define the sections, placeholders, and files that appear in a casefolder as well as fields, properties, tasks, and deadlines. Responsibility for a casefolder, content element, or task can be assigned to an individual user or to a role.

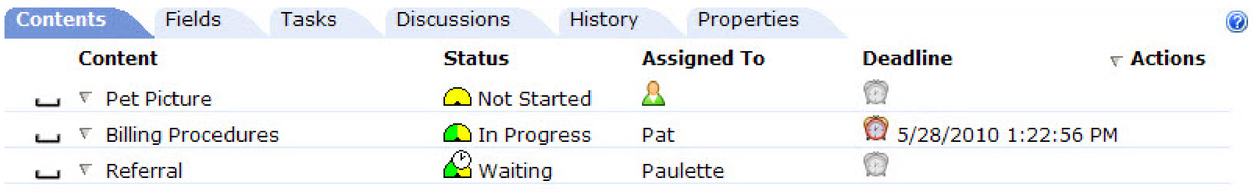
If a template is enabled with What's New? functionality, users see a What's New icon next to casefolder components that have changed since the last time they accessed the casefolder.

If a template is enabled with Subscription functionality, users are notified whenever one of the casefolders to which they subscribe changes.

The Case360 database keeps track of which template is used to create each casefolder. The database ensures that a user cannot delete a template if the template is still in use.

Contents

The contents of a casefolder consist of a list of filled and unfilled placeholders. Placeholders reserve spaces for objects that you expect to receive during the course of the project. Application designers can organize placeholders into sections that users can display or hide.



Content	Status	Assigned To	Deadline	Actions
Pet Picture	Not Started			
Billing Procedures	In Progress	Pat	5/28/2010 1:22:56 PM	
Referral	Waiting	Paulette		

The Contents function displays a list of the placeholders that are contained in the selected casefolder. It also displays the status of the placeholder, the user assigned to it, and any deadlines.

Placeholders can be created in a casefolder in a number of ways.

- The application designer can predefine a set of placeholders to appear in every casefolder created from a casefolder template.
- The application designer can add placeholders to the Actions menu. As content is required, a user can select that item from the menu and it will be created.
- A user (or a user in a role) with appropriate permissions can create placeholders spontaneously as needed.

Application designers can group placeholders in logically organized sections. Users with appropriate access rights can add, delete, rename, or reorder sections at any time, and add or delete placeholders within a section.

A casefolder actively manages the completion of its contents by tracking each document's status (Not Started, In Progress, Completed, Waiting, or Deferred). Placeholders are assigned to users who are responsible for the completion of the document (a placeholder is complete when its status is set to Completed).

Placeholders can have deadlines. If the placeholder is not completed when the deadline approaches, the assignee is sent an e-mail reminder. If the deadline is missed, the casefolder can send an alert to the assignee and/or the assignee's manager. A deadline can be a specific date and time or it can be relative to an event in the casefolder (for example, 5 business days after the application is received).

Placeholders can be filled interactively by users with the appropriate access permissions or automatically by an automated capture process. Placeholders can be filled by

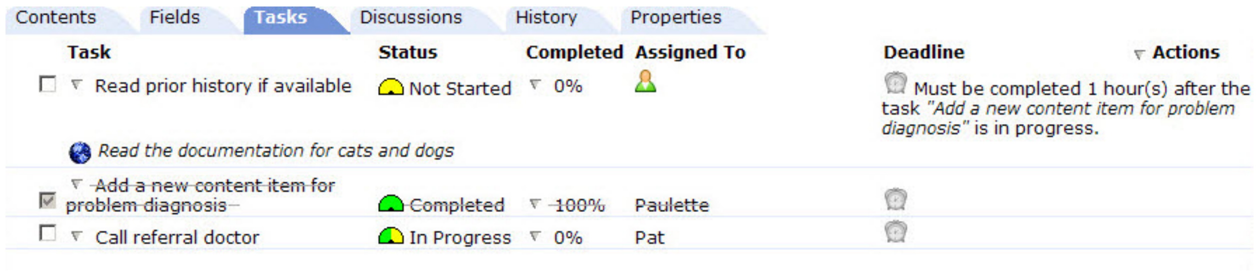
- Creating an object from a template.
- Running a query.
- Entering a URL.

Fields

The Fields function enables users to access and modify fields on the forms defined for a Casefolder. See the chapter in this document called "Form Data and Fields" for details.

Tasks

The Tasks function lists the tasks that need to be performed in order to complete a project.



Task	Status	Completed	Assigned To	Deadline	Actions
Read prior history if available	Not Started	0%		Must be completed 1 hour(s) after the task *"Add a new content item for problem diagnosis"* is in progress.	
Read the documentation for cats and dogs					
~~Add a new content item for problem diagnosis~~	~~Completed~~	~~100%~~	Paulette		
Call referral doctor	In Progress	0%	Pat		

Tasks can be created in a casefolder in a number of ways:

- The application designer can predefine a set of tasks to appear in every casefolder created from a casefolder template.
- The application designer can add tasks to the Actions menu. As a task is required, a user can select that item from the menu and it will be created.
- A user (or a user in a role) with appropriate permissions can create tasks spontaneously as needed.

Application designers can group tasks in logically organized sections. Users with appropriate access rights can add, delete, rename, or reorder sections at any time, and add or delete tasks within a section.

A casefolder actively manages the completion of its tasks by tracking the task's status (Not Started, In Progress, Completed, Waiting, or Deferred) as well as its overall progress. Tasks are assigned to users who are responsible for the completion of the task.

Tasks can have scheduled start dates and/or deadlines. If a task is not started by the scheduled start date, the assignee is sent a reminder. If the task is not completed as the deadline approaches, the assignee is sent an e-mail reminder. If the deadline is missed, the casefolder can send an alert to the assignee and/or the assignee's manager. A start date or a deadline can be a specific date and time or it can be relative to an event in the casefolder (for example, 5 business days after the application is received).

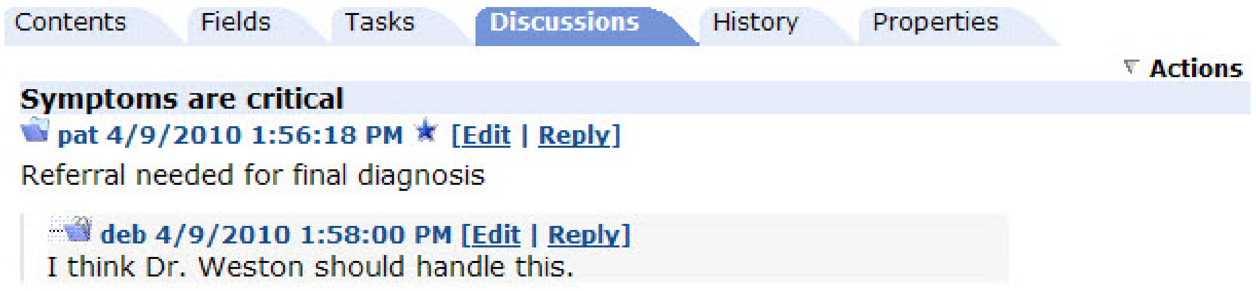
Discussions

The Discussions function provides a threaded discussion forum where users can communicate about a particular casefolder. A topic can be attached to a case, a placeholder, or a task. Each topic has an ACL that controls access to the whole topic.

Unlike ad hoc e-mail, or an independent discussions service, casefolder discussions are tied to an individual casefolder and are saved as part of that case's record.

An application designer can predefine a set of discussion topics that users can then comment upon. This can be used for instructional purposes or for common discussions related to a case, content, or task.

The Discussions tab displays the list of discussion topics, the author, and the date.



Symptoms are critical

pat 4/9/2010 1:56:18 PM [Edit | Reply]

Referral needed for final diagnosis

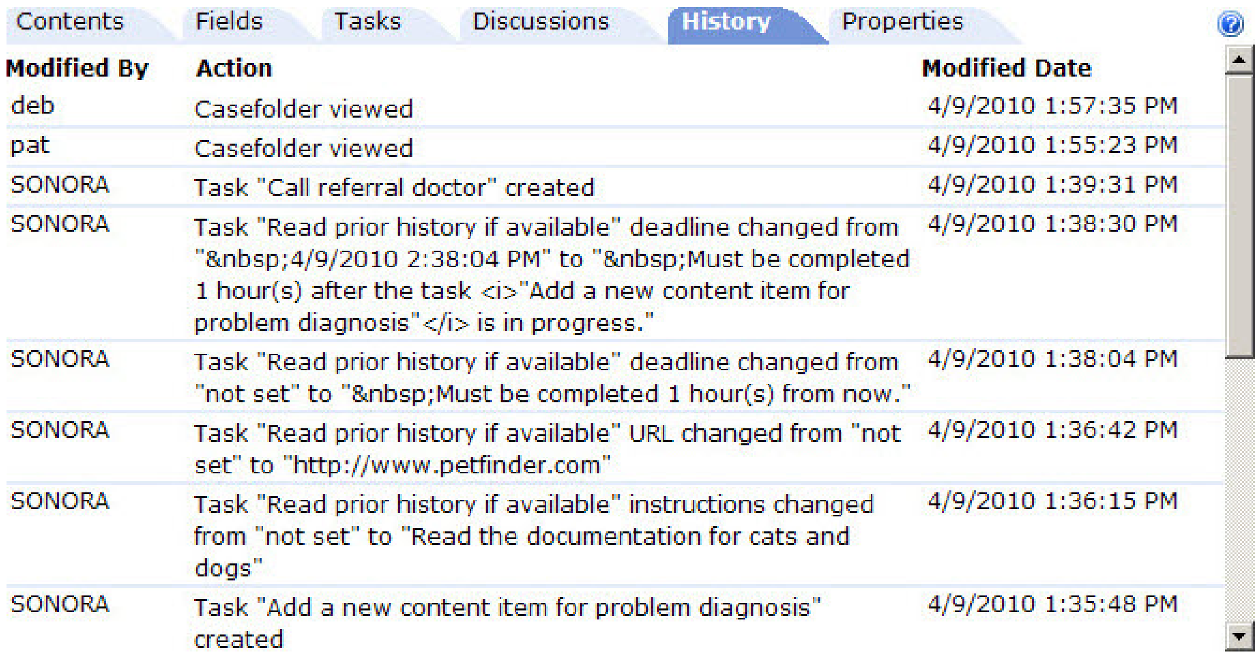
deb 4/9/2010 1:58:00 PM [Edit | Reply]

I think Dr. Weston should handle this.

History

The History function tracks activity that occurs on the selected object. It identifies the user who made the change, the date the change was made, and the action that was performed on the object.

If desired, the History function can also maintain a record of every user who opened the casefolder for read access.

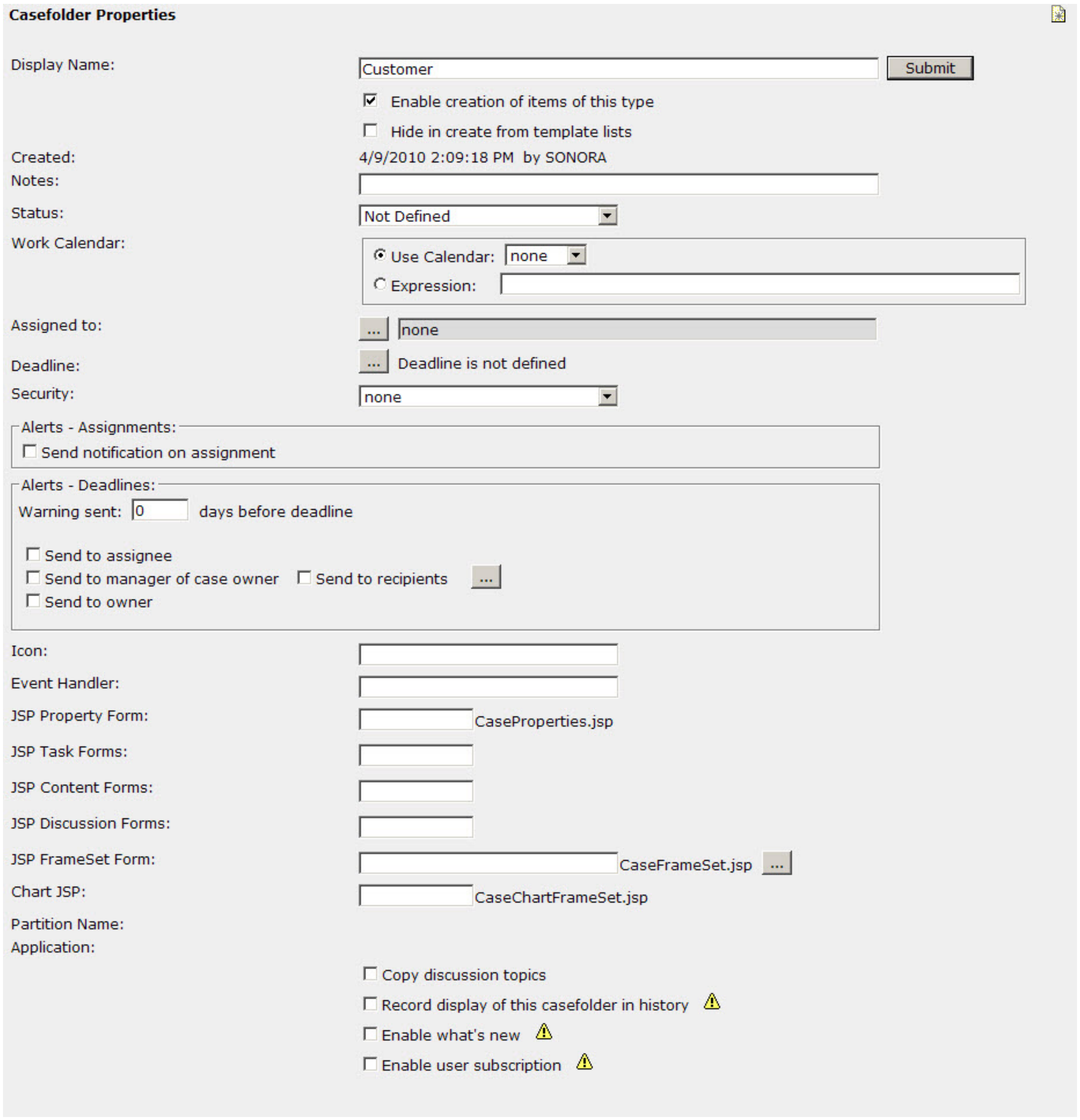


Modified By	Action	Modified Date
deb	Casefolder viewed	4/9/2010 1:57:35 PM
pat	Casefolder viewed	4/9/2010 1:55:23 PM
SONORA	Task "Call referral doctor" created	4/9/2010 1:39:31 PM
SONORA	Task "Read prior history if available" deadline changed from " 4/9/2010 2:38:04 PM" to " Must be completed 1 hour(s) after the task <i>"Add a new content item for problem diagnosis"</i> is in progress."	4/9/2010 1:38:30 PM
SONORA	Task "Read prior history if available" deadline changed from "not set" to " Must be completed 1 hour(s) from now."	4/9/2010 1:38:04 PM
SONORA	Task "Read prior history if available" URL changed from "not set" to "http://www.petfinder.com"	4/9/2010 1:36:42 PM
SONORA	Task "Read prior history if available" instructions changed from "not set" to "Read the documentation for cats and dogs"	4/9/2010 1:36:15 PM
SONORA	Task "Add a new content item for problem diagnosis" created	4/9/2010 1:35:48 PM

Properties

The Properties function displays information about the casefolder, such as the template’s name, creation date, and security. Application designers see additional information when editing templates.

An application designer can select which prefixed CaseFrameSet.jsp will be used to display casefolder instances as well as define which JSP property, task, content, and discussion forms will be used when an instance is created. A number of CaseFrameSet.jsp samples are available from the Properties page by accessing the frameset chooser; this includes a Persona-based Casefolder Frameset.



Casefolder Properties

Display Name:	Customer
	[✓] Enable creation of items of this type
	☐ Hide in create from template lists
Created:	4/9/2010 2:09:18 PM by SONORA
Notes:	
Status:	Not Defined
Work Calendar:	● Use Calendar: none; ○ Expression:
Assigned to:	none
Deadline:	Deadline is not defined
Security:	none

Alerts - Assignments:
☐ Send notification on assignment

Alerts - Deadlines:
Warning sent: 0 days before deadline
☐ Send to assignee
☐ Send to manager of case owner ☐ Send to recipients
☐ Send to owner

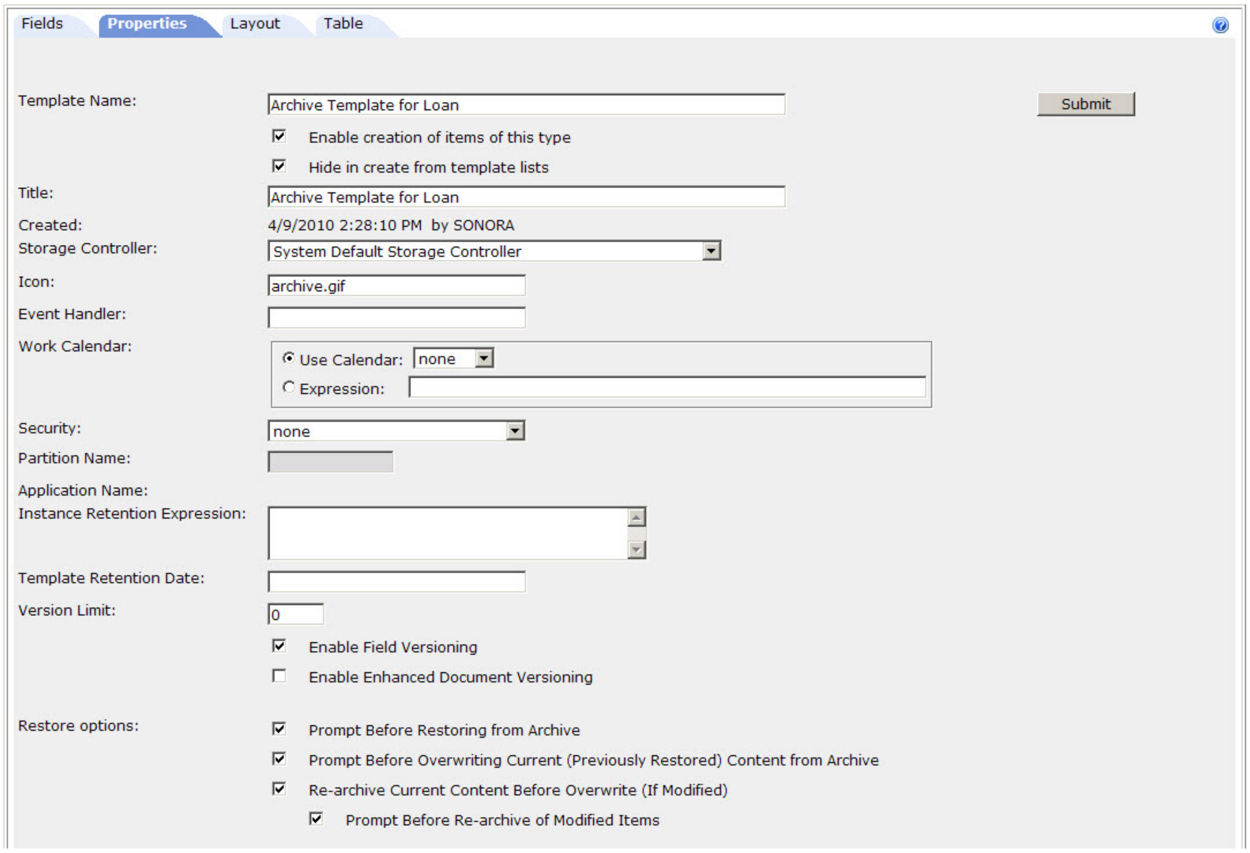
Icon:	
Event Handler:	
JSP Property Form:	CaseProperties.jsp
JSP Task Forms:	
JSP Content Forms:	
JSP Discussion Forms:	
JSP FrameSet Form:	CaseFrameSet.jsp
Chart JSP:	CaseChartFrameSet.jsp
Partition Name:	
Application:	

☐ Copy discussion topics
☐ Record display of this casefolder in history
☐ Enable what's new
☐ Enable user subscription

Archive

Application designers defining casefolder templates have access to an Archive function that enables them to define what happens when instances of the template are archived.

When a casefolder is archived, all the information maintained in the database is converted into a zipped XML file. That file is then stored using Filestore and the casefolder is deleted from the database. If needed, the archived casefolder can be restored to the database at a later date.



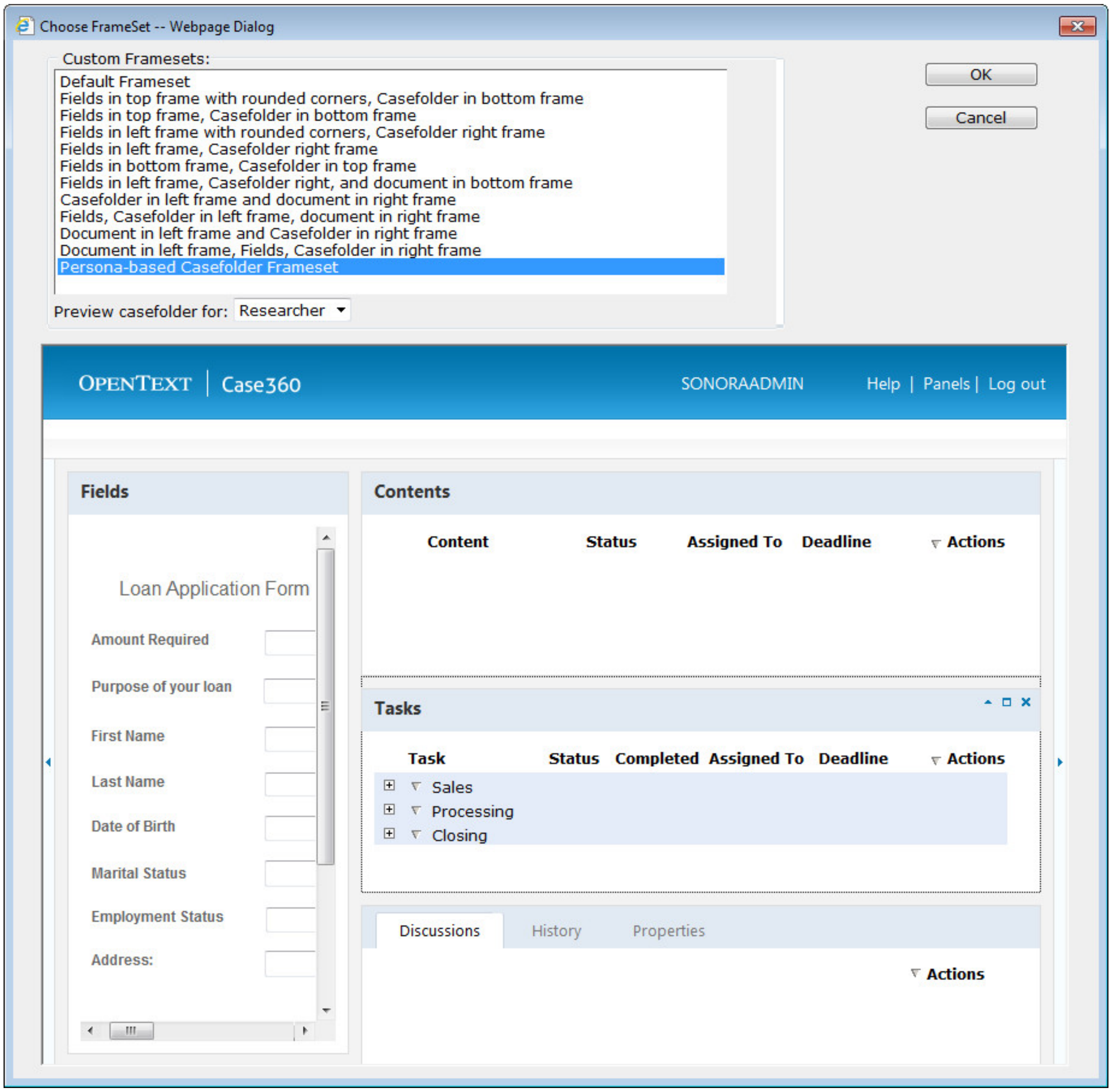
Template Name:	Archive Template for Loan
	☑ Enable creation of items of this type
	☑ Hide in create from template lists
Title:	Archive Template for Loan
Created:	4/9/2010 2:28:10 PM by SONORA
Storage Controller:	System Default Storage Controller
Icon:	archive.gif
Event Handler:	
Work Calendar:	● Use Calendar: none ○ Expression:
Security:	none
Partition Name:	
Application Name:	
Instance Retention Expression:	
Template Retention Date:	
Version Limit:	0
	☑ Enable Field Versioning
	☐ Enable Enhanced Document Versioning
Restore options:	☑ Prompt Before Restoring from Archive
	☑ Prompt Before Overwriting Current (Previously Restored) Content from Archive
	☑ Re-archive Current Content Before Overwrite (If Modified)
	☑ Prompt Before Re-archive of Modified Items

Customized Interfaces

An application designer can create customized layouts for different types of casefolders. For example, a customized layout may simply rearrange the casefolder components in a layout tailored for a specific application, perhaps using a portion of the casefolder’s window to display casefolder content.

Application designers can integrate existing applications into a frameset. For example, they can include a custom menu with a tab that presents an existing application inside the casefolder’s browser window.

Casefolder provides several frameset views. Application designers can select one of these predefined views or they can create their own customized framesets.



Custom Framesets:

- Default Frameset
- Fields in top frame with rounded corners, Casefolder in bottom frame
- Fields in top frame, Casefolder in bottom frame
- Fields in left frame with rounded corners, Casefolder right frame
- Fields in left frame, Casefolder right frame
- Fields in bottom frame, Casefolder in top frame
- Fields in left frame, Casefolder right, and document in bottom frame
- Casefolder in left frame and document in right frame
- Fields, Casefolder in left frame, document in right frame
- Document in left frame and Casefolder in right frame
- Document in left frame, Fields, Casefolder in right frame
- Persona-based Casefolder Frameset

Preview casefolder for: Researcher

Loan Application Form

Amount Required	
Purpose of your loan	
First Name	
Last Name	
Date of Birth	
Marital Status	
Employment Status	
Address:	

To facilitate integration with existing applications, the Casefolder user interface is generated on the server by a set of Java Server Page (JSP) files that are designed to be easily customized. There is a separate JSP for each of the pages that can be displayed when working with a casefolder.

Each type of casefolder can be associated with its own customized set of JSPs using properties on the casefolder. This makes it easy to deploy multiple applications on a single server.

Chapter 5

Form Data and Fields

The Form Data repository can be used to create forms that can be used either as standalone objects or in conjunction with other server components such as Casefolder, Process, and Filestore.

Forms can also be used to define tables to be included in an object's fields as related tables. This repository is called the Form Data repository because of this dual function. For simplicity, form templates and form instances used as standalone objects are often just referred to as Forms.

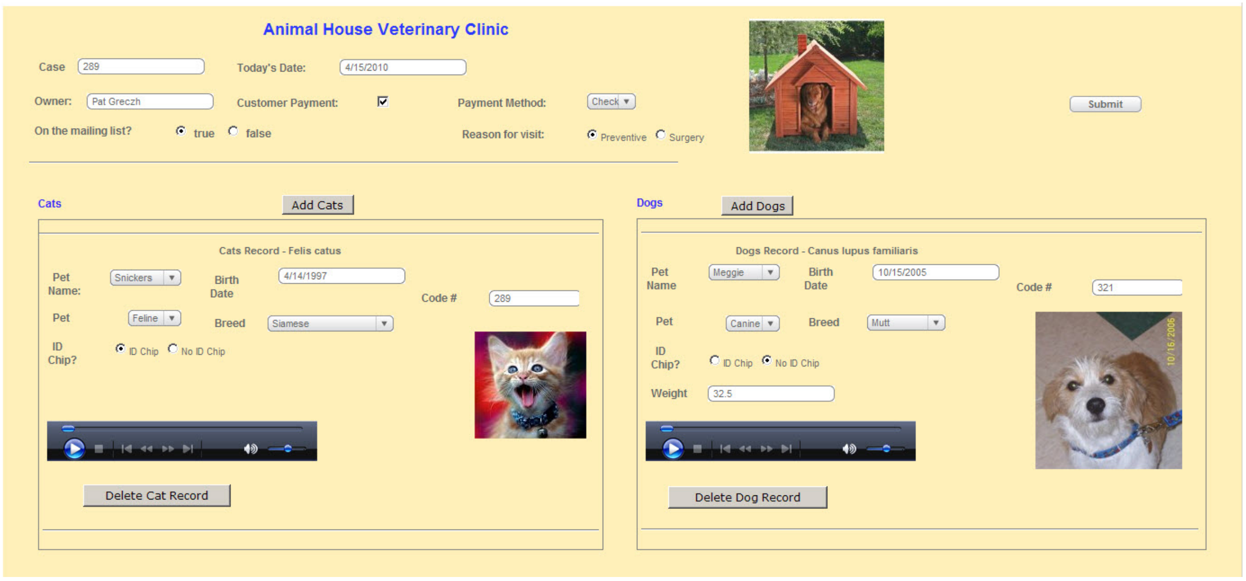
Accessing Form Instances

The way users access form instances depends on the design of the application and the user's permissions. A form instance can be accessed in several ways:

- Directly from a query result list.
- Indirectly from a casefolder content list by clicking the icon of a filled placeholder to access the form instance.
- Indirectly from a process instance that contains a repository key field that references a form.
- As a row of data embedded from a form in a related table.

Fields

Fields are used to store and track structured data associated with a casefolder, process, or document content. Enumerated values for fields can also be dynamically supplied by running a script. The Query mechanism uses fields to locate casefolders, processes, or content. The following illustration shows a Fields presentation using a layout. The fields are positioned as they would be for a customized application.



Animal House Veterinary Clinic

Case: 289 | Today's Date: 4/15/2010

Owner: Pat Greczh | Customer Payment: ☑ | Payment Method: Check

On the mailing list? [✓ true] false | Reason for visit: [✓ Preventive] Surgery

Submit

Cats — Add Cats

Cats Record - Felis catus

Pet Name: Snickers | Birth Date: 4/14/1997 | Code #: 289

Pet: Feline | Breed: Siamese

ID Chip? [✓ ID Chip] No ID Chip

Delete Cat Record

Dogs — Add Dogs

Dogs Record - Canus lupus familiaris

Pet Name: Meggie | Birth Date: 10/15/2005 | Code #: 321

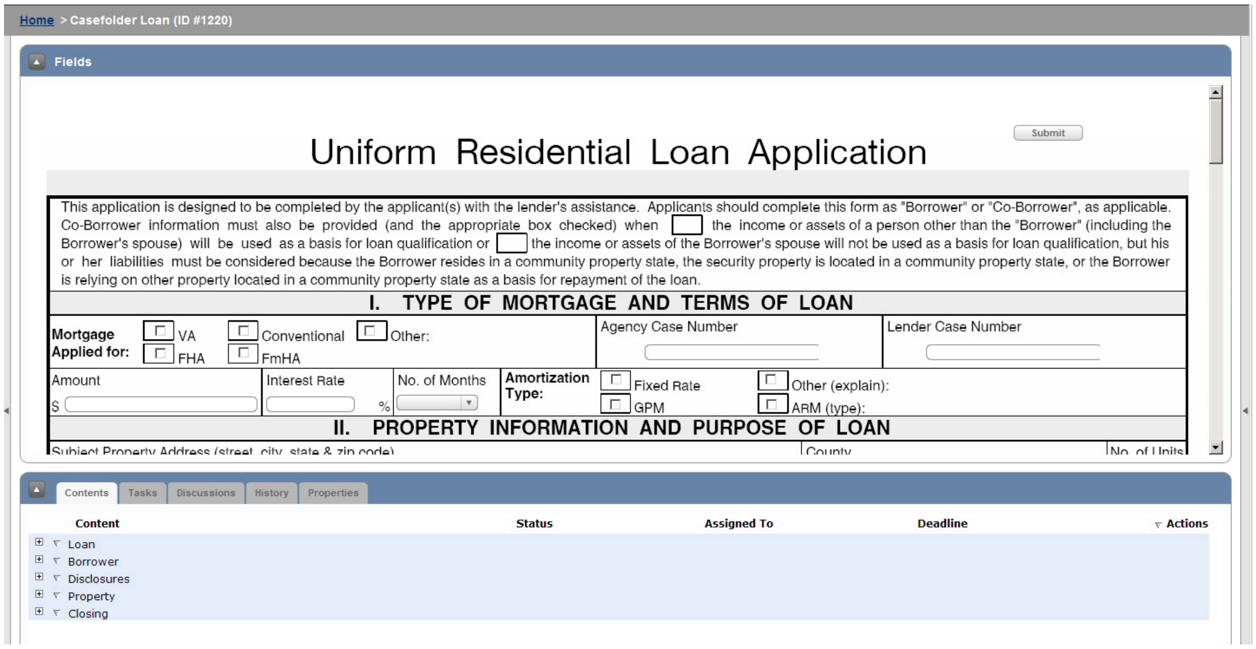
Pet: Canine | Breed: Mutt

ID Chip? ID Chip [✓ No ID Chip]

Weight: 32.5

Delete Dog Record

The following illustration shows an object with a presentation that was customized using a layout.



Home > Casefolder Loan (ID #1220)

Fields

Submit

Uniform Residential Loan Application

This application is designed to be completed by the applicant(s) with the lender's assistance. Applicants should complete this form as "Borrower" or "Co-Borrower", as applicable. Co-Borrower information must also be provided (and the appropriate box checked) when ☐ the income or assets of a person other than the "Borrower" (including the Borrower's spouse) will be used as a basis for loan qualification or ☐ the income or assets of the Borrower's spouse will not be used as a basis for loan qualification, but his or her liabilities must be considered because the Borrower resides in a community property state, the security property is located in a community property state, or the Borrower is relying on other property located in a community property state as a basis for repayment of the loan.

I. TYPE OF MORTGAGE AND TERMS OF LOAN

Mortgage Applied for: ☐ VA ☐ FHA ☐ Conventional ☐ FmHA ☐ Other: | Agency Case Number | Lender Case Number

Amount $ | Interest Rate % | No. of Months | Amortization Type: ☐ Fixed Rate ☐ GPM ☐ Other (explain): ☐ ARM (type):

II. PROPERTY INFORMATION AND PURPOSE OF LOAN

Subject Property Address (street, city, state & zip code) | County | No. of Units

Contents | Tasks | Discussions | History | Properties

Content	Status	Assigned To	Deadline	Actions
Loan				
Borrower				
Disclosures				
Property				
Closing				

XML Field Capabilities

XML fields enable large quantities of structured data to be stored with a form, Casefolder, Filestore, or Process instance record. This enables an application designer to design a database schema that represents the key data for the record as standard simple fields (for easy access and high performance query), while capturing other essential but non-key information within a single XML field. This can significantly simplify the design of applications where large amounts of data must be captured, stored, and retrieved, since the designer does not have to represent every data element as a field in a database table.

Every element of data in an XML field can be accessed easily from within scripts and expressions, process conditions, JSP forms, and so on, meaning that the use of XML is still very dynamic and usable within the system.

Using the XQuery capabilities of the database, data within XML fields can be used to query the database in conjunction with standard key fields.

For example, assume that an insurance agent submits a new customer application for an insurance policy from his internal system. This application form is actually a very large XML document that describes everything about the customer: demographics, policy requirements, health, financial status, and so forth.

The Case360 application designer can simply extract the information that is needed routinely for query, database relations, and process coordination, leaving the XML document intact and stored in a single XML field. The other non-key information elements remain accessible within the XML document for display, update, and query. If the data definition that is passed from the insurance agent to Case360 needs to change in the future, there may be no impact on the Case360 database design, since only the key fields are modeled directly in the database. The XML holds all the other information that is required for record keeping, display, and so forth, and the format of this may be far more flexible.

Enumerations

The decimal, duration, integer, and text fields support the ability to define a list of possible field values for the user to select instead of typing in a value. When a field has a value list, the field is presented to the user as a drop list instead of a free-form input field. There are two types of enumerations.

Static Enumerations

Static enumerations are defined when the field is defined. For example, values called North, East, South, and West might be defined for a REGION field in a sales forecasting application. The application designer enters a field's value list into the value list text field on the field definition dialog box. Each choice in the value list is entered on a separate line. Each choice consists of two components: the value to be stored in the database and the optional display name (the text the user sees and selects in the list, if different from the database value) separated by a semicolon.

Dynamic Enumerations

Dynamic enumerations are defined using scripts that run a query or perform another data manipulation operation and return results as dynamically generated values that can be selected by a user. For example, a script might retrieve a list of current sales figures from which the user can make a selection. When you define a dynamic enumeration, you specify any dependency fields and enter the name of the supporting script and the mapping script. A dynamic field can depend on another field for input. Any field can be defined as a dependency field. For example, a field called Sales Rep might be a dependency for a dynamic field called Sales Figures.

Chapter 6

Process

Process provides automated data processing based on a business model. This model defines the actions performed on objects and enforces the rules required to support those actions.

A process is a set of instructions that routes objects (instances) to locations (activities) where pre-defined work is performed on them. When the processing is complete, the process moves the instance to the next activity. Work can be performed by human workers, programmed activities, or automated robot agents.

Designers create processes using the Visio-based Process Designer. A Visio plug-in provides all the process functions, connectivity to the Case360 Server for accessing metadata information, and process publishing functions. An Import function can be used to import a Visio drawing of a process as a new Case360 process.

Multiple processes can be installed on a single server. As designers modify processes, they can retain multiple versions and run them at the same time. An ACL can be assigned to each version to control access to that version and to all the instances in process.

A process engine runs independently of any user activity. It "drives" instances through the process by monitoring the database for instances ready to be processed and then executing the instance rules. There is one process engine for each process. Administrators can start and stop a process engine by starting or stopping the process.

Process form definitions may include repository key fields that can link to, or reference, objects in any installed repository. While a casefolder is the most common object to be routed by a process, there is no direct link between Casefolder and Process. Case360 customers can use Process without using Casefolder and vice versa.

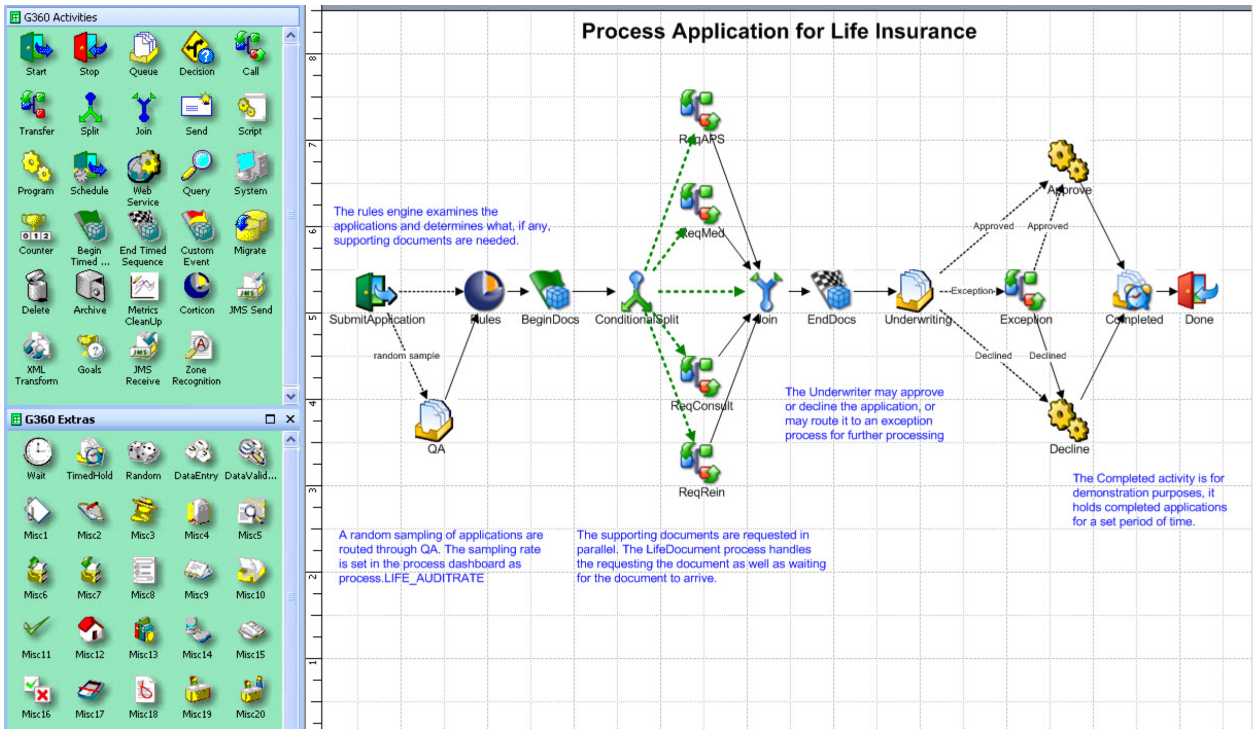
Process designers can specify that selected users perform certain types of work. Users access their assigned work by running queries. A single query can return work in multiple activities or even in multiple processes.

Immediate processing causes the server to evaluate process rules while the user waits or to define a sequence of activities to be performed while the caller waits. This enables the calling software to obtain the results of that processing.

Using immediate processing, an application designer can specify that an activity should be performed by the same user as the previous activity. When the user clicks a dispatch button in the first queue activity, the process is executed immediately, the rules are evaluated on the server while the user waits, and the user interface for the next queue activity is displayed. The capability to walk a user through a series of activities enables application designers to easily develop process controlled wizard interfaces or complex forms applications.

Process Designer

Process Designer includes a palette of built-in activities. The following illustration shows a sample process.



The rules engine examines the applications and determines what, if any, supporting documents are needed.

The Underwriter may approve or decline the application, or may route it to an exception process for further processing

The Completed activity is for demonstration purposes, it holds completed applications for a set period of time.

A random sampling of applications are routed through QA. The sampling rate is set in the process dashboard as process.LIFE_AUDITRATE

The supporting documents are requested in parallel. The LifeDocument process handles the requesting the document as well as waiting for the document to arrive.

Process Designer displays a stencil of process shapes that represent activities such as Split and Join. Microsoft Visio features an easy-to-use graphical interface and predefined shapes with properties that can be set. These can be used in conjunction with the Process Designer shapes.

To create a process definition, a designer drags an activity shape from the stencil to the drawing page. Double-clicking the shape enables a designer to set properties and other information such as

- The Name of the activity.
- Comments.
- Entry, Exit, and Deadline expiration rules.
- Custom event handling.
- Calls to other processes or external programs.

Finally, the designer connects the activities in the order in which they will be executed in the process. A Complexity Filter feature enables the application designer to manage the complexity in a process diagram by showing or hiding various levels of detail. The process designer can change the way a process works, alter routing, and change who sees process instances.

When the process design is complete, the designer uses Process Designer to validate the process and generate an XML process definition for the Case360 Server. The activities, behaviors, and rules for the process are included in the definition.

Process Designer uploads the Visio process file along with the process definition to the server. The Visio file is saved using a system defined Filestore template named Process Definitions. Icons in the Process Dashboard’s properties page enable you to download the definition of any process version.

Instances

An instance is an object that is routed to one or more activities in a process. Instances are stored in the Process repository.

An envelope is a mechanism used to route instances from one activity to another. The envelope fields contain system-generated information about the instance such as ID, lock status, when the instance entered the current activity, and so on.

Each instance consists of one set of custom fields and one or more envelopes. Instances that will be processed in parallel contain one envelope for each path the instance will follow during processing.

Fields

Process uses several types of fields. Application designers can define the fields in the schema using the Case360 Toolbox or they can create new fields using the Table Definitions interface from Process Designer.

Process designers define the fields for process instances. Each process instance has its own copy of these fields. In addition, the process designer can define a set of process global fields. There is a single copy of these global fields shared by all of the process instances. Process fields can also be used in process queries to control the query results.

See the chapter in this document called "Forms and Fields" for additional information.

Custom Fields

Application designers can associate custom fields with each instance to

- Control routing when the fields are used as variables in activity rules.
- Select and sort instances in queries.
- Reference objects in repositories.

Process Fields

Process fields are used to hold global values that can control the behavior of a process. There is a single instance of each of these fields for the entire process. Process fields can also be used in process queries to control the query results.

User Fields

A user field is associated with the user who has the instance’s envelope locked. If the envelope is not currently locked, the user fields are obtained from the user who last locked the instance’s envelope.

Envelope Fields

Envelope fields are used internally by Process Management to determine the state of the process instance and the lock user and type. They are also used to keep track of an instance's location, step name, entry and exit times, timeout time, workflow ID, version, and so forth.

Metrics Fields

Metrics fields are custom, user defined fields used to track the status of a process with managerView.

System Fields

System fields provide access to information about the server configuration. In addition to the predefined system fields, the application designer can define additional custom configuration properties.

Activities

An activity is a point in the process where an action is performed on an instance. Process Designer provides various types of activities to perform specific actions. Each type of activity has its own actions and rules.

The Extensible Activity framework enables developers to add new types of activities to Process Designer. An extensible activity consists of a shape in a Visio stencil, an activity definition file that specifies the behavior of the activity, a design time user interface, and a runtime Java class that implements the actual processing of the activity.

Activity Types

The Process Designer Help contains detailed information about the available activities. The following list illustrates the type of processing that some of the activities perform:

Start - Specifies where processing on an instance begins. There can be multiple start activities in a process, allowing for subflow processing.

Stop - Specifies where the instance is routed when processing is complete. There can be multiple Stop activities in a process, allowing for independent subflow processing.

Queue - Requires a user action on an instance before the instance continues to the next activity. The Queue and Start activity options enable the designer to modify the frameset look and feel as well as design custom dispatch buttons.

Split - Specifies the point where an instance splits and enters two or more paths, in order to be processed in parallel. Every Split activity must have a corresponding Join activity.

Join - Specifies the point where multiple paths that originate at the Split activity converge. Every Join activity requires a corresponding Split activity.

Program - Performs an automated action on an instance.

Call - Invokes another process. When that process is complete, the instance returns to the originating process to continue activity processing. A Call activity is also used when a process spans multiple pages.

Transfer - Sends instances to another process and continues processing with the target process. Instances do not return to the originating process.

Decision - Specifies that a decision is made based on information supplied in a field value or by automation.

Schedule - Creates new process instances on a defined schedule. This enables the designer to create processes to perform system maintenance.

Counter - Increments a system counter that can then be displayed using the system metrics tool.

Script - Performs a script defined by the application designer.

Metrics cleanup - Deletes obsolete process metrics data. This is a housekeeping activity that should be triggered periodically.

Query - Runs a query to locate one or more objects in the system that can then be manipulated by subsequent process activities.

Life Cycle Activities (migrate, delete, archive) - Provide the capability to manage the life cycle of documents, casefolders, and process instances on the system.

Send - Enables a process designer to send an e-mail message.

Web Service - Can be used to dynamically invoke a Web Service from a process without the need to write code. The application designer specifies a sequence of methods to be invoked, mapping the method parameters to fields in the process instance, or any object referenced by the process instance through Repository Key fields.

Begin Timed Sequence - Marks the starting point of a timed sequence within a process. Timed sequences enable measuring elapsed time between any two points in the process.

End Timed Sequence - Computes the elapsed time since the Begin Timed Sequence activity started. Timed sequences enable measuring elapsed time between any two points in the process.

Custom Event - Supports the concept of a custom logged event. This is simply a named event, with no elapsed time information, but with just a timestamp and associated metrics field values.

Corticon - Provides built-in support for the Corticon rules engine.

JMS Send - Enables a process to send a JMS message to a specified queue or topic and to optionally wait for a reply message.

JMS Receive - Enables a process to retrieve a message from a JMS queue.

XML Transform - Enables a process to perform a transform on an XML document using an XML and XSL source.

Goals - Can be used to obtain the current state of the process's Key Performance Indicators (KPIs) from managerView. You must purchase managerView separately.

Zone Recognition - Can be used to extract information from bar codes and text zones on scanned image documents. This activity uses the recognition server, which must be purchased and installed separately.

System - An activity that represents a generic action that will take place on the server. This activity may appear after a process definition is created by importing an XPDL document. It has no inherent behavior and will only perform actions specified by the rules on the activity. The System activity is analogous to a Queue activity. It can be changed to any activity type by using the Change Activity Type selection from the Process Designer menus.

Rules

A rule is an action that is performed when a specified condition exists. It consists of an action and its parameters, along with an optional condition that determines whether the rule is executed.

There are three locations where rules that control the flow of work in and out of the activity are defined. A process engine executes rules in these locations, depending on the state of the instance. A process can have many rules in each location.

Entry rules - Executed against each instance arriving at the activity.

Exit rules - Executed against each instance leaving the activity.

Deadline rules - Executed against an instance if it remains at the activity beyond the specified due date.

Actions

An action is the processing performed when a rule is executed. Most actions take parameters that specify how to perform the action, including

- Routing the current envelope to a specified activity.
- Creating a new parallel envelope for the instance at a specified activity.
- Calling a method in a Java class.
- Setting the value of a specified field to a value obtained by evaluating an expression.
- Putting an instance into error if a value is incorrect.
- Setting metrics on a particular instance to determine processing time.
- Running a script that will set a value, create an object, and so forth.
- Setting a timeout on an activity.

A process designer can also define application specific dispatch actions and buttons (for instance, Approve and Deny instead of the default Route On action).

Conditions

A condition is a Boolean expression that is written using available fields. The expression can use comparison, assignment, computational, and logical operators as well as constants. The result of the condition determines whether the action will be performed.

Process Metrics

An important feature of any process management system is the ability to collect information about the behavior of the business process. This type of information enables companies to re-engineer the process to improve efficiency.

Administrators can enable the metrics gathering feature for individual processes. The information is written to a database table. Be aware that metrics gathering can reduce system performance and can take up a significant amount of disk space.

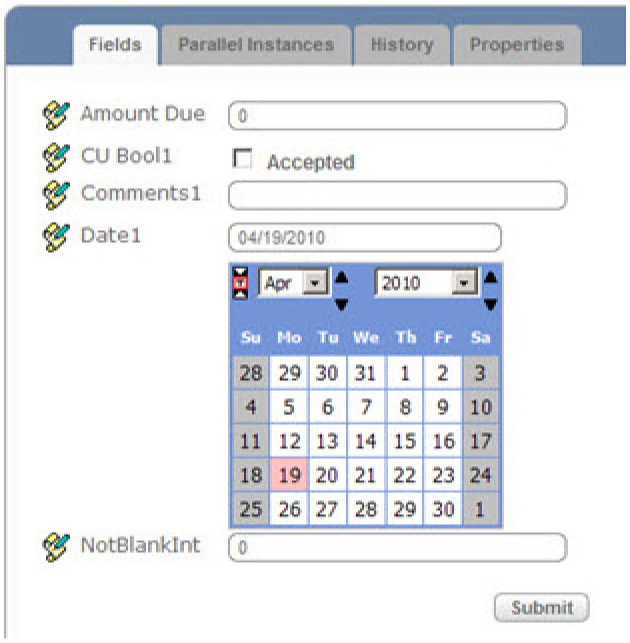
To allow for control over the process metrics table, an activity is provided that deletes any metrics records older than a user-defined retention period. This is a self contained scheduled activity and it does not create any instances.

Administrators can run queries against the collection of historical data to report useful information. For example, administrators can determine the average time it takes for an instance to be routed through the entire process.

Process User Interface

Process provides a default user interface for instances. Application designers can easily create customized framesets. It is often convenient to embed the Casefolder interface in an instance frameset.

The persona-based interface for creating an instance has several components, as shown in the following illustration, where the Fields tab is selected.



Amount Due: 0
CU Bool1: ☐ Accepted
Comments1:
Date1: 04/19/2010

Apr 2010

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

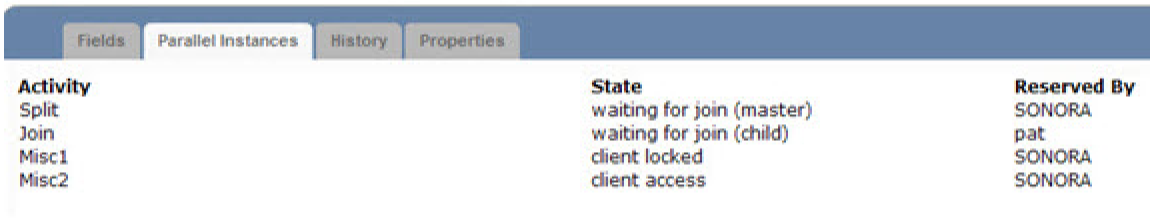
NotBlankInt: 0

Fields

The Fields function enables users to view and modify fields on the form defined for an instance. See the chapter in this document called "Form Data and Fields" for details.

Parallel Instances

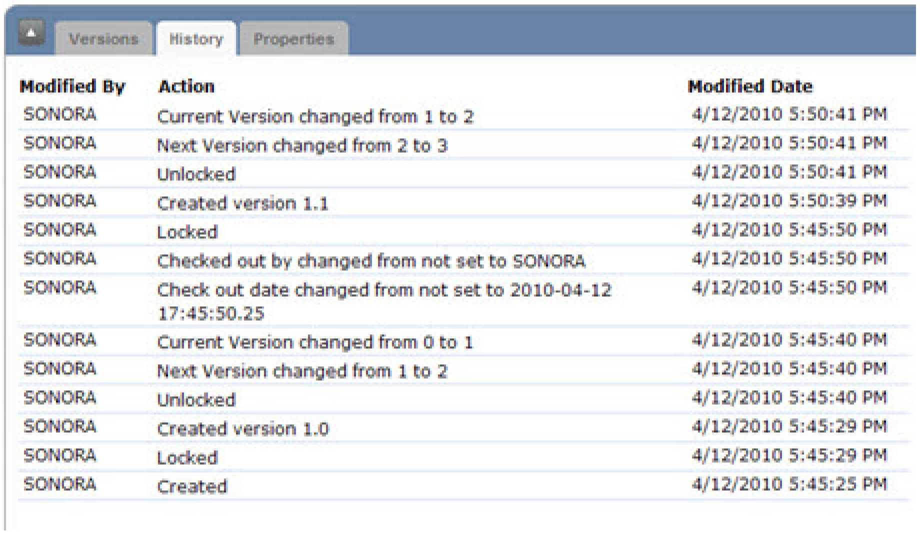
The Parallel Instances function enables users to view the name of the other child activities that are created when an instance enters a Split activity.



Activity	State	Reserved By
Split	waiting for join (master)	SONORA
Join	waiting for join (child)	pat
Misc1	client locked	SONORA
Misc2	client access	SONORA

History

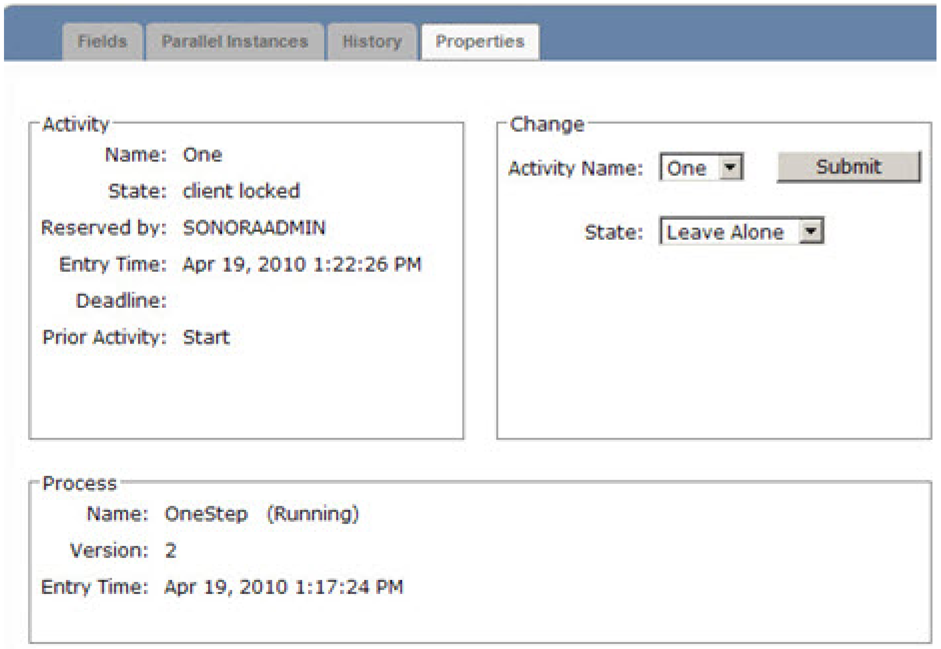
The History function tracks activity that occurs on the selected object. It identifies the user who made the change, the date the change was made, and the action that was performed on the object.



Modified By	Action	Modified Date
SONORA	Current Version changed from 1 to 2	4/12/2010 5:50:41 PM
SONORA	Next Version changed from 2 to 3	4/12/2010 5:50:41 PM
SONORA	Unlocked	4/12/2010 5:50:41 PM
SONORA	Created version 1.1	4/12/2010 5:50:39 PM
SONORA	Locked	4/12/2010 5:45:50 PM
SONORA	Checked out by changed from not set to SONORA	4/12/2010 5:45:50 PM
SONORA	Check out date changed from not set to 2010-04-12 17:45:50.25	4/12/2010 5:45:50 PM
SONORA	Current Version changed from 0 to 1	4/12/2010 5:45:40 PM
SONORA	Next Version changed from 1 to 2	4/12/2010 5:45:40 PM
SONORA	Unlocked	4/12/2010 5:45:40 PM
SONORA	Created version 1.0	4/12/2010 5:45:29 PM
SONORA	Locked	4/12/2010 5:45:29 PM
SONORA	Created	4/12/2010 5:45:25 PM

Properties

The Properties tab displays information about the selected instance, such as the activity name, the entry time, and who has locked the instance. From this tab, an administrator can change the state of the instance, or move it to a different activity. If the state is changed to Locked, an administrator can also change the Reserved By user.



Activity

Name:	One
State:	client locked
Reserved by:	SONORAADMIN
Entry Time:	Apr 19, 2010 1:22:26 PM
Deadline:	
Prior Activity:	Start

Change

Activity Name: One

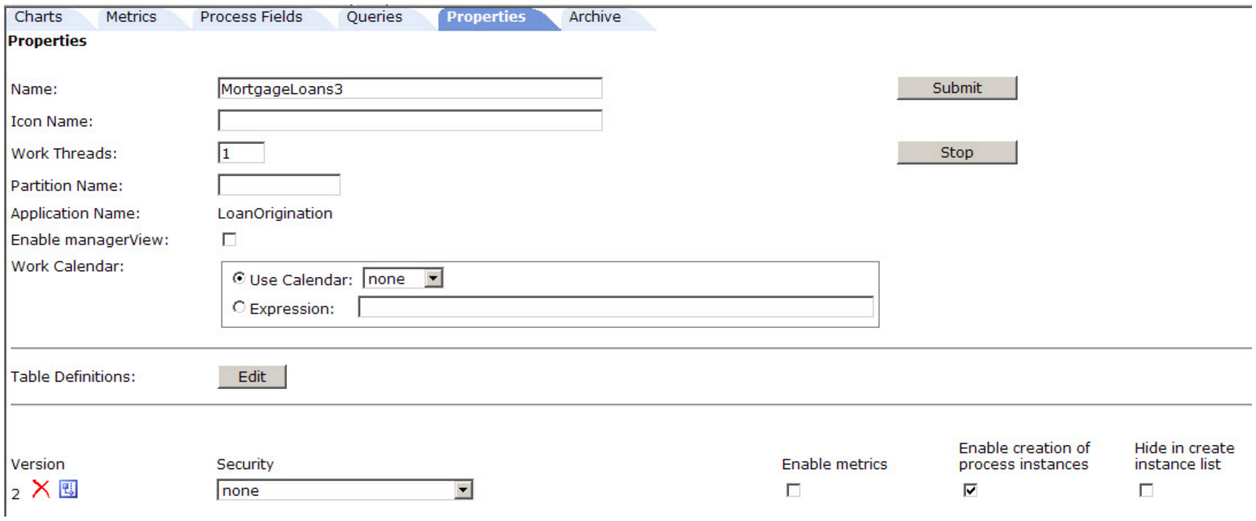
State: Leave Alone

Process

Name:	OneStep (Running)
Version:	2
Entry Time:	Apr 19, 2010 1:17:24 PM

Process Dashboard

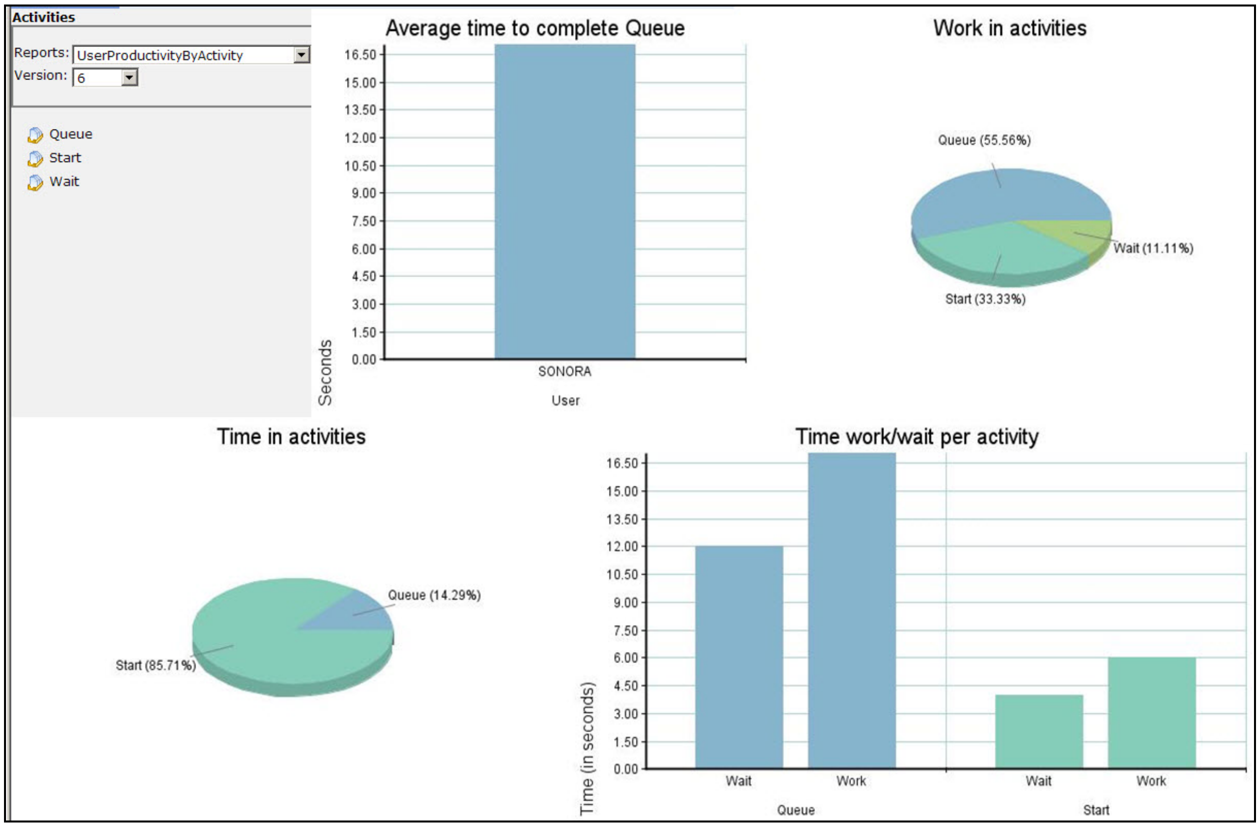
Processes are administered using a graphical tool called Process Dashboard. This tool displays the name and status of each installed process and it is used by both system administrators and application designers.



Name:	MortgageLoans3
Icon Name:	
Work Threads:	1
Partition Name:	
Application Name:	LoanOrigination
Enable managerView:	☐
Work Calendar:	● Use Calendar: none ○ Expression:
Table Definitions:	Edit

Version	Security	Enable metrics	Enable creation of process instances	Hide in create instance list
2	none	☐	☑	☐

The following illustration shows the Charts tab for the Process Dashboard.



Administrators use Process Dashboard to

- Access process status reports.
- Access system metrics generated by a process.
- Update values of process fields.
- Define or change schedules for scheduled activities.
- Start and stop processes.

Application designers use Process Dashboard to

- Define process properties.
- Create process queries.
- Delete process versions, if necessary.
- Define archive templates.
- Delete or modify process queries.

Process managerView Integration

managerView, which provides analytic reporting for Business Process Management (BPM) applications, can be used in conjunction with process management. To utilize managerView, you need to purchase and install managerView and configure managerView support on your server.

managerView with Case360 offers the capability to affect the way processes route and deliver work, based on the real-time business metrics produced by managerView process analytics. This enables processes to rapidly and automatically respond to changes in workload, type of work or worker availability to constantly meet predefined business goals.

The goals activity enables process designers to set up management processes that obtain the current performance of a process against its Key Performance Indicators (KPIs) in managerView. The following list describes some of the functions you can perform:

- Specify which metrics data to send to managerView.
- Enable or disable managerView events for a particular activity.
- Enable or disable managerView reporting for any process version.
- Synchronize process information between the Case360 server and managerView.
- Track the actual time users spend working on process instances down to a specific activity level.
- Copy KPI values from managerView into process and global fields.

Chapter 7

Filestore

Filestore is an internal Case360 repository that provides content management functions, including uploading and downloading of files to and from the file server, versioning, and check-in and check-out.

Filestore can manage any type of file. Extended support for TIFF image files is provided by G360 Imaging for the Web. At any given time, a document may be locked by a user.

Filestore uses templates that are defined by application designers to create new documents. Users access documents directly through queries created by application designers or administrators, or indirectly through casefolders or instances that reference the objects.

Filestore Templates

Application designers can define any number of Filestore templates, each of which can have its own set of custom fields.

The templates are used to create documents. The documents inherit the properties of the template used to create them. Properties include the name, the path to the object’s location, and the type of security assigned. A retention date property specifies that a document can be deleted no sooner than a specified date.

The Case360 database keeps track of which template is used to create each instance of a document. The database ensures that a user cannot delete a template if the template is still in use. That is, a template cannot be deleted if a document exists that was created from that template.

File Access

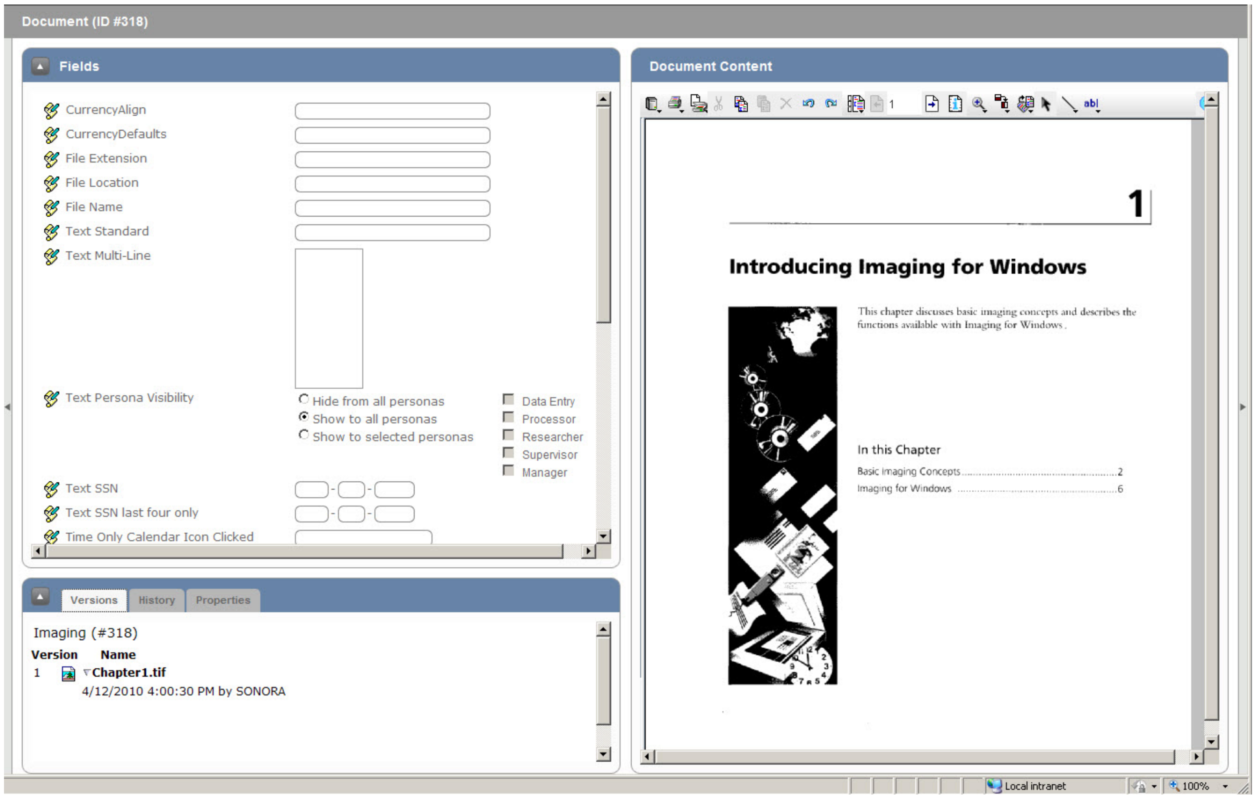
Users typically retrieve Filestore repository objects through the query mechanism, which is common to all the repositories. User queries are based on fields defined in documents.

Application designers can specify the set of action buttons that appear next to each entry in a query results list. They can use the button sets provided by Case360 or create their own button sets.

Content Management

The Filestore servlet provides the user interface for document management. By default, when a document is opened, a window similar to the following is displayed. This window shows the document in three panels; Fields in the upper left, Versions, History, and

Properties in the lower left, and the Document Content in the right panel. Each panel can be expanded or collapsed as needed.



CurrencyAlign
CurrencyDefaults
File Extension
File Location
File Name
Text Standard
Text Multi-Line
Text Persona Visibility: Hide from all personas [✓ Show to all personas] Show to selected personas — Data Entry, Processor, Researcher, Supervisor, Manager
Text SSN
Text SSN last four only
Time Only Calendar Icon Clicked

Imaging (#318)

Version	Name
1	Chapter1.tif 4/12/2010 4:00:30 PM by SONORA

1

Introducing Imaging for Windows

This chapter discusses basic imaging concepts and describes the functions available with Imaging for Windows.

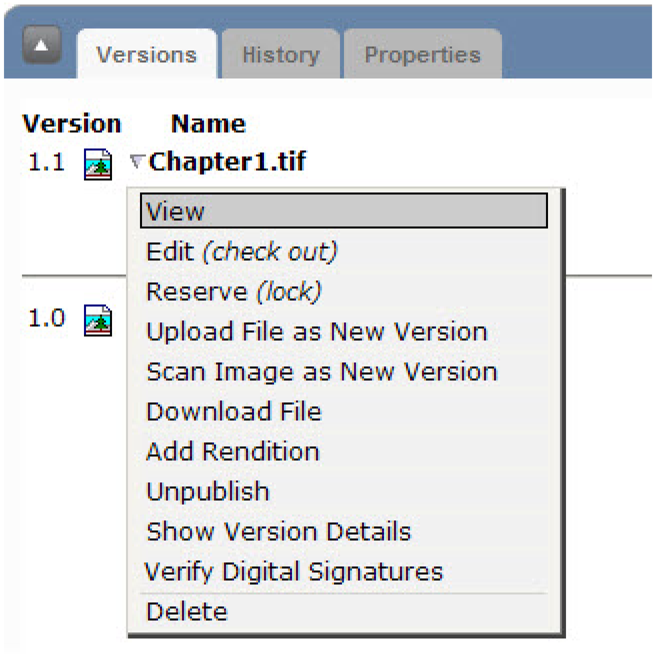
In this Chapter

Basic Imaging Concepts 2
Imaging for Windows 6

Local intranet 100%

Application designers can easily create customized framesets to present the document in different ways. It is often convenient to embed the Filestore interface in a Casefolder or Process frameset.

Clicking an entry in the version list displays a pop-up menu that lists the functions that can be performed on that version.



The following functions may be available, depending on the options selected when the Filestore template was created and on the current state of the document.

View - Displays the selected version or rendition in either the Document Content panel or in a separate window (renditions).

Edit (check-out) - Locks an object when the user starts editing it. A document can be locked by only one user at a time. Only that user can make changes to the document.

Reserve (lock) - Reserves (locks) a document for a user who expects to edit that document at a later date and who wants to ensure that no other user modifies the document first.

Upload file as new version - Uploads a file as a new version without going through the Edit/Save Changes functions.

Scan image as new version - Opens Imaging for the Web to scan any paper in the user's scanner. When the document is ready, the Save Changes function is used to upload the image document to the server.

Download file - Downloads a copy of a version or rendition from the server to the user's computer.

Add rendition - Uploads an alternate rendition of a version to the server.

Publish - Makes a document that has been configured with enhanced versioning available to any user with read permission on the document. An unpublished version is visible only to users with write permission on the document.

Show version details - Displays information about the selected version, if a document has either enhanced versioning or field versioning enabled. Documents with only field versioning enabled show the version's fields. Documents with only enhanced versioning show the enhanced versioning information.

Verify digital signatures - If the application designer has specified a signature pool where digital signatures of the files are to be stored, compares the digital signature to the file to ensure that the file has not been tampered with.

Delete - Deletes a version or rendition from the document on the server.

Save Changes (check-in) - Saves the changes as a new version when the user finishes editing the object and automatically unlocks the document.

Discard Changes (unlock) - Discards the user’s changes to a document and unlocks it.

Full Text Indexing

Full text indexing is the process whereby the system indexes all the words in the uploaded files so that users can quickly search for a document based on its content.

There are two types of full text support. One uses the indexed blob data manager and the other uses the file system. Both types of full text support are implemented using the underlying database, so there are significant differences in behavior based on your choice of database server.

A blob is a SQL built-in type that stores a binary image object as a column value in a row of a database table.

- Blob storage should be used for content indexing if file content is stored in the database as blobs.
- File system storage for content indexing should be used if the file content is stored in the file system.

Versioning

The Case360 versioning capabilities enable users to create a new copy of a file each time they check a file out, update it, and check it back in. Whenever a document is checked in, a new version is created and a sequential number is assigned to it. Users can view the history of changes made to files, and, if desired, retrieve an older version of the file.

Field Versioning

If Field Versioning is enabled in a template, a copy of the document’s fields is made as each version is checked in. This option is intended to support full text search setup with the full text index appearing as a custom field. This enables each individual version with a given search term to be displayed in the query results. Field Versioning is not necessary if the full text index is defined using the Indexed Blob data manager, or if searching older versions is not necessary.

Image Page Versioning

The Filestore Image Page Versioning feature avoids storing redundant copies of image data as image documents are modified by saving only the information that changed in each new version. Image Page Versioning is an optional feature, selected by the application designer with an option in the template.

Enhanced Versioning

If Enhanced Versioning is enabled in a template, a dialog box appears when a user checks out or checks in a document that is based on that template.

- The Check Out dialog box prompts the user for a checkout comment that explains why the document is being checked out and an expected check-in date.
- The Check In dialog box prompts the user for a check-in comment, a version label, and an indicator of whether the version is published or unpublished.

A user can go to any version and view any of the check-in and check-out information at any time.

Renditions

Any number of files can be attached to the current version of a document. These additional files are referred to as renditions. While the uses for renditions are limitless, they are typically alternate renderings of the same document. For example, the primary rendition may be a PDF file that can be displayed by anyone with a PDF viewer while a secondary rendition might be the same document in a word processing document. Another example would be a scanned document and its OCR’d results. The user interface displays the primary rendition of a document first with its renditions listed below it.

History

The History function tracks all the activity that occurs on the selected object. It identifies the user who made the change, the date the change was made, and the action that was performed on the object. It does not track changes that are made to a file. For example, it does not track annotations made to a TIFF file.

An option is available for logging an entry in History whenever a Filestore object is viewed.

Imaging Options

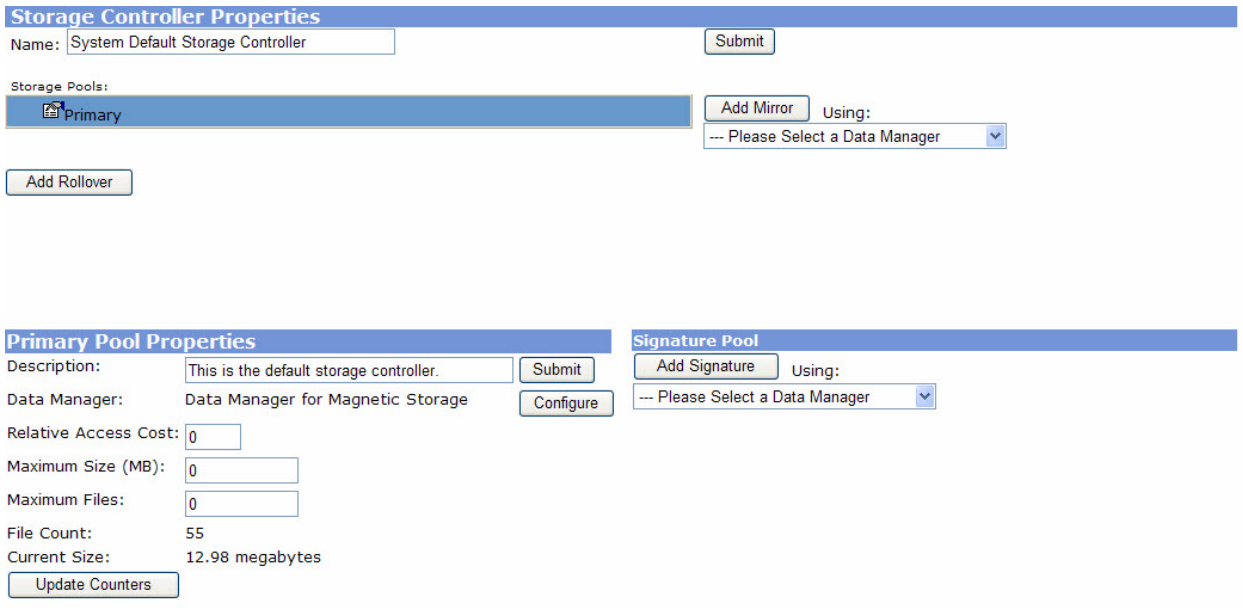
The Imaging Options tab in Filestore templates provides the application designer with total control over the Imaging for the Web functionality available to a user viewing or editing an image document.

Initial Action

The Initial Action template option can be used to cause an automatic upload or scan operation when a document is created from a template.

File Storage

Filestore offers a sophisticated set of storage management features.



Storage Controller Properties

Name: System Default Storage Controller [Submit]

Storage Pools: Primary

[Add Mirror] Using: --- Please Select a Data Manager

[Add Rollover]

Primary Pool Properties

Description:	This is the default storage controller.
Data Manager:	Data Manager for Magnetic Storage
Relative Access Cost:	0
Maximum Size (MB):	0
Maximum Files:	0
File Count:	55
Current Size:	12.98 megabytes

[Submit] [Configure] [Update Counters]

Signature Pool

[Add Signature] Using: --- Please Select a Data Manager

At the lowest level is the data manager, which implements basic storage to physical media. The system includes three data managers: magnetic, indexed blob, and blob. The data manager interface is open, enabling customers and integrators to extend the system to include additional storage destinations.

- The magnetic data manager stores files as ordinary files on file servers visible to the Filestore servlet. A file naming scheme distributes the files into a hierarchical folder structure distributed across up to eight file servers. The magnetic data manager offers an option to perform secure erasure on deleted content. The file servers are not directly accessible to client computers. The application server is located between the client computer and the file server, enabling Filestore to enforce security.
- The magnetic data manager implements the US Department of Defense clearing standard DOD 5220.22-M, and you can specify the number of overwrite passes it makes (DOD recommends using 7 passes).
- The database blob data manager stores files inside the database. While this does not perform as well as magnetic disk, it may be easier to manage for backup and recovery purposes (depending on your application's volumes and the tools you use).
- The optical data manager stores files on optical (WORM) media. WORM media is required for some applications to meet regulatory requirements. The optical data manager requires the Optical Storage Manager software, which is available separately.
- The Centera and SnapLock data managers provide the capability to store content on those servers.

At the next level up is the storage pool. A storage pool is based on one data manager and includes any configuration options for the data manager as well as storage limits and an access cost.

At the top level is the storage controller, which uses an array of storage pools to implement mirroring, digital signatures, and rollover.

- Mirroring is the storing of content in more than one location for disaster recovery. When retrieving files, the storage controller uses the pools’ access cost to decide which mirror pool to use to retrieve the file. The use of mirrors hides problems such as broken network connections from users.
- Digital signatures are used to detect unauthorized "back door" tampering.
- Rollover is switching to a new pool when the current pool is filled.

Applets Used by Filestore

Filestore uses the Download, Upload, and G360 Imaging for the Web applets to manage files. These applets do not need to be installed ahead of time on the client computer; they are downloaded to the client computer as needed. All the required installation settings are set automatically when the software is downloaded.

The applets require the Sun Java Runtime Environment (JRE) to be installed on the client computer. If the JRE is not already installed, the user is prompted to install it when running an applet for the first time. (The Case360 package includes the JRE and installs it for the user.)

While you can disable use of the applets, the Check-in and Check-out operations are enhanced through their usage. If the applets have been generally disabled, you can designate specific users to use the applets.

Download

When a user first edits (checks out) a file, the Download applet prompts the user to specify where to copy the file on the local computer. The applet then copies the file from the server to the specified location and launches the application registered to handle that file type on the local computer. It also records the destination file name on the server. If a user later edits the file, the applet opens the downloaded copy that is already on the local computer.

Upload

After a user finishes editing a file and saves changes (checks it in), the Upload applet locates the copy of the file on the local computer. It then uploads the file as a new version in the document.

OpenText Imaging for the Web

Imaging for the Web is a Web-based application used for viewing, editing, scanning, and annotating TIFF files. Imaging for the Web processes single and multi-page TIFF files and supports image annotations.

Imaging for the Web downloads only the data that is needed to display the first image page and thumbnail views. This technique provides quicker performance; the user does not have to wait for all the image data to be downloaded before viewing the first page. While the user is viewing the first page, it continues to download nearby pages to improve response time as the user navigates within the document.

Imaging for the Web communicates with a servlet to perform operations on the image file on the server, retrieving data as needed. Image pages, thumbnails, and annotations are all independently read and written from the server.

While Imaging for the Web does not rely on a client-side software installation, it does download and use an ActiveX control. Because of this, Imaging for the Web is used only when the browser client is running Microsoft Windows.

An application designer has total control over the Imaging for the Web functionality available to a user viewing or editing an image document. When an image file is opened, the toolbar buttons are enabled based on the user’s permissions. Thumbnails are displayed in a pane on the left. The selected page is displayed in a pane on the right, and is scaled so that the entire page fits in the window. A user can create a new version of an image by scanning a document from a scanner connected to the user’s computer.

If the data is black and white, Imaging for the Web uses a scale to gray algorithm to improve readability.

To enable faster display of thumbnail images, the thumbnails are stored separately.

Imaging for the Web includes the following functions:

- Rotation and mirroring
- Page straightening
- Cropping
- Despeckling
- Modification of contrast and brightness
- Inversion (black on white to white on black)
- Printing
- Scanning
- Clipboard Cut, Copy, and Paste
- Undo/redo
- Annotation

The annotation function enables you to annotate an image and save the annotations. The annotations for each page can be stored separately from the image data. This enables the page to be annotated without changes being made to the image itself. To enhance security, image data and annotations are uploaded separately.

- Annotations you can apply to an image include
- Attach-a-Note comments
- Polygons
- Auto Polygons (automatically find text borders for redaction)
- Filled and hollow rectangles and polygons
- Straight or freehand lines
- Highlighting
- User initials

- Rubber stamps
- Text from files or text that you type on the page.

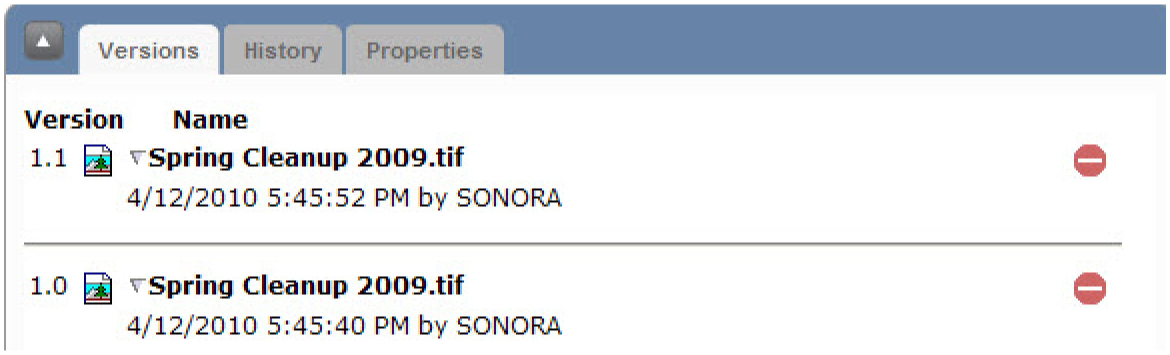
The Imaging for the Web security features treat annotating an image separately from modifying an image. In most applications, users are not allowed to modify the images, but they are allowed to add and modify annotations.

Filestore User Interface

The Filestore servlet provides a default user interface. Application designers can easily create customized frames. It is often convenient to embed the Filestore interface in a Casefolder or Process frameset.

Versions

The Version list shows the versions created as files are checked in and out. Numbers are incremented by one as each new version of the file is checked in. Clicking an entry in this list pops up a menu with the functions that can be performed on that entry based on the document options and the user's security.



Version	Name
1.1	Spring Cleanup 2009.tif 4/12/2010 5:45:52 PM by SONORA
1.0	Spring Cleanup 2009.tif 4/12/2010 5:45:40 PM by SONORA

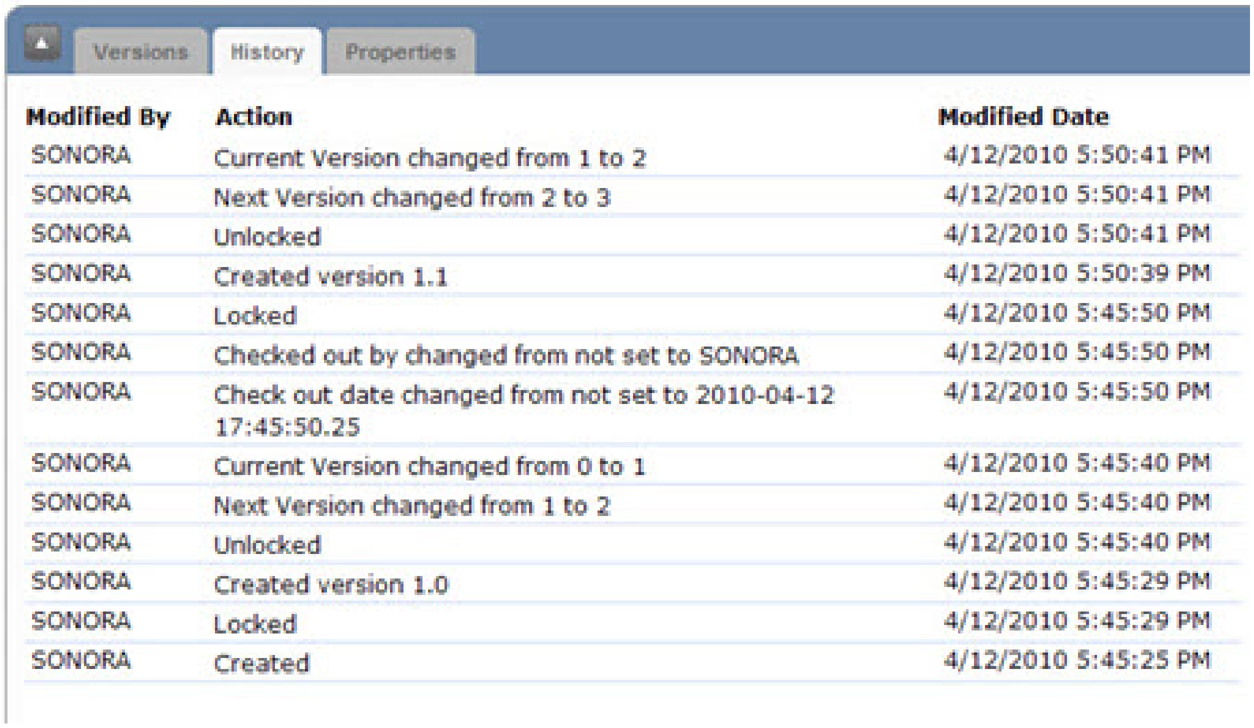
Fields

The Fields function enables users to access and modify the document's fields using the form created by the application designer.

Each field is secured independently. When a user accesses a document, any fields the user cannot access are either hidden or they are made read-only. See the chapter in this document called "Form Data and Fields" for more information.

History

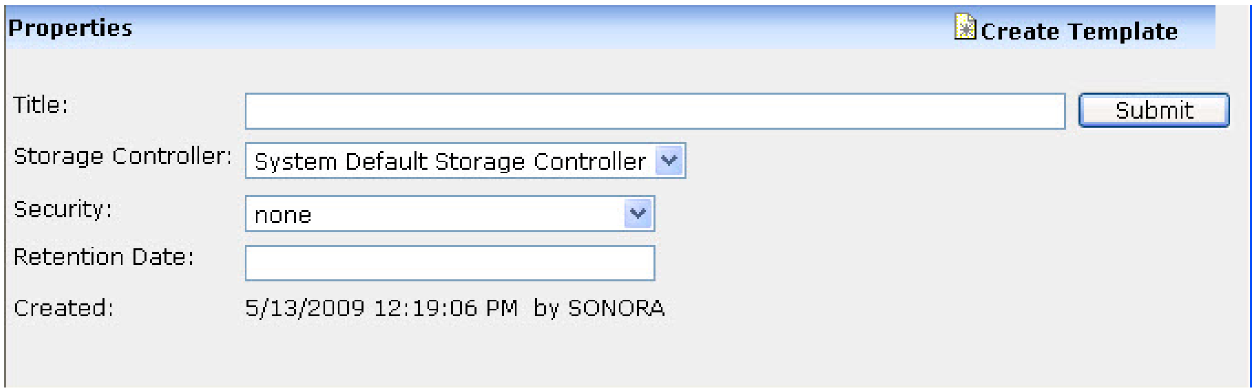
The History function tracks all the activity that occurs on the selected object. It identifies the user who made the change, the date the change was made, and the action that was performed on the object. A template option specifies whether the history should also record an entry any time a user views the document.



Modified By	Action	Modified Date
SONORA	Current Version changed from 1 to 2	4/12/2010 5:50:41 PM
SONORA	Next Version changed from 2 to 3	4/12/2010 5:50:41 PM
SONORA	Unlocked	4/12/2010 5:50:41 PM
SONORA	Created version 1.1	4/12/2010 5:50:39 PM
SONORA	Locked	4/12/2010 5:45:50 PM
SONORA	Checked out by changed from not set to SONORA	4/12/2010 5:45:50 PM
SONORA	Check out date changed from not set to 2010-04-12 17:45:50.25	4/12/2010 5:45:50 PM
SONORA	Current Version changed from 0 to 1	4/12/2010 5:45:40 PM
SONORA	Next Version changed from 1 to 2	4/12/2010 5:45:40 PM
SONORA	Unlocked	4/12/2010 5:45:40 PM
SONORA	Created version 1.0	4/12/2010 5:45:29 PM
SONORA	Locked	4/12/2010 5:45:29 PM
SONORA	Created	4/12/2010 5:45:25 PM

Properties

The Properties function displays information about the selected object, such as title, creation date, and security. Application designers see additional information, such as the template name and the version limit, when editing templates.



Title:	
Storage Controller:	System Default Storage Controller
Security:	none
Retention Date:	
Created:	5/13/2009 12:19:06 PM by SONORA

Chapter 8

Layout Designer

Each repository provides a Web-based drag and drop tool called Layout Designer that can be used to define custom forms to be presented to a user. Layout Designer is intended for use by designers who do not have any coding skills. It does not expect designers to have any knowledge of HTML, CSS, JavaScript, or AJAX (although designers with those skills will be able to do more with the Layout Designer than users without those skills).

Layout Designer does not appear in the toolbox. Instead, it appears inside many objects as a Layout tab. Layout Designer generates a JSP that is used to draw the form. An application designer can edit the generated JSP to add custom behaviors but once the JSP has been modified manually, Layout Designer can no longer be used to edit the layout.

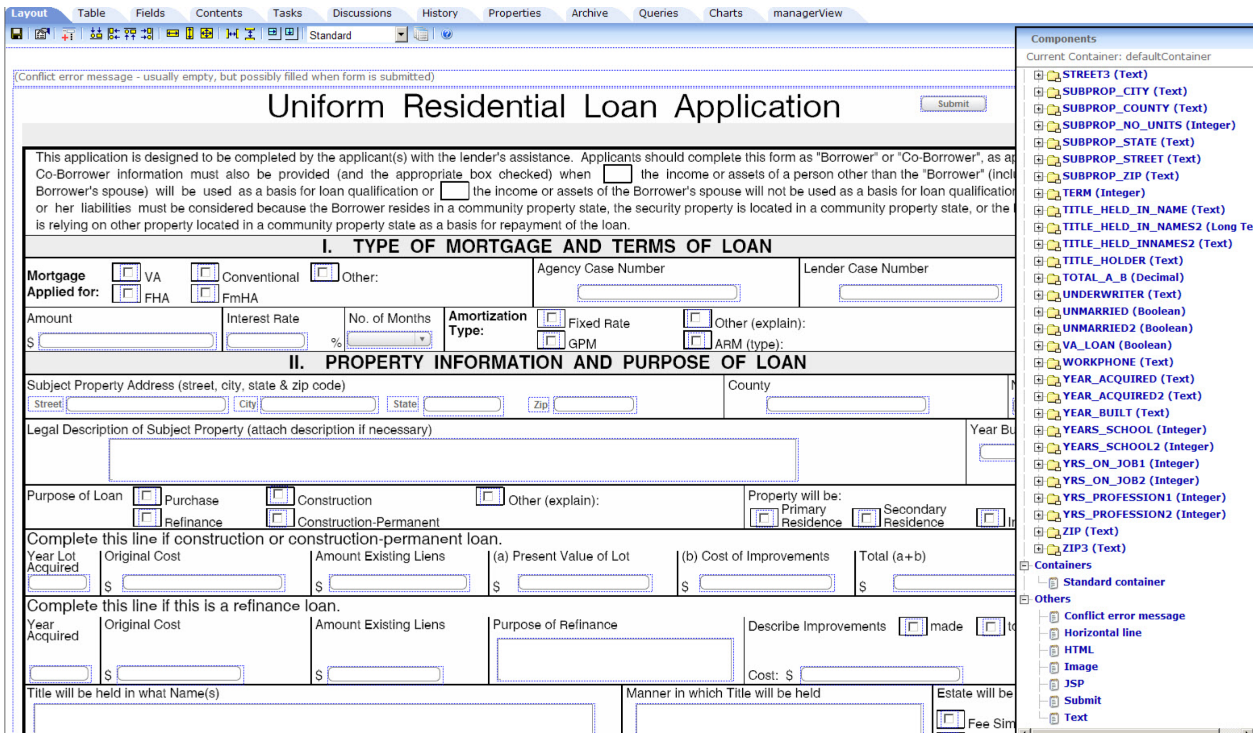
Contexts

Any object that has custom fields provides access to Layout Designer to define the presentation of those fields. Specifically. Layout Designer can be used in the following contexts:

- In Filestore templates to customize the presentation of document fields.
- In Casefolder templates to enable customizing the presentation of casefolder fields.
- In Form Data templates to enable creation of custom forms.
- In process definitions, which have both instance and process global fields, to provide access to two sets of layouts.
- In user roles to define the presentation of role specific fields.
- In queries to define query forms to parameterize queries. These fields are not persisted in the database.
- In menu toolbar buttons to define forms to parameterize toolbar actions. These fields are not retained in the database.

Layout Designer Window

The following illustration shows Layout Designer and a sample layout.


Layout Table Fields Contents Tasks Discussions History Properties Archive Queries Charts managerView

Standard

(Conflict error message - usually empty, but possibly filled when form is submitted)

Uniform Residential Loan Application

Submit

This application is designed to be completed by the applicant(s) with the lender's assistance. Applicants should complete this form as "Borrower" or "Co-Borrower", as ap
Co-Borrower information must also be provided (and the appropriate box checked) when ☐ the income or assets of a person other than the "Borrower" (incl
Borrower's spouse) will be used as a basis for loan qualification or ☐ the income or assets of the Borrower's spouse will not be used as a basis for loan qualificatio
or her liabilities must be considered because the Borrower resides in a community property state, the security property is located in a community property state, or the
is relying on other property located in a community property state as a basis for repayment of the loan.

I. TYPE OF MORTGAGE AND TERMS OF LOAN

Mortgage Applied for: ☐ VA ☐ FHA ☐ Conventional ☐ FmHA ☐ Other:
Agency Case Number
Lender Case Number
Amount $
Interest Rate %
No. of Months
Amortization Type: ☐ Fixed Rate ☐ GPM ☐ Other (explain): ☐ ARM (type):

II. PROPERTY INFORMATION AND PURPOSE OF LOAN

Subject Property Address (street, city, state & zip code)
Street City State Zip
County
Legal Description of Subject Property (attach description if necessary)
Year Bu
Purpose of Loan ☐ Purchase ☐ Refinance ☐ Construction ☐ Construction-Permanent ☐ Other (explain):
Property will be: ☐ Primary Residence ☐ Secondary Residence ☐ In

Complete this line if construction or construction-permanent loan.
Year Lot Acquired | Original Cost $ | Amount Existing Liens $ | (a) Present Value of Lot $ | (b) Cost of Improvements $ | Total (a+b) $

Complete this line if this is a refinance loan.
Year Acquired | Original Cost $ | Amount Existing Liens $ | Purpose of Refinance | Describe Improvements ☐ made ☐ to | Cost: $

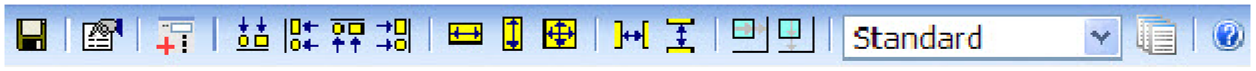
Title will be held in what Name(s)
Manner in which Title will be held
Estate will be ☐ Fee Sim

Components
Current Container: defaultContainer
STREET3 (Text)
SUBPROP_CITY (Text)
SUBPROP_COUNTY (Text)
SUBPROP_NO_UNITS (Integer)
SUBPROP_STATE (Text)
SUBPROP_STREET (Text)
SUBPROP_ZIP (Text)
TERM (Integer)
TITLE_HELD_IN_NAME (Text)
TITLE_HELD_IN_NAMES2 (Long Te
TITLE_HELD_INNAMES2 (Text)
TITLE_HOLDER (Text)
TOTAL_A_B (Decimal)
UNDERWRITER (Text)
UNMARRIED (Boolean)
UNMARRIED2 (Boolean)
VA_LOAN (Boolean)
WORKPHONE (Text)
YEAR_ACQUIRED (Text)
YEAR_ACQUIRED2 (Text)
YEAR_BUILT (Text)
YEARS_SCHOOL (Integer)
YEARS_SCHOOL2 (Integer)
YRS_ON_JOB1 (Integer)
YRS_ON_JOB2 (Integer)
YRS_PROFESSION1 (Integer)
YRS_PROFESSION2 (Integer)
ZIP (Text)
ZIP3 (Text)
Containers
Standard container
Others
Conflict error message
Horizontal line
HTML
Image
JSP
Submit
Text

The main elements of the Layout Designer window are the toolbar, the canvas, and the components list.

Toolbar

The toolbar provides the tools that are used to define a layout.



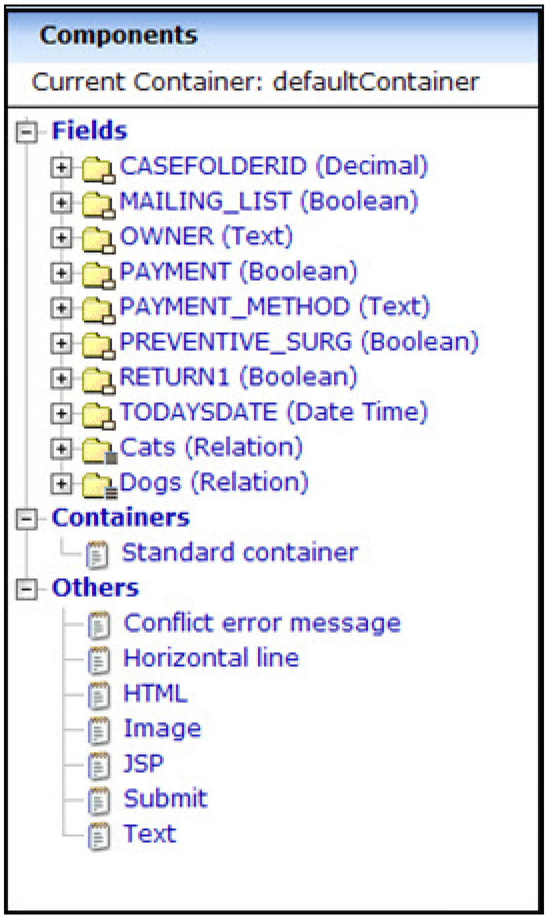
Using the toolbar, you can save the current layout to the server; edit its properties; add field components; size, align, and distribute components; switch to a different layout; create, copy, rename, delete, and select the default layouts.

Canvas

The canvas is the main area of the window where you draw your layout.

Components List

The Components list is a floating dialog box that displays all of the components that can be added to a layout. The components are displayed in a tree hierarchy. The components list can be moved by dragging the title bar and resized by dragging the edges of the window. Clicking a component adds it to the canvas.



Components

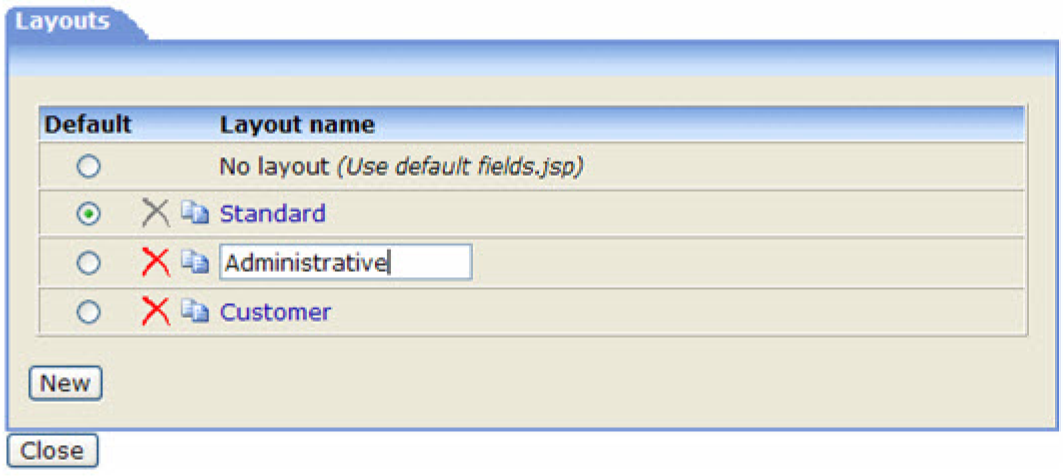
Current Container: defaultContainer

- Fields
  - CASEFOLDERID (Decimal)
  - MAILING_LIST (Boolean)
  - OWNER (Text)
  - PAYMENT (Boolean)
  - PAYMENT_METHOD (Text)
  - PREVENTIVE_SURG (Boolean)
  - RETURN1 (Boolean)
  - TODAYSDATE (Date Time)
  - Cats (Relation)
  - Dogs (Relation)
- Containers
  - Standard container
- Others
  - Conflict error message
  - Horizontal line
  - HTML
  - Image
  - JSP
  - Submit
  - Text

Multiple Alternative Layouts

A table can have any number of alternative layouts defined. These alternative layouts can be used to present the same underlying information in different ways. The application designer can present different layouts to users based on their roles or at different points in the life cycle of an object.

The Manage Layouts dialog box enables an application designer to create, copy, rename, and delete layouts. Any layout can be defined as a default layout. This typically depends on the users or roles assigned to a user. In a process, each queue activity can have a different layout. Layouts can be used by defined custom framesets, button sets, or event handlers.



Default	Layout name
○	No layout *(Use default fields.jsp)*
●	Standard
○	Administrative
○	Customer

There are several ways to use alternative layouts in an application. The following list describes some of the more commonly used techniques:

Create a Custom Menu - If you create a custom tab menu with a Fields tab, you can specify which layout is to be used to display the fields.

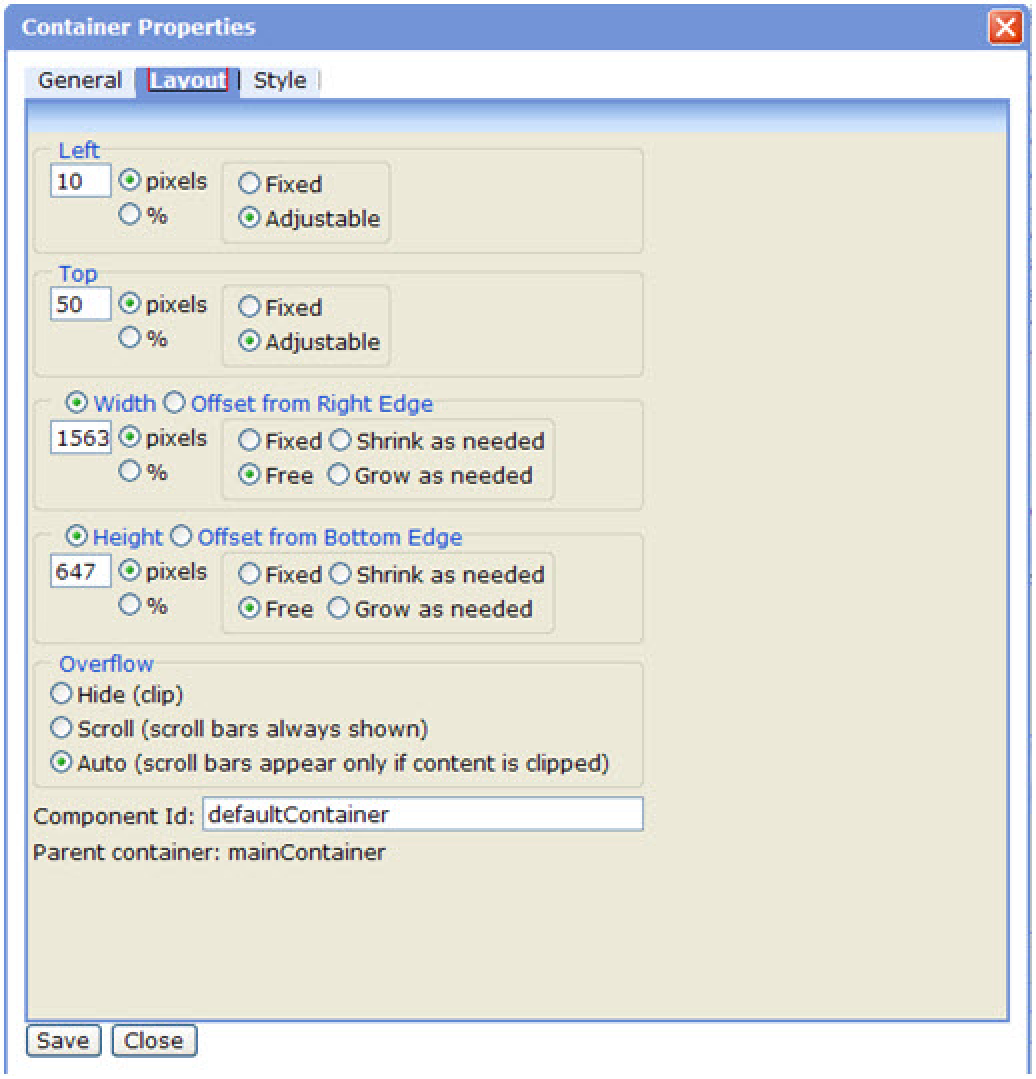
Create a Custom Casefolder Frameset - If you have a custom frameset that displays the fields in a dedicated frame, you can specify which layout is to be used.

Define a Custom Activity User Interface - You can specify the layout to be used in a given process activity in the Process Designer using the form under the Options tab in a Queue activity's properties dialog box.

Define an OnTable Event Handler - You can use any of the information associated with an object, the current user, or system configuration options to dynamically select an alternative layout for any table.

Layout and Component Properties

Each component in a layout has properties that are accessed by opening the component. These properties control how the layout will display when an instance is created and can affect the whole layout. With a couple of exceptions there are three tabs for properties: a General tab, a Layout tab and a Style tab. The General tab is unique to each component type. The component's properties Layout tab enables an application designer to specify exactly how the component is to be positioned and sized in the layout.



Left
10 [✓ pixels] %
Fixed [✓ Adjustable]

Top
50 [✓ pixels] %
Fixed [✓ Adjustable]

[✓ Width] Offset from Right Edge
1563 [✓ pixels] %
Fixed Shrink as needed [✓ Free] Grow as needed

[✓ Height] Offset from Bottom Edge
647 [✓ pixels] %
Fixed Shrink as needed [✓ Free] Grow as needed

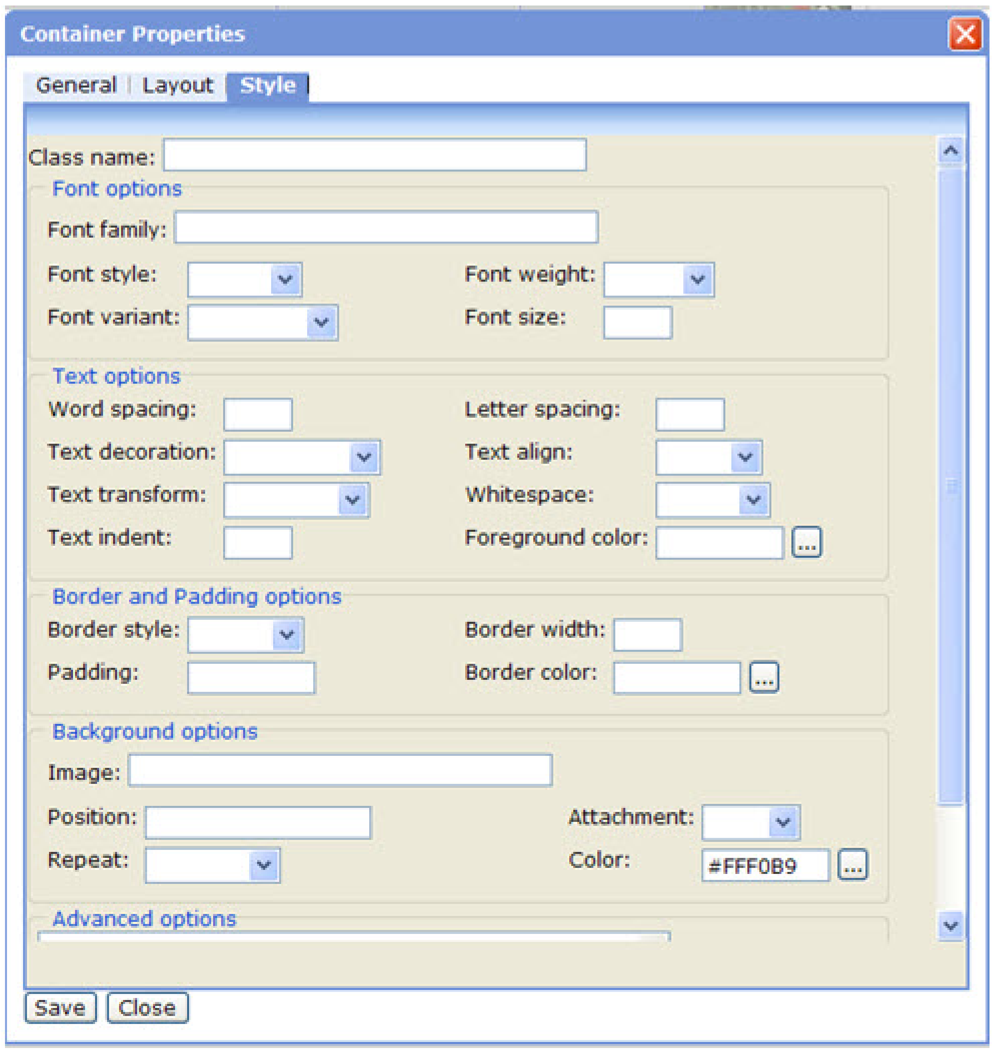
Overflow
Hide (clip)
Scroll (scroll bars always shown)
[✓ Auto (scroll bars appear only if content is clipped)]

Component Id: defaultContainer

Parent container: mainContainer

The Style tab enables an application designer to specify exactly how the component is to appear in the layout. While anyone can use the style options to define the appearance of a component, an understanding of the W3C CSS specification enables an application designer to take full advantage of the power of this capability.

The most important concept for the general user is to know that some style options specified on a container are applied to all of the components it contains (including nested containers). Careful use of containers in a layout can greatly simplify the application of visual styles. Several of the style options on the Style tab also enable an application designer to specify a color to be used when displaying a component.



Class name:

Font options

Font family:			
Font style:		Font weight:	
Font variant:		Font size:	

Text options

Word spacing:		Letter spacing:	
Text decoration:		Text align:	
Text transform:		Whitespace:	
Text indent:		Foreground color:	

Border and Padding options

Border style:		Border width:	
Padding:		Border color:	

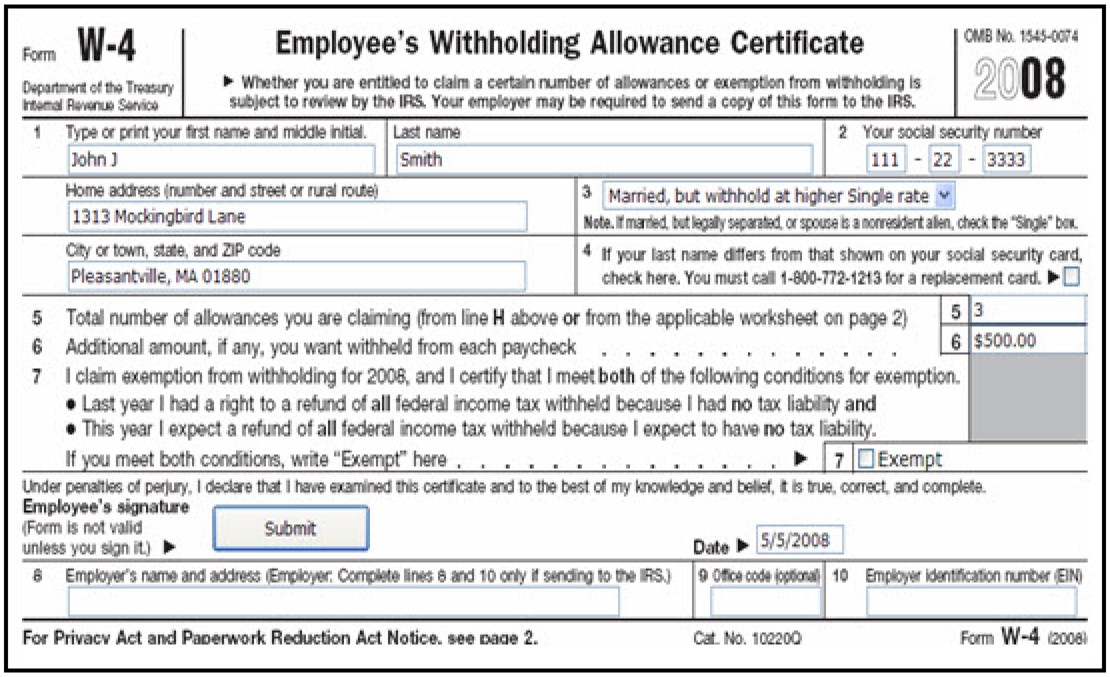
Background options

Image:			
Position:		Attachment:	
Repeat:		Color:	#FFF0B9

Advanced options

Forms with Background Images

An application designer can use background images to quickly generate professional looking versions of printed forms. The following illustration shows a sample form layout that uses a background image (this is a US tax form). Notice that the input elements are positioned at the appropriate locations over the background form image.



Form **W-4**
Department of the Treasury
Internal Revenue Service

Employee's Withholding Allowance Certificate

▶ Whether you are entitled to claim a certain number of allowances or exemption from withholding is subject to review by the IRS. Your employer may be required to send a copy of this form to the IRS.

OMB No. 1545-0074
2008

1 Type or print your first name and middle initial. John J
Last name Smith
2 Your social security number 111 - 22 - 3333

Home address (number and street or rural route) 1313 Mockingbird Lane
3 Married, but withhold at higher Single rate
Note. If married, but legally separated, or spouse is a nonresident alien, check the "Single" box.

City or town, state, and ZIP code Pleasantville, MA 01880
4 If your last name differs from that shown on your social security card, check here. You must call 1-800-772-1213 for a replacement card. ▶ ☐

5	Total number of allowances you are claiming (from line **H** above **or** from the applicable worksheet on page 2)	5 3
6	Additional amount, if any, you want withheld from each paycheck	6 $500.00

7 I claim exemption from withholding for 2008, and I certify that I meet **both** of the following conditions for exemption.
- Last year I had a right to a refund of **all** federal income tax withheld because I had **no** tax liability **and**
- This year I expect a refund of **all** federal income tax withheld because I expect to have **no** tax liability.

If you meet both conditions, write "Exempt" here . . . . ▶ 7 ☐ Exempt

Under penalties of perjury, I declare that I have examined this certificate and to the best of my knowledge and belief, it is true, correct, and complete.
Employee's signature
(Form is not valid unless you sign it.) ▶ Submit
Date ▶ 5/5/2008

8 Employer's name and address (Employer: Complete lines 8 and 10 only if sending to the IRS.)
9 Office code (optional)
10 Employer identification number (EIN)

For Privacy Act and Paperwork Reduction Act Notice, see page 2. Cat. No. 10220Q Form **W-4** (2008)

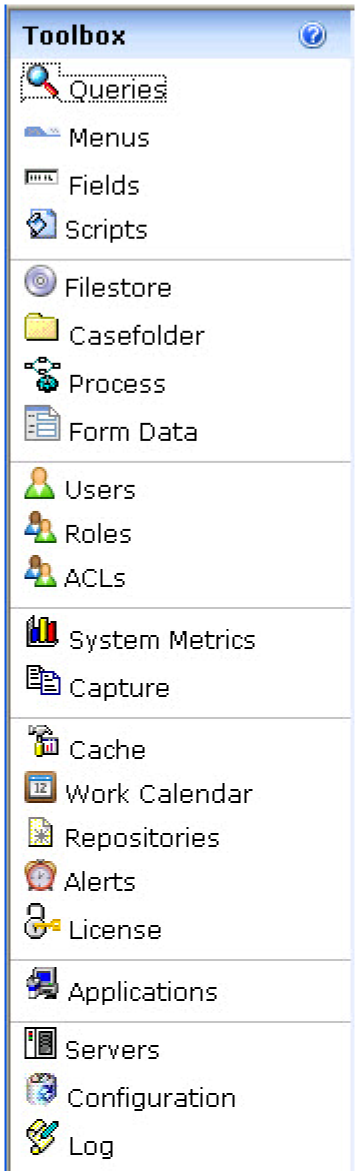
Chapter 9

Tools

System administrators and application designers use the Case360 tools to perform various functions.

Administration and Design Tools

Administrators and designers use the Toolbox to perform the tasks required to keep the system running smoothly and to create and maintain applications. Use of these tools requires appropriate permissions.



The following list describes the tools:

Queries - Displays the list of available queries. Enables authorized users to run, create, and modify queries.

Menus - Enable application designers to create customized menus that appear as popup menus, tabbed dialog boxes, or icon button sets.

Fields - Displays the fields in the schema. Enables application designers to add new fields and modify the definition of existing fields.

Scripts - Enables the creation of scripts that can be reused in a variety of contexts to define custom system behaviors.

Filestore - Enables authorized users to create, modify, and delete Filestore templates, storage controllers, and data managers.

Casefolder - Enables authorized users to create, modify, and delete Casefolder templates.

Process - Lists all the processes installed on the server and their status. Enables authorized users to view process fields, view and modify properties, start and stop each process, view and download the process definition file, and view metrics specific to that process.

Form Data - Enables the application designer to manage Form Data templates that specify options for how the form will behave.

Users - Lists the Case360 users and enables authorized users to view and create users. User attributes are created using the Roles tool.

Roles - Lists the Case360 roles and enables authorized users to create and modify roles and user attributes.

ACLs - Lists the Case360 ACLs and enables authorized users to create and manage ACLs for security and to define rules for accessing objects.

System Metrics - Displays the values of various system counters, providing administrators with a view of system activity.

Capture - Lists the Capture processes that are running and their status. Enables authorized users to start and stop the Capture threads and to import and export the required descriptor files. Application designers can create new capture packages that take advantage of script output, mail input, and message input as well as file input and generic output.

Cache - Displays in-memory caches maintained by the Case360 Server. Enables administrators to clear individual caches or all caches.

Work Calendar - Enables administrators to define the start and end of the workday and exclude days that are not part of the business day calculations. Process and Casefolder deadline functions use the calendar information to compute business hours, business days, and business minutes.

Repositories - Enables authorized users to create and configure external repositories and to apply access control lists to internal repositories.

Alerts - Enables authorized users to start and stop the Alerts manager.

License - Displays license information for each software component (Casefolder, Filestore, and Process). Enables authorized users to update the licenses, create license groups, and display license metrics.

Applications - Enables application designers to create new applications that include all of the components required for the application to run on a Case360 system. The application can include templates, queries, scripts, menus, capture packages, work calendars, roles, fields, field presentation formats, event handlers, ACLs, icons and other supporting files and can be exported and imported on any number of systems. From the Actions menu, the Applications tool also provides Sample applications for the Enterprise Document Access Connectors and Select Multiple toolbars.

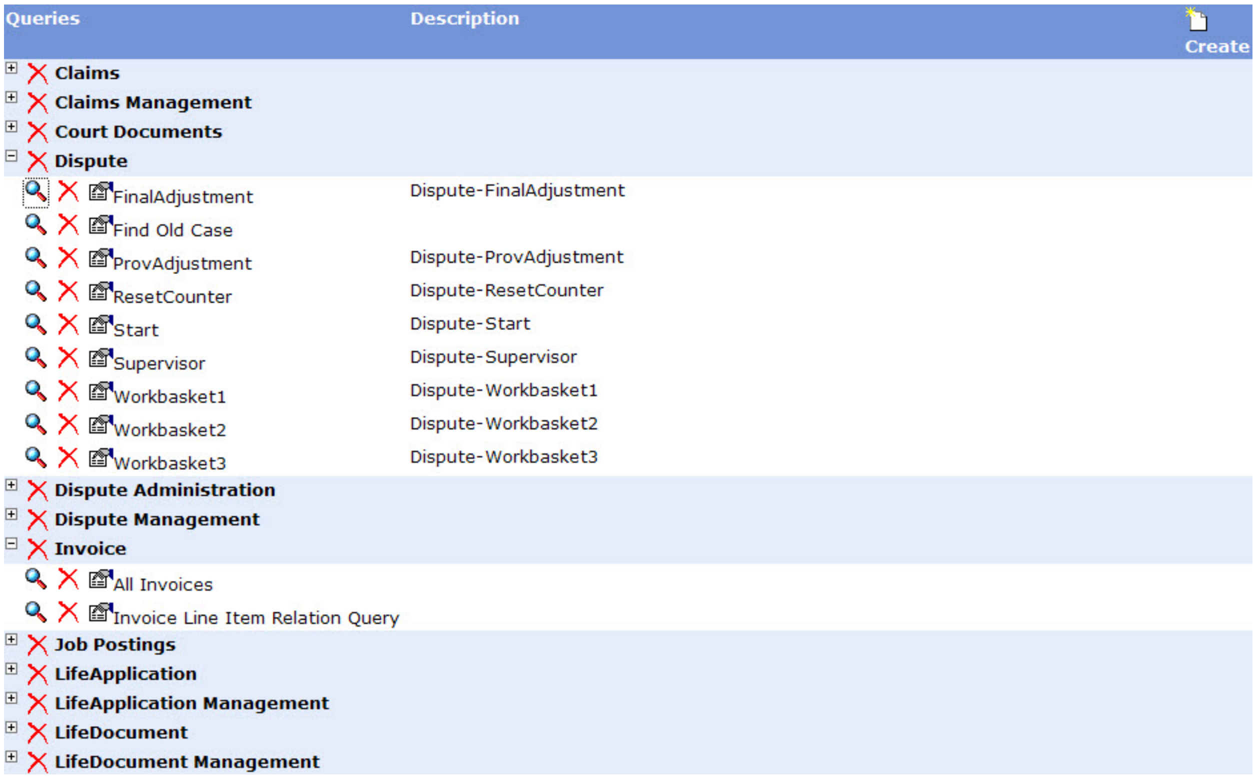
Servers - Displays a list of Case360 Servers defined by the current configuration.

Log - Displays activity logged by the Case360 Server.

Queries Tool

A query is the primary method for accessing objects in installed repositories. Application designers use the Queries tool to create and modify queries. Queries can be organized in groups that users can expand or collapse. Output definitions can be copied from one query to another.

Access to queries is controlled through the use of ACLs, which the application designer assigns at creation. Users see only the queries that they are authorized to run. With persona-based home pages, queries can have a visibility option as well as an ACL. The visibility option does not replace the ACL but can be used as an alternative when using the persona-based home pages.



Queries	Description	Create
Claims		
Claims Management		
Court Documents		
Dispute		
FinalAdjustment	Dispute-FinalAdjustment	
Find Old Case		
ProvAdjustment	Dispute-ProvAdjustment	
ResetCounter	Dispute-ResetCounter	
Start	Dispute-Start	
Supervisor	Dispute-Supervisor	
Workbasket1	Dispute-Workbasket1	
Workbasket2	Dispute-Workbasket2	
Workbasket3	Dispute-Workbasket3	
Dispute Administration		
Dispute Management		
Invoice		
All Invoices		
Invoice Line Item Relation Query		
Job Postings		
LifeApplication		
LifeApplication Management		
LifeDocument		
LifeDocument Management		

Queries can run against repositories or database tables. A query can also display a form to receive user input. When the form is displayed, the user enters the appropriate parameters and the query runs with that information.

You can run queries against the internal repositories (Filestore, Casefolder, Form Data, Process, and User Attributes) or against external repositories that are configured with an Enterprise Document Access Connector supported by Case360. A query returns a list of repository objects that meet the specified criteria.

A single query can run one or more subqueries against different repositories or database tables in the Case360 database. These subqueries are executed in parallel and their results are merged into a single set of results for the user. If your server is configured to use XA data sources, you can also create data queries that query external databases by specifying a data source name.

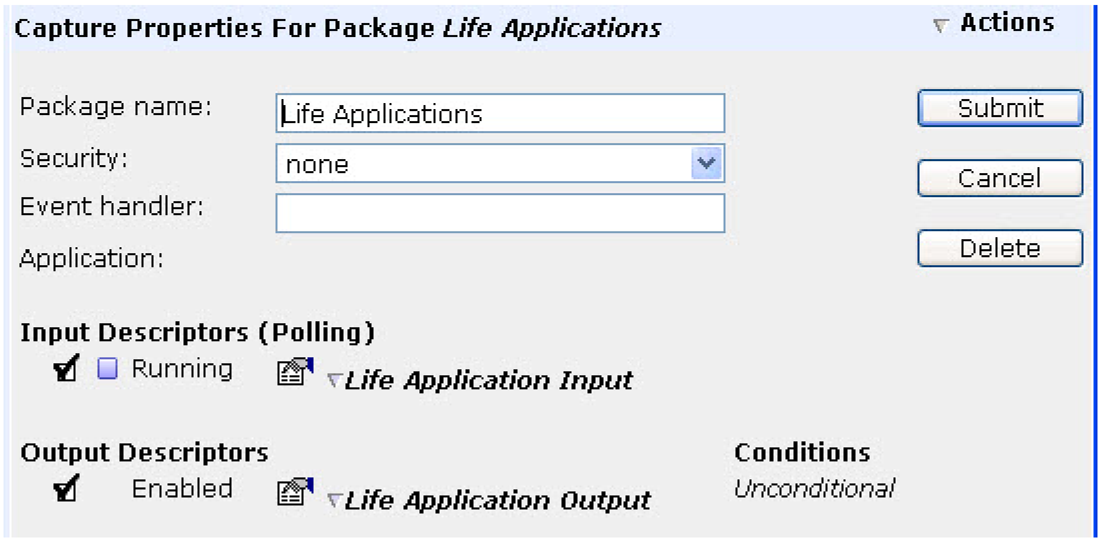
Query results can be output to a table, a graph, JSP display, a URL display, or a Select Multiple display.

- When output to a table, data is displayed in rows and columns. Options are available to define how the results will be displayed. For instance, you can page through results using Previous and Next.
- Graph options let you specify formats such as pie charts or bar graphs and automatic scheduling of the result data.
- Using JSP display as output enables application designers to format query results in any way they like by creating a custom query JSP.
- Selecting URL display as output permits the display of management reports from the managerView Dashboard Server powered by Tableau. In addition any URL can be used to display reports or charts.
- Choosing Select Multiple output enables the user to select any number of entries from the result list and click a button in a toolbar to perform an action on all of the selected objects.

Special process queries called Get Next queries push instances to a user's desktop using a pre-defined user interface. Get Next queries are created using the Query tool.

Capture

Capture processes input files into objects that can be used by the Case360 system.



Capture Properties For Package *Life Applications*

Package name: Life Applications

Security: none

Event handler:

Application:

Input Descriptors (Polling)

✓ ☐ Running *Life Application Input*

Output Descriptors **Conditions**

✓ Enabled *Life Application Output* *Unconditional*

You can create a capture package that encapsulates all the capture rules for an application in a form that can be included as part of the application export and import process. This simplifies the deployment of multiple applications on an enterprise server cluster. Capture packages have an associated ACL that controls who can manipulate the capture rules.

All capture configuration information is stored in the database to simplify management and enforce consistency across the servers in a cluster.

The Capture tool provides the following actions:

- New FileInput
- New MailInput
- New MessageInput
- New GenericOutput
- New ScriptOutput

JMS Capture

Capture supports the use of the JMS Messaging facility to provide a non-polled real-time means to capture data. As with other Capture Input Drivers, the JMS Input driver can capture data files with or without XML buddy files or even XML files alone.

Systems outside the "core users" of Case360 can submit requests to be processed through application specific Web services. Often this is required where an organization or standards body has adopted a standard Web service interface definition for all systems performing a type of function (for example, insurance claims processing). This capability enables the consumer of the service to talk to any provider without requiring custom integration.

It is the provider’s responsibility (in this example, Case360) to present an interface that can be consumed. An application developer can do this by creating a custom Web service that acts as a plug-in capture driver and provides application specific Web services (for example, create insurance claim). Requests that are received at the create insurance claim Web service are sent straight to Capture Broker, the same as for a JMS capture request. At the end of the process, a response is sent through the Case360 Web service activity to inform the external organization that the process was completed successfully and the customer’s claim was accepted.

Input Files

Input files are available from a variety of sources and the data can be structured in different ways. The following list describes several sources for input files and the types of data they provide:

- High-end image capture systems scan paper documents, process form information, and gather field values before releasing the input files to a folder monitored by capture. Mid-range image capture systems do not gather many field values. Instead, they gather basic field values, such as date and time captured, batch number, document number, page number, number of pages, and bar code information from documents. In many applications, bar code and index information may be sufficient for capturing required information. Desktop scanning software can capture, process, and output individual documents. The resulting files may include a complete set of field values or no field values.

  G360 Scan Manager, Kofax® Capture®, and Captiva InputAccel® are examples of high-end image capture systems. G360 Scan Manager can be used as either a high-end or mid-range image capture system. G360 Imaging for Windows and Flow (a component of G360 Imaging for Windows) are examples of desktop capture systems.

- Fax servers can capture inbound faxes but they typically do not gather many field values during processing. The field values they do capture include a Direct Inward Dial (DID) number and the Call Subscriber ID (CSID).
- E-mail agents can capture e-mail messages arriving at an inbox and they typically do not capture many field values during processing. The field values they do gather include the sender's e-mail address, date and time sent, subject text, and message body.

Extending Capture

Capture provides an extensible framework that supports any number of different content sources and destinations. Each descriptor specifies the Java class that contains the code to be used to perform the input or output processing. In addition, each descriptor defines an options string that specifies any necessary parameters.

Configuration Tool

The Configuration Tool provides an intuitive user interface for specifying most configuration properties. These properties are stored in the database where they are shared by all the servers in an enterprise cluster. The cache manager automatically synchronizes configuration changes across the cluster. Most configuration changes do not require a server restart.

Menu Editor

Using the Menu Editor tool, application designers can create and modify customized menus that provide access to various actions without writing programming code. A menu can be presented as a pop-up menu, tabbed menu, or icon button set.

Each menu has the following properties:

Menu Name - the name that is used to attach the menu to its user interface.

ACL - a list that controls who can view and modify the menu. This supports division of responsibility between applications and application designers.

Intended menu type - the intended usage for the menu (button set, tabbed menu, or pop-up menu).

Menu help URL - a URL to link to a help icon in the menu (used only for tabbed menus). If this property is not specified, no help icon is displayed.

Each action on a menu has the following properties:

Name - the internal script name for the action.

Display name - the name that appears in the menu for the action. Display names can be stored in a properties file, enabling them to be localized and presented in more than one language.

Tooltip - the tooltip to be displayed when a user holds the cursor over the menu entry. Tooltips can be stored in a properties file, enabling them to be localized and presented in more than one language.

URL - the URL to invoke when a user selects the menu entry.

Icon - an icon to associate with the menu entry.

Scripting Tool

By enabling you to create named scripts that can be reused in a variety of contexts, the Scripts tool provides the key for quickly and easily implementing custom system behaviors. You can specify an expression that is evaluated to control a specific behavior or a series of actions to be performed that causes a desired behavior. In many cases, you simply enter the expression or script into the appropriate field of the tool you are using. Scripts are organized into a hierarchy within the following top-level groups.

Scripts - general callable script functions that are organized in a hierarchy of user-defined packages. The application designer defines the package hierarchy.

Types - includes all of the data types defined in the grammar and represents an object type derivation (super type to subtype inheritance) hierarchy. Application designers can add script methods at any level of the hierarchy.

Events - event handler scripts. The first level of the events subtree lists all of the system defined event interfaces. The second level lists all the user defined event handler names. The third level lists the event scripts themselves.