BMC® Remedy® Action Request System® 7.0

Concepts

bmcsoftware

May 2006
Part No: 58459

Restricted Rights Legend

Contacting Us

If you need technical support for this product, contact Customer Support by email at `support@remedy.com`. If you have comments or suggestions about this documentation, contact Information Development by email at `doc_feedback@bmc.com`.

This edition applies to version 7.0 of the licensed program.

BMC Software, Inc.
`www.bmc.com`

Contents

Preface

When Remedy® first introduced the BMC® Remedy® Action Request System®® (AR System®) in 1991, it quickly gained acceptance as a leader in the help desk arena. As companies, large and small, began using the AR System, they realized that its use extended far beyond just the Remedy Help Desk application—that it could, in fact, be used to automate, track, and manage any business process. Today, the AR System is used as the foundation for a wide range of departmental and enterprise-wide solutions, from help desk call tracking to inventory management to integrated systems management.

This book focuses on the core concepts of the AR System. Read this book to familiarize yourself with the components and architecture of the AR System. Procedures, performance, and other topics are documented in related books listed in the appendix.

The book is written primarily as a guide for beginning administrators who will use the AR System to create or modify applications, although other audiences, including business managers and those evaluating and prototyping applications with the AR System, will also find the information valuable.

Your role as an AR System administrator

As an administrator, you might have many roles. Depending on your level of expertise, you might be responsible for some or all of the following areas:

- Determining how to allocate server and database resources.
- Defining your group's work processes and business rules.
- Designing and implementing an AR System application that reflects your group's work processes and business rules or working with a consultant to create an application.
- Localizing an AR System application for use in other languages or countries.
- Installing AR System software.
- Modifying an AR System application to reflect changes in your group's work processes.
- Maintaining the AR System by performing such tasks as adding and deleting users, backing up AR System servers, and importing data from other systems.

How to use this book

If you are new to the AR System, you will find it useful to read the book sequentially, since each chapter builds upon the previous one. However, if you are already familiar with the AR System, you can go directly to a chapter that interests you.

The book is organized into eight chapters and an appendix:

- Chapter 1, "Overview," is a high-level view of the AR System. It lays the groundwork for the rest of the book, discussing basic terms and concepts relevant to the AR System.
- Chapter 2, "Using forms," discusses forms—the core element of the AR System that captures and displays crucial information needed for your business. The chapter describes what forms are, the different types of forms, the fields that make up a form, and how forms can be grouped into applications for deployment on Windows and the web.

- Chapter 3, “Using menus as automation aids,” describes the different types of menus and how you can use menus to help users fill in information about their forms.
- Chapter 4, “Introduction to workflow,” introduces the workflow components—active links, filters, and escalations—that enable you to automate your business processes.
- Chapter 5, “Workflow actions and firing conditions,” examines each workflow component in detail, describing the actions that active links, filters, and escalations can perform, followed by the firing conditions that trigger the actions.
- Chapter 6, “Controlling access to AR System,” discusses access control in the AR System, describing users and groups, the multi-tiered access control model, and licensing.
- Chapter 7, “Expanding beyond the core AR System product,” discusses other products available from BMC and third parties that are based on the AR System.
- Chapter 8, “Putting it all together,” brings together the concepts discussed throughout the book. It walks you through the process of creating a sample application for a wild animal park that uses the AR System for a variety of tracking and management needs.
- Appendix A, “For more information,” points you to additional sources of information about the AR System. It includes a list of relevant publications from BMC Remedy, with a brief description of each. It also includes information about BMC Remedy user groups, electronic forums, training courses, and BMC Remedy’s web pages.
- The “Master Glossary” defines key terms used throughout the AR System documentation set. Terms of special significance to administrators are **boldfaced** throughout the book. Many of these are defined in this glossary.

Important: The compatibility information listed in the product documentation is subject to change. See the compatibility matrix at `http://supportweb.remedy.com` for the latest, most complete information about what is officially supported.

Carefully read the system requirements for your particular operating system, especially the necessary patch requirements.

Chapter 1 Overview

Every company, whether it makes bicycles or provides telecommunications services to the entire world, has its own business needs and processes. BMC® Remedy® Action Request System® (AR System®) enables you to automate these business processes without learning a programming language or understanding complex development tools.

This chapter introduces the components and architecture of the AR System and explains how they fit together to address your organization's needs.

What is AR System?

AR System is a professional development environment that is built to leverage the best practices Information Technology Infrastructure Library (ITIL) standards, which you can use to create business workflow applications. Using the AR System, nonprogrammers can build powerful applications once and deploy them on Windows, the web, and wireless environments simultaneously.

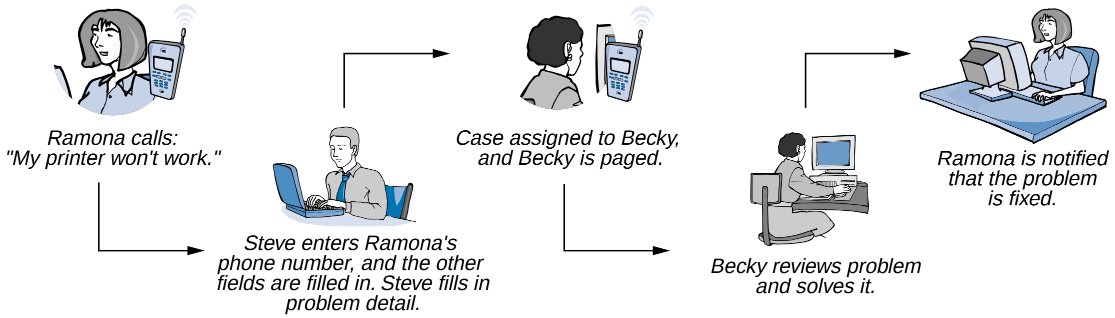
The AR System enables you to automatically keep track of anything that is important to the processes in your enterprise. Companies have used the AR System to track such diverse items as stock trades, benefits data, inventory assets and spare parts, and order fulfillment. One of the most common uses of the AR System is to automate the internal help desk. The following example illustrates a help desk solution in which the AR System is used to solve an employee's problem. The next section describes the components of the AR System and explains how each of the components is used in this example.

Example of a help desk solution, Part 1

At a hypothetical company, Ramona could not get her printer to work, so she called the employee assistance number and left voice mail outlining the problem. The first-level help desk employee, Steve, entered Ramona's telephone number into the blank form on his screen, pressed the Enter key, and the details of Ramona's configuration and location were automatically loaded onto the screen. Steve then filled in the remaining details about the problem and entered the new case into the AR System. He immediately received a message on his screen that the case had been assigned to Becky.

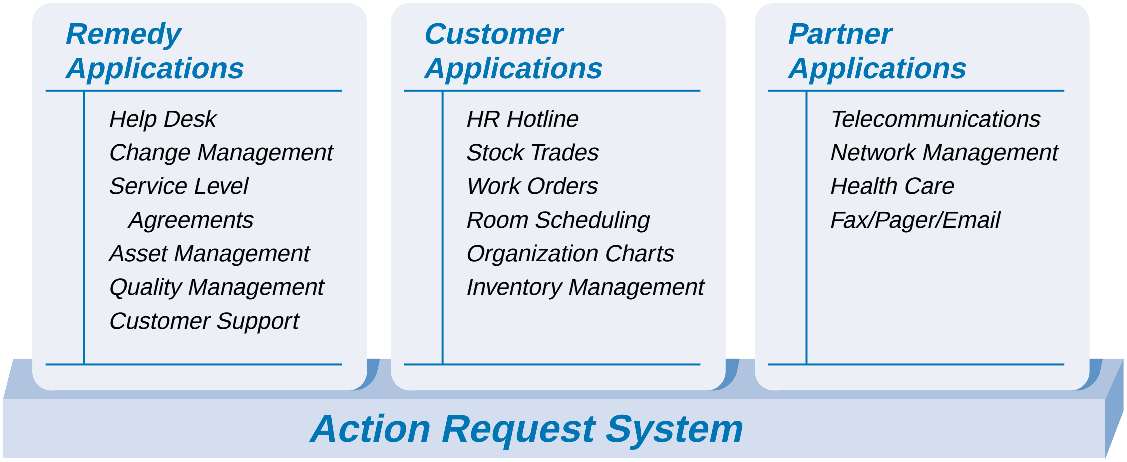
Becky was automatically paged, and reviewed the problem on her system. Using her knowledge of similar problems, she fixed the printer and marked the case as closed, at which point Ramona was notified that the problem was fixed. If Ramona's problem had been an emergency and not been handled within an hour, the system would have automatically paged all the appropriate support personnel and would have sent email to Ramona informing her of the status.

Figure 1-1: Example help desk scenario


Ramona calls: "My printer won't work."
Steve enters Ramona's phone number, and the other fields are filled in. Steve fills in problem detail.
Case assigned to Becky, and Becky is paged.
Becky reviews problem and solves it.
Ramona is notified that the problem is fixed.

A help desk application such as the one in this example as well as other pre-built AR System applications are available from BMC and its world class partners. It is also easy to create your own custom solutions.

Figure 1-2: AR System and other applications



The AR System includes intuitive software tools—for nonprogrammers—that enable you to readily customize existing AR System applications or to build your own applications. Specifically, the product consists of:

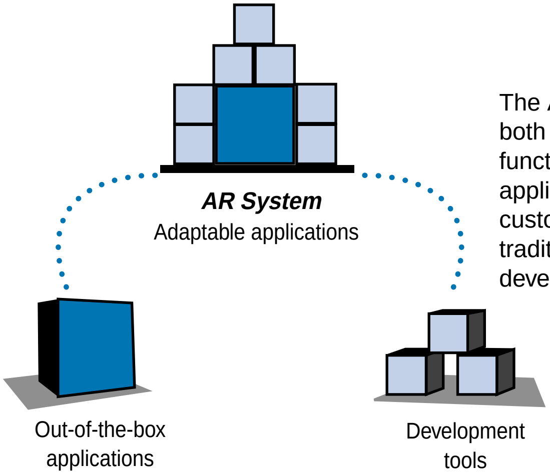
- The AR System server (including the AR System application programming interface [API]), which is connected to one or more data sources.
- Various add-in services, such as a Java server page (JSP) engine, which allow users to view your application on the web. (A third-party web server is required for full web deployment.)
- Various client tools (including BMC Remedy User, BMC Remedy Administrator, BMC Remedy Alert, and BMC Remedy Import) that are used to run, manage and build workflow applications. These client programs are discussed later in this chapter.

  Additional applications and options such as BMC Remedy Help Desk, Distributed Server Option, and BMC Remedy Flashboards are also available for purchase. These and other modules are discussed in Chapter 7, "Expanding beyond the core AR System product."

Perhaps the best way to understand the AR System is through an example. Paul owns a small video store and buys the AR System to help him track inventory. He builds a simple form to track his stock which includes the title, rating, format, type, number of copies, and the rental price. When his business grows, he adds a form to track employees which includes their employee number, salary, start date, training, and a time tracking timecard. Eventually Paul links his application to a Credit/Debit verification system through the open API. Later he adds a order tracking/purchasing application to automatically order items when they are released from the various vendors through the use of Web Services. At that time, he opens a website to enable his customers to order movies through their browser, pay their bills online, and connect their choices to a knowledge base where their rental habits can be stored. Based on this information, the system can be automated to provide proactive movie suggestions. Due to the rapid growth of his business and the flexible, adaptable architecture of the AR System, Paul opens new stores in cities across the country. He is able to link all the stores into one system and keep track of his entire operation through the use of real-time graphic Flashboards in one highly scalable application. Paul is able to track incidents, inventory, HR, Order processing, and customer satisfaction from his office knowing that he can easily extend or modify his system whenever changes occur in his organization.

This kind of adaptability differentiates the AR System from both out-of-the-box solutions—which are typically inflexible—and development toolkits—which require extensive technical knowledge and time to develop applications. The AR System strikes a balance between these two environments by providing a foundation from which you can modify existing applications or create your own robust applications to fit the unique nature of your enterprise.

Figure 1-3: Ways to use AR System



The AR System provides both the ready-to-use functionality of out-of-the-box applications and the customization power traditionally associated with development tools.

Whether you plan to build an entirely new application using the AR System or to simply modify an existing application, you need to understand the components and architecture of the product.

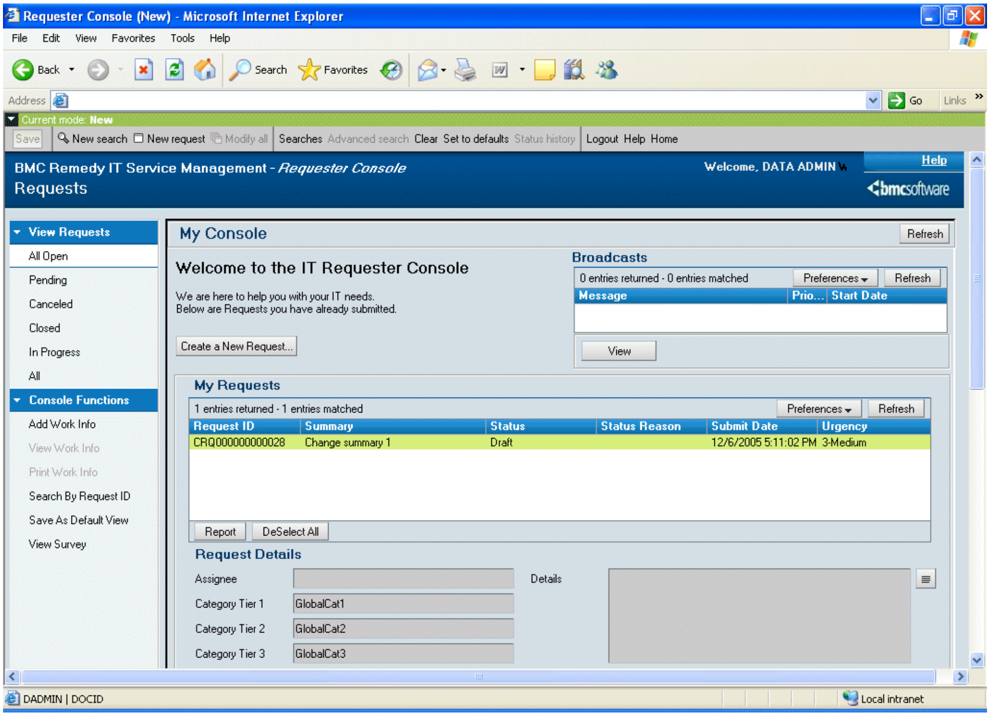
Components

This section introduces the main components of the AR System. All of these components, with the exception of fields and views, are AR System server objects. These components are described in greater detail in subsequent chapters of this book.

The main component with which users interact is a **form**. Each form is composed of fields, which are units of information, such as an employee's last name or the urgency of the request you are making. When the fields are filled in and saved by the user, a request is created for the system to track. Each filled-out form is analogous to a **request**. In database terms, each request is a record. You can design different arrangements, or **views**, of forms appropriate for different user functions.

If you choose, you can bundle a group of related forms into an **application.** For example, a human resources application could include forms for basic employee data, for health benefits, and for salary information. You can name the application and install it on different servers. You can also display your application on the web to allow access from a browser on any platform, as shown in the following figure.

Figure 1-4: Console viewed in a browser



Welcome to the IT Requester Console

We are here to help you with your IT needs.
Below are Requests you have already submitted.

Broadcasts

0 entries returned - 0 entries matched

Message	Prio...	Start Date

My Requests

1 entries returned - 1 entries matched

Request ID	Summary	Status	Status Reason	Submit Date	Urgency
CRQ000000000028	Change summary 1	Draft		12/6/2005 5:11:02 PM	3-Medium

Request Details

Assignee	
Category Tier 1	GlobalCat1
Category Tier 2	GlobalCat2
Category Tier 3	GlobalCat3
Details	

DADMIN | DOCID

Local intranet

Menus are lists that help you fill in fields on forms. A menu can contain all of the possible choices for a field, or it might contain some possible choices and yet allow you to fill in something not on the menu. You can design dynamic menus that change their contents based on the data already on the form.

Forms provide the mechanism to structure data capture and menus offer options for specific field data. Three additional components—active links, filters, and escalations—act on the data to automate business processes—or workflow—in the AR System. The icons in the margin are used in this book to represent these components. All three components perform actions in response to firing conditions that you define. In the AR System, workflow generally refers to the operations performed by these components. However, the product also addresses the broader meaning of workflow, which is the processes your organization uses to run itself and the importance of automating those processes.

An **active link** is an action or group of actions performed on the client, which is the portion of the software with which the user interacts. Active links occur in response to user actions on the screen and can be used for a variety of tasks, such as giving quick responses during data entry and auto-filling fields. For example, an active link can verify the value in the Employee ID field after it is entered and pull information from a supporting People form to fill in the values for other fields on the form such as the Requestor Name, Department, Phone Number, etc., which dramatically reduces the time it takes the support staff to fill out the request. A group of active links can also be combined in an object called an **active link guide. Active link guides** can augment training by assisting a user with the steps necessary to fill in one or more forms to accomplish a specific task. For example, an active link guide could open a business cards form and then display input instructions, field by field, until the request is complete and ready to submit. Active link guides can also be used as subroutines to accomplish common tasks.

A **filter** is an action or group of actions performed on the AR System server. As a request is processed by the server, the filters tied to that form and action are evaluated. Filters are used to enforce business rules and to ensure system and data integrity. For example, you can verify the values in a completed form by using a filter to compare the current transaction values against your business rules and return an error if the request violates any of your processes or procedures such as re-opening a closed ticket. A **filter guide** is a group of filters that can be used as a subroutine in workflow. Because filter guides reside on the server, they cannot be used like active link guides to lead users through completing a form.

An **escalation** is an action or group of actions performed on the server at specified times or time intervals. In a sense, it is an automated, time-based process that searches for requests that match certain criteria at intervals or specified days and times you define and takes actions based on the results of the search. For example, an escalation can notify the next level of management if a problem has not been assigned to a technician within one hour of submission.

You can create and publish a **web service** object that performs a very basic action like creating a record in a form. You can also perform more complicated actions like processing a purchase order that spans multiple AR System forms. You can also create filter workflow to use an external web service to enter data (for example, stock quotes or currency values) from the web service into a form.

Searching for requests

The AR System supports many ways of searching for requests. AR System workflow components can search for requests and act on the results of the search. Clients can use several search methods: query-by-example (QBE), advanced search bar, predefined, and recent.

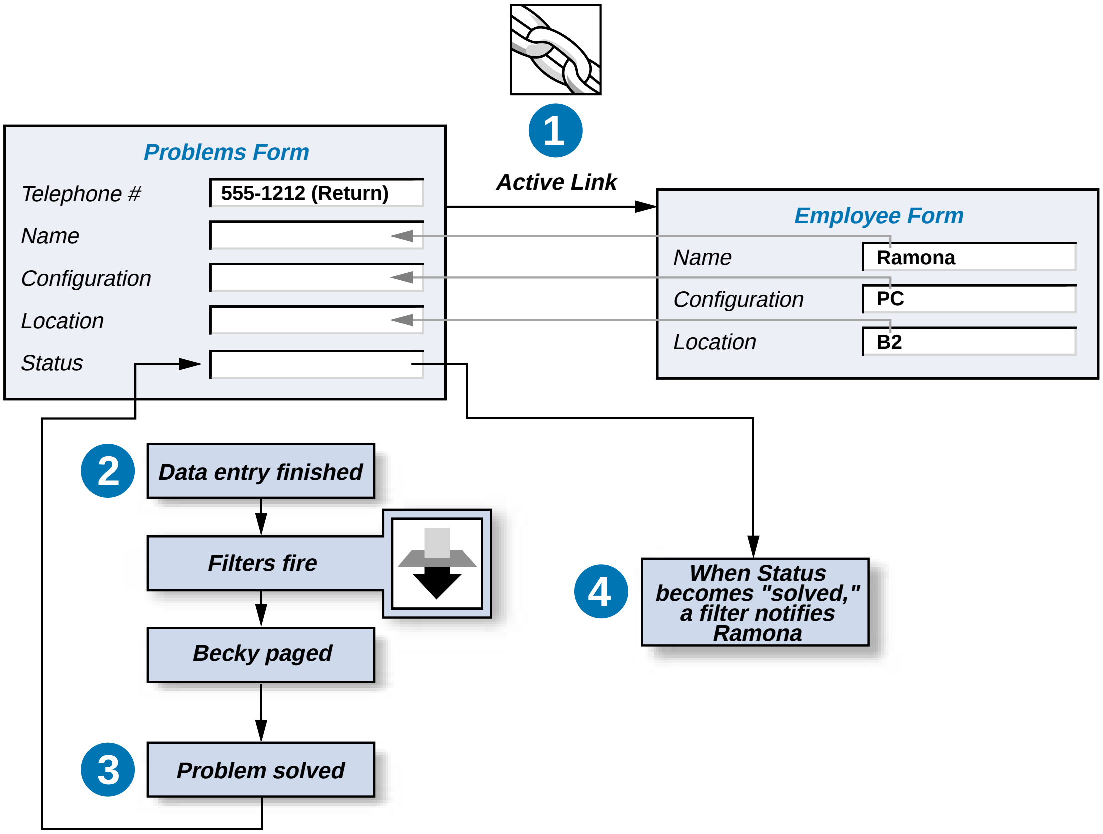
The simplest way to perform a search is to fill in fields and choose to display the matching requests (QBE). For example, if a user named Naomi wants to find all her requests for assistance that are related to a specific printer, she could enter her employee number and the printer name in the appropriate fields, and choose to list the results. The advanced search bar is a part of the screen where the user can enter more complex searches. For example, if Naomi wanted to find all problems related to two printers, she could specify both printer names in the advanced search bar. In addition, the advanced search bar can be used in conjunction with the QBE feature.

The administrator can create predefined searches for searches that are common to a form's users. A user can also define personal searches for forms to which the user has access.

Example of a help desk Solution, Part 2

In the example at the beginning of this chapter, when Steve entered Ramona's telephone number into the Telephone # field, an active link (1) searched the Employee form to retrieve Ramona's name, configuration, and location from another set of forms. After Steve finished the data entry (2), several filters determined that Becky needed to be notified by way of an external paging system integrated with the AR System. After Becky has solved the problem (3), she changes the status of the request, and a filter (4) notifies Ramona that her problem is solved. If the situation had been labeled an emergency and no one had been assigned to the request within an hour, an escalation would have paged all the support personnel required, and a filter would have sent email to Ramona explaining the status.

Figure 1-5: Active link example



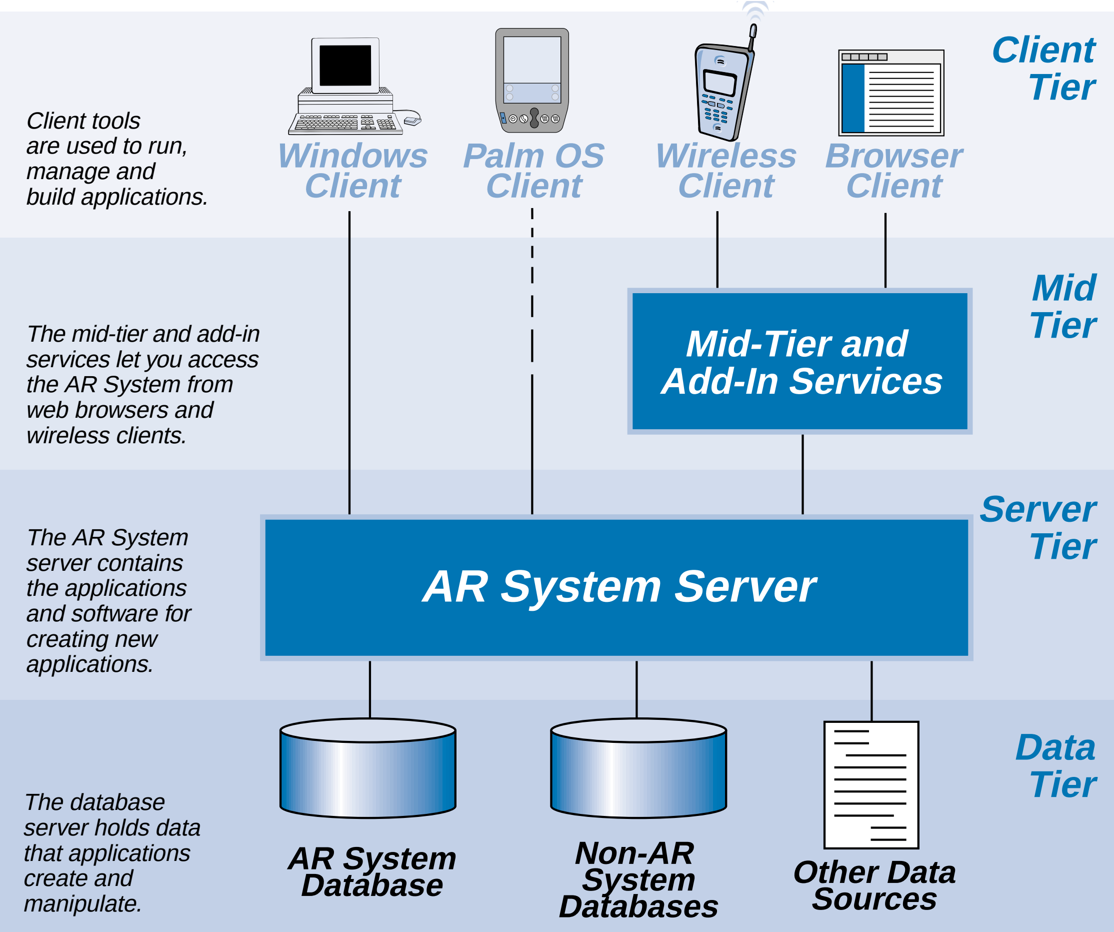
Architecture

The AR System is based on a multi-tier client/server architecture summarized in the figure in this section:

- The client tier is the presentation layer that provides the user interface and user interaction aspects of the product. Most clients are for user interaction, but the tools for system administration are also clients. Clients can run within a browser, on Windows, on Palm OS devices, on wireless devices such as cell phones, and in other environments as well.
- The mid tier provides components and add-in services that run on a web server, enabling users to view applications on the web.

- The **server tier** contains the AR System **server**, which controls workflow processes and access to databases and other data sources in the data tier.
- The **data tier** contains database servers and other data sources that can be accessed by the AR System server. The **database server** acts as the data storage and retrieval engine.

Figure 1-6: AR System architecture


Client tools are used to run, manage and build applications.
Windows Client
Palm OS Client
Wireless Client
Browser Client
Client Tier
The mid-tier and add-in services let you access the AR System from web browsers and wireless clients.
Mid-Tier and Add-In Services
Mid Tier
The AR System server contains the applications and software for creating new applications.
AR System Server
Server Tier
The database server holds data that applications create and manipulate.
AR System Database
Non-AR System Databases
Other Data Sources
Data Tier

AR System clients

AR System clients can be broadly divided into two groups: user clients and administrative clients.

User clients

Several user clients are available that use the standard user interfaces for the web, Windows, Palm OS, and wireless devices within their respective environments.

- Web clients use a browser to provide the user interface to BMC Remedy applications. These applications can be complete e-business web-based solutions for submitting new requests, searching and modifying existing requests, charting data, generating reports, and receiving and responding to AR System notifications.
- BMC Remedy User provides the day-to-day user interface to AR System applications for users of Microsoft Windows. It can be used to submit new and modify existing requests, to search for requests, and to generate reports.
- Wireless clients that use Wireless Markup Language (WML), such as cell phones and two-way pagers, can also communicate with the AR System.
- BMC Remedy Alert, sometimes thought of as a “desktop pager,” can notify users of events by making an icon flash, making an audible “beep,” playing a sound file, running a process, or opening a message window. For example, it can display a message alerting help desk personnel that a new problem has been assigned to them. The user can display a list of alerts in BMC Remedy User, even if BMC Remedy Alert is not installed.

Administrative clients

Administrative clients allow you to create, modify, extend, and set permissions on applications that you create with the AR System. Most administrative clients run on Windows.

- BMC Remedy Administrator is used by AR System administrators to license and configure AR System servers, and to create new and modify existing applications. All of the components that make up an application, such as forms and workflow elements, are created and modified using BMC Remedy Administrator.

- The BMC Remedy Mid Tier Configuration Tool is accessible through a browser. AR System administrators use it to manage the mid tier.
- BMC Remedy Import is used to load external data into AR System forms. For example, employee information could be extracted from a human resources application and loaded into a People Information form as a batch process, eliminating the need to retype data. This tool is also used to import legacy information during the implementation process.
- BMC Remedy Migrator is used by AR System administrators to migrate applications between servers. This tool helps you reduce the difficulty and amount of time required to synchronize your development and production AR System servers.

Integration clients

BMC also offers a number of tools for expanding the capabilities of the core system. These tools act as clients of the AR System and include Enterprise Integration Engines (EIE), Knowledge Based Systems (KBS), Network Management Platform Integration Accessories, and Systems Management Integration Utilities. These are independent parts of the AR System that are separate from the standard desktop tools. For more information about these utilities, see “Expanding beyond the core AR System product” on page 117.

AR System mid tier

The mid tier translates client requests, interprets responses from the server, handles web service requests, and runs server-side processes that bring AR System functionality to web and wireless clients. For example, unlike BMC Remedy User, a browser is a generic client that has no inherent knowledge of any application that might run within it. By acting as an interpreter, the mid tier allows a browser to become a fully functional AR System client.

Note: The BMC Remedy Mid Tier requires a supported Java Server Pages (JSP) engine. ServletExec 5 is bundled with the mid tier and is installed by default, as part of the mid tier installation. See the compatibility matrix on the BMC Remedy Customer Support website for a list of other supported JSP engines.

The mid tier is available for each of the following operating systems, web servers, and JSP engines:

Operating Systems	Web Servers	JSP Engines
HP-UX	Apache, iPlanet	BEA WebLogic, Apache with Tomcat
IBM AIX	Apache, iPlanet	BEA WebLogic, IBM WebSphere, Apache with Tomcat
Linux	Apache	IBM WebSphere, Apache with Tomcat
Microsoft Windows Server	IIS, iPlanet	BEA WebLogic, IBM WebSphere, iPlanet, Apache with Tomcat
Sun Solaris	Apache, iPlanet	BEA WebLogic, IBM WebSphere, iPlanet, Apache with Tomcat

For the latest and most accurate information about supported platforms and software, *always* see the BMC compatibility matrices at `http://supportweb.remedy.com`.

AR System server

The AR System server processes all of the data entered by a client. As the workflow engine between the client and the database server, the AR System server writes data into the database when a request is created, and retrieves that data when a client requests it. The server verifies that a user has permission to do each transaction that is performed, thereby enforcing any access control that you have defined as part of an application. The server also continuously evaluates the data in the database and each transaction to determine whether workflow should be triggered.

The AR System server communicates with the mid tier, AR System clients, and external applications by means of a well-defined application programming interface (API). The server is available for each of the following operating systems:

- HP-UX
- IBM AIX

- Microsoft Windows Server
- Sun Solaris
- Linux

For the latest and most accurate information about supported platforms and software, *always* see the BMC compatibility matrices at `http://supportweb.remedy.com`.

Database servers

The AR System uses standard relational databases for storing and retrieving data. Architecturally, the database server is a set of processes that are completely separate from the AR System server processes. Physically, the database server processes can be running on the same computer as the AR System server or on a different computer. The database server can be run on any platform that the particular database supports.

Because all of the workflow is managed by the AR System server, applications are independent of the database. Therefore, applications created on an AR System server running one type of database can be easily moved to a server running a different database. BMC provides a simple export/import utility for this purpose.

The AR System can use any of the following databases:

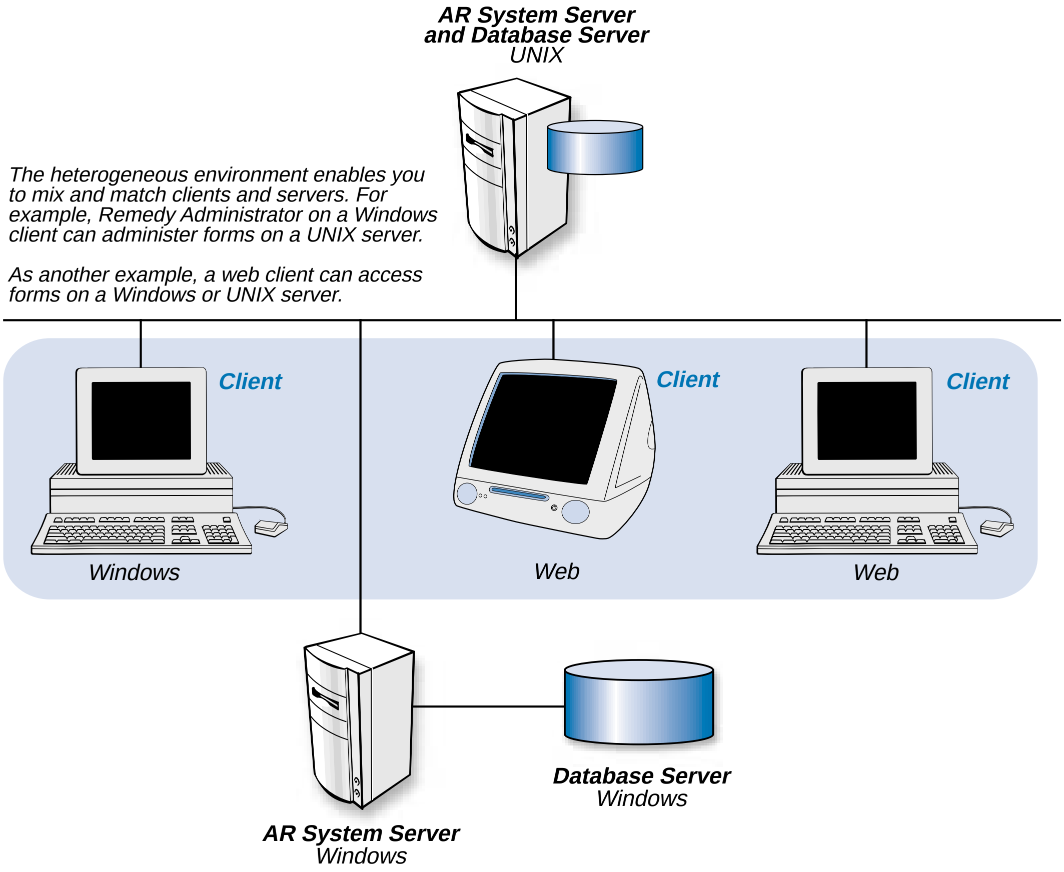
- DB2
- INFORMIX
- Microsoft SQL Server
- Oracle
- Sybase

The AR System can work with data stored in databases and other data sources that are not managed by the AR System. These data sources are accessed through view forms. In addition, the AR System can work with data not stored in databases as if the data were owned locally using the AR System database connectivity (ARDBC) feature.

The heterogeneous environment

Because the multiple layers of the AR System are independent of one another, you can combine platforms to fulfill different functions. The following figure shows how the pieces of the AR System can fit together in a mixed computing environment.

Figure 1-7: AR System in a mixed computing environment



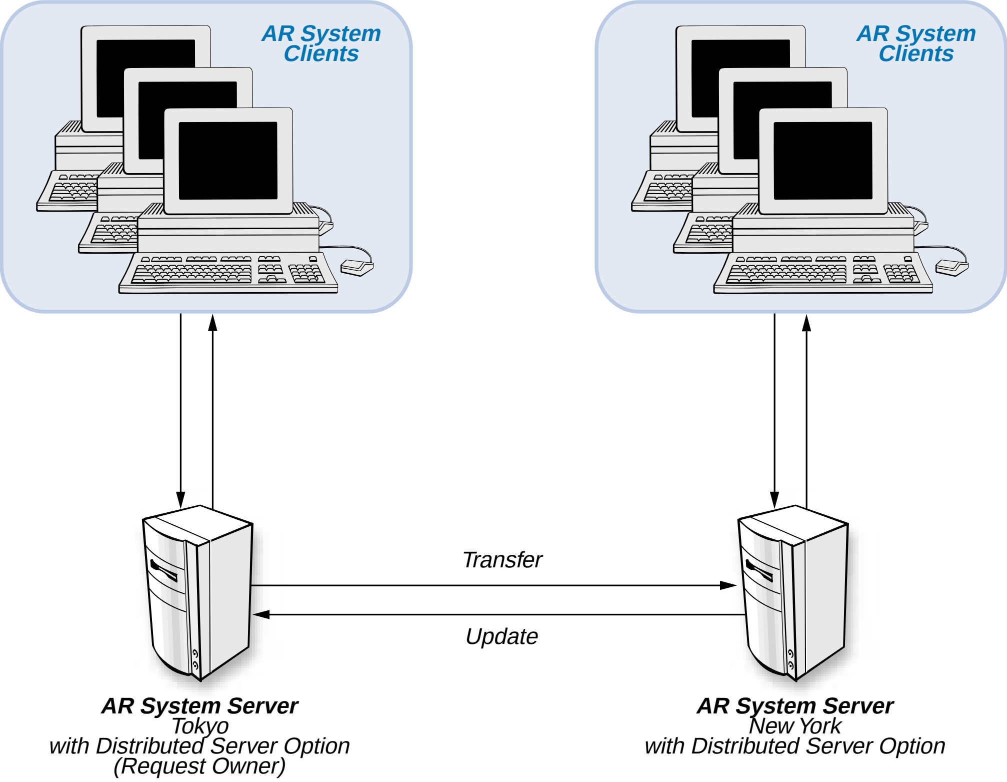
The heterogeneous environment enables you to mix and match clients and servers. For example, Remedy Administrator on a Windows client can administer forms on a UNIX server.

As another example, a web client can access forms on a Windows or UNIX server.

Distributing your AR System servers

You can build large-scale, distributed environments that behave as a single virtual system by using the Distributed Server Option. This option lets you share common information among servers and keep that information consistent. The following figure illustrates this concept. Much in the same way you define your business processes for an application, you define the processes for transferring information. Managers at each site must agree on what information to transfer from one application to another, what conditions drive transfers, and which sites control the ability to update a record. The administrator at each site then implements these decisions by using the Distributed Server Option.

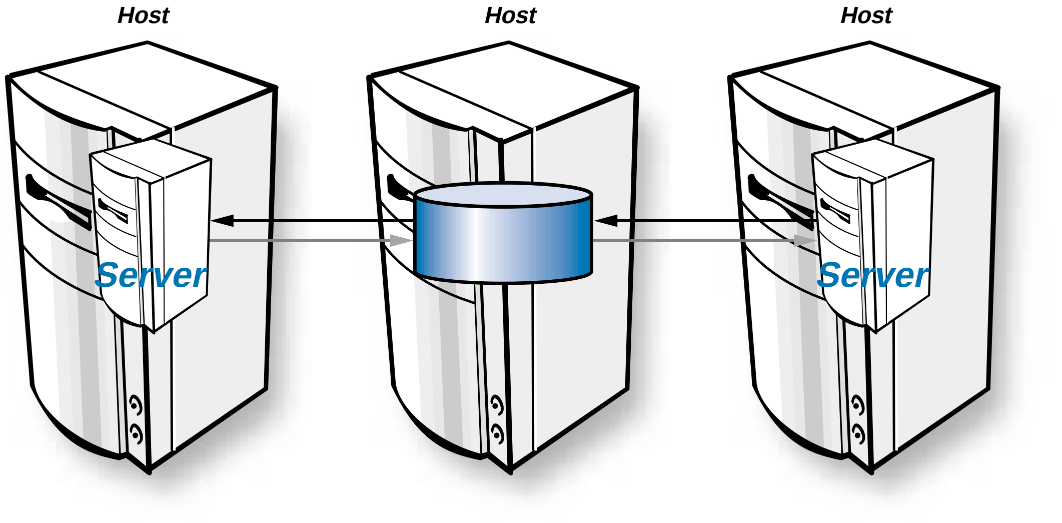
Figure 1-8: AR System in a distributed environment



Managing groups of servers

To provide scalability and increase reliability, a group of servers can be connected to the same database and managed as a unit. Servers in a group act as a "single" server to support the applications they are running. They can be configured to spread the load of shared services, and they can also provide backup to each other to ensure those services are always available.

Figure 1-9: Multiple servers using the same database



Chapter 2 Using forms

Forms are the foundation of the AR System. Menus and workflow components (discussed in subsequent chapters) are related to particular forms. Forms can be grouped into applications.

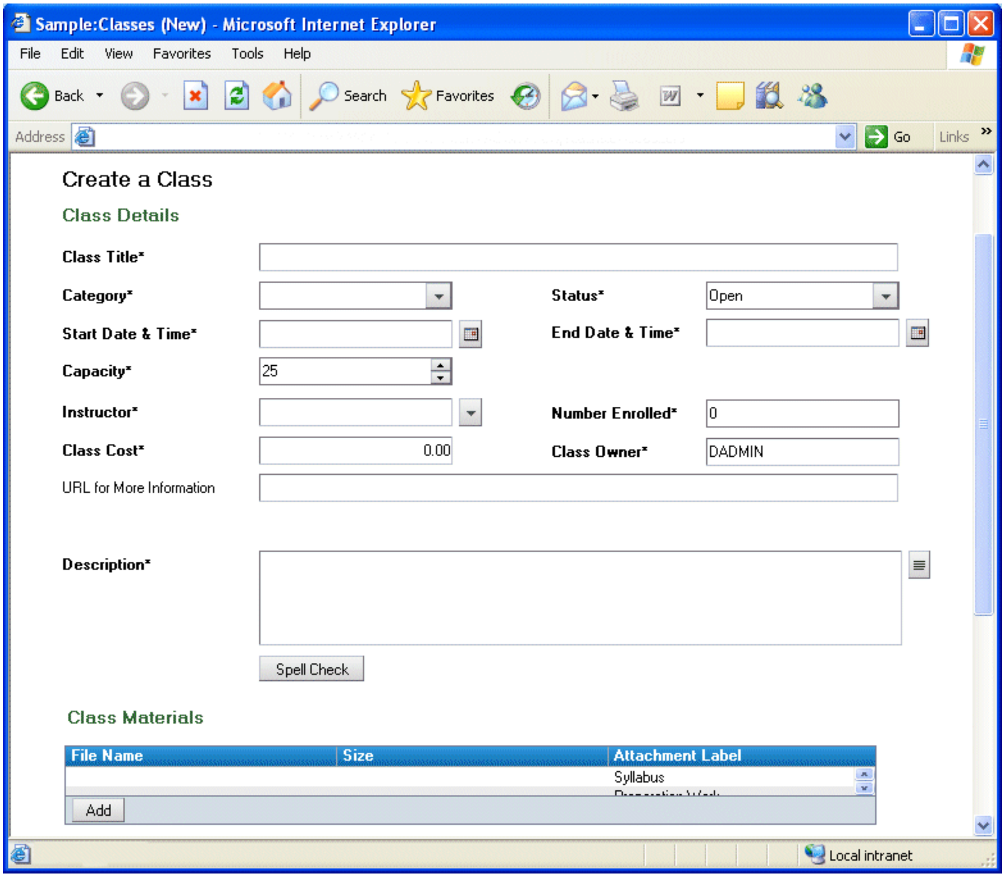
This chapter describes what forms are and how to use them in your business applications. In addition, the chapter describes the fields that comprise a form and how you can customize your forms to fit your business needs.

What is an AR System form?

A form captures and displays information. For example, a Help Desk form might capture information needed to fix a user's computer problem. A Purchase Requisition form captures the information needed to purchase an item.

Each form contains a set of **fields**. Each field captures a certain type of information, such as problem details or asset types and numbers, and has its own set of rules about who can view or modify the information it contains. Some fields have **menu items** attached that help users fill in the form. (Menus are discussed in Chapter 3, "Using menus as automation aids.")

Figure 2-1: Example form



Create a Class

Class Details

Class Title*			
Category*		Status*	Open
Start Date & Time*		End Date & Time*	
Capacity*	25		
Instructor*		Number Enrolled*	0
Class Cost*	0.00	Class Owner*	DADMIN
URL for More Information			
Description*			

Class Materials

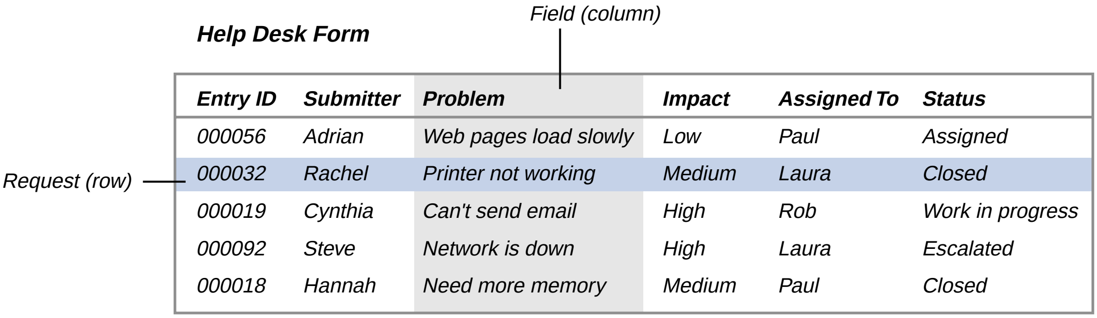
File Name	Size	Attachment Label
		Syllabus

Each form used in an application is like a template. Each time a user opens a form to perform a task, the template is presented to help the user complete the task. When the form is filled in and submitted to the AR System, it generates a **request**.

Users can create, modify, or search for requests if they have appropriate access permissions. (Permissions are discussed in Chapter 6, “Controlling access to AR System.”) Users can also create reports based on the requests that match their search criteria. They can use the AR System native reporting capability or they can use Crystal Reports, which is a report package that integrates with the AR System.

Forms are stored as tables in the database. Each data field on the form is a column in the table. (Data fields and other field types are discussed later in this chapter.) A request corresponds to a row (or record) in the table.

Figure 2-2: A form from the view of the database



Entry ID	Submitter	Problem	Impact	Assigned To	Status
000056	Adrian	Web pages load slowly	Low	Paul	Assigned
000032	Rachel	Printer not working	Medium	Laura	Closed
000019	Cynthia	Can't send email	High	Rob	Work in progress
000092	Steve	Network is down	High	Laura	Escalated
000018	Hannah	Need more memory	Medium	Paul	Closed

Types of forms

There are five types of forms you can create: regular, display-only, join, view, and vendor.

Regular forms, or data forms, are the forms discussed so far that contain information stored in database tables.

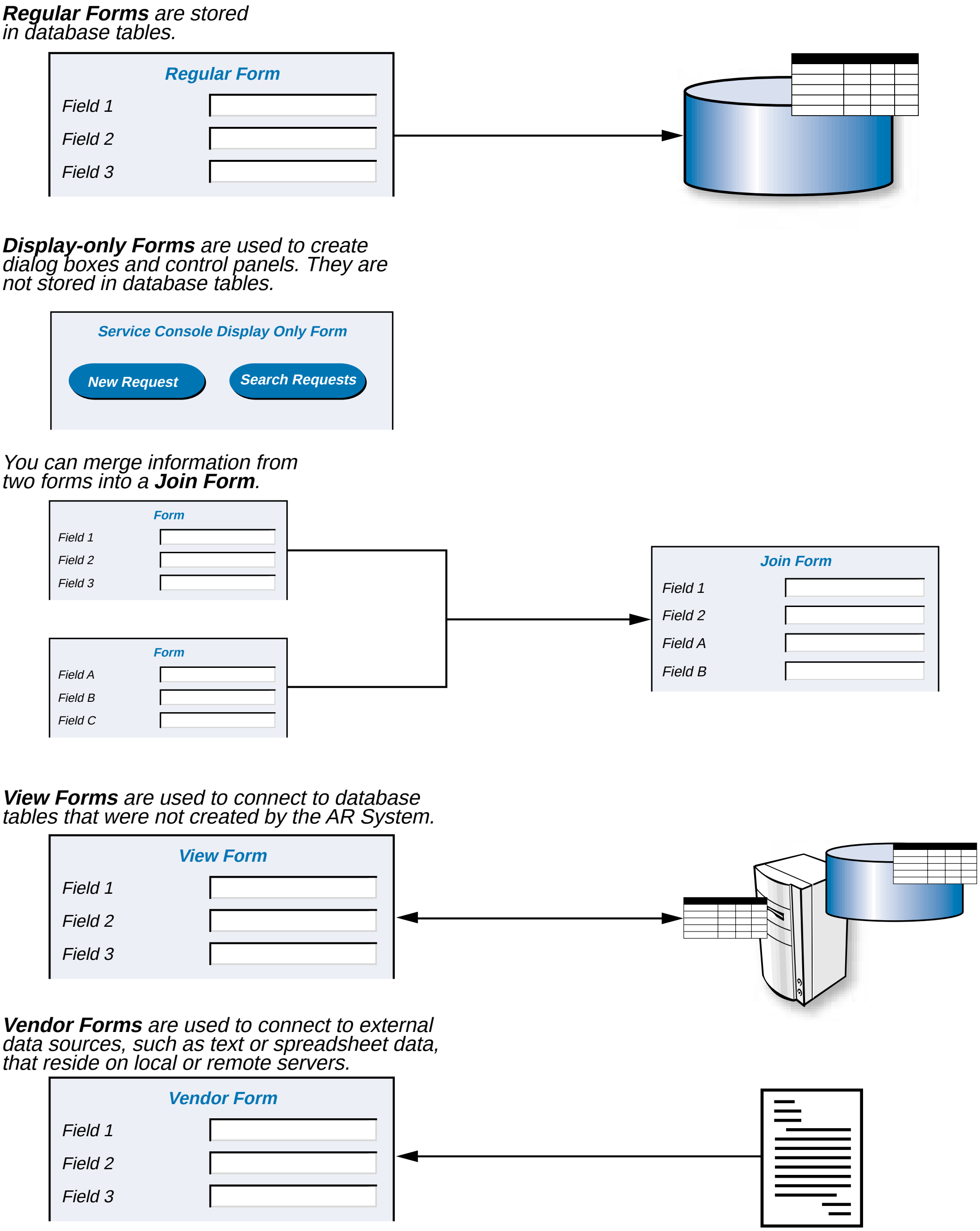
Display-only forms contain one or more display-only fields that enable users to accomplish specific tasks. They are most commonly used to create control panels, which serve as launch points from which users choose other tasks. (Control panels are discussed in more detail in the section “Acting as a control panel” on page 38.) Display-only forms can also be used to create dialog boxes, which prompt users as they fill out a form. (For more information about dialog boxes, see the section “Open a window” on page 83.) Display-only forms do not actually contain data, so no database tables are associated with these forms.

Join forms are composed of fields from one or more existing forms. They are useful when you have information in multiple forms that you want to display as a single form, for example, for reporting purposes. Like display-only forms, join forms do not actually contain data, so they have no database tables associated with them. The data is contained in the underlying forms that comprise the join. Join forms are discussed more fully in the section “Join forms” on page 40.

View forms enable users to connect to database tables that were not created by the AR System. View forms enable you to bring data from outside applications that are stored in a database directly into the AR System without replication or programming.

Vendor forms enable users to connect to external data sources, such as text or spreadsheet data, that reside on local or remote servers. Vendor forms enable you to bring data from other applications that are not stored in a database, using the AR System without replication; however, some programming is required to link to the other data source.

Figure 2-3: Type of AR System forms



__Regular Forms__ are stored in database tables.

__Display-only Forms__ are used to create dialog boxes and control panels. They are not stored in database tables.

You can merge information from two forms into a __Join Form__.

__View Forms__ are used to connect to database tables that were not created by the AR System.

__Vendor Forms__ are used to connect to external data sources, such as text or spreadsheet data, that reside on local or remote servers.

How forms are used

A single form can capture all of an application's functionality. For example, a small application that tracks product defects can use a single defect tracking form to capture and display all of the information needed.

Most applications, however, use several forms to capture, track, and organize information. One or more forms comprise the **main forms** (sometimes referred to as **primary** forms) that users interact with directly. Other forms used in the application are **supporting forms**, which supply information needed by the main forms.

Supporting forms

Supporting forms supply information needed by the application's main forms. Many supporting forms work "behind the scenes," so users might never see these forms. Supporting forms can be used for the following purposes:

- To hold reference information used by the main forms
- To hold workflow information used by the main forms
- To serve as a control panel for an application, initiating a variety of functions from a centralized point

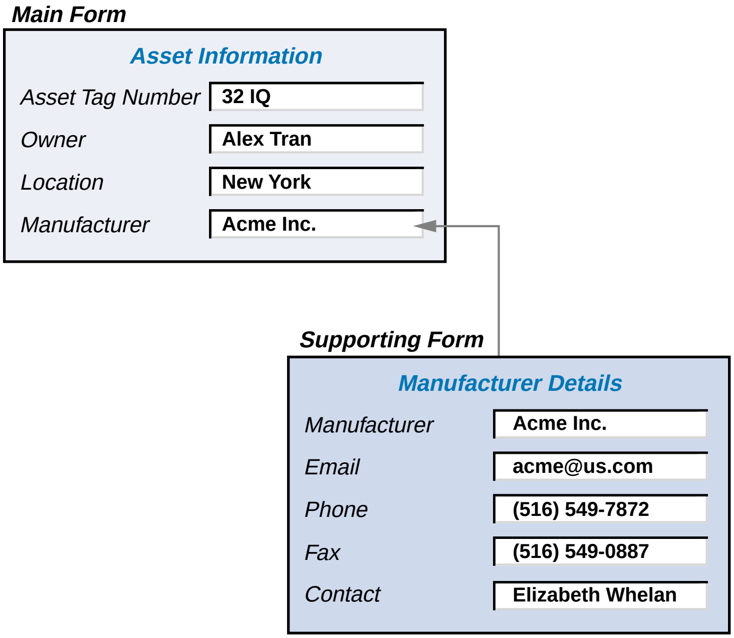
These three general uses of supporting forms are discussed in the following sections.

Holding reference information

Supporting forms can hold information that is used to assist the user fill out fields on the main form to decrease the overall create time. The workflow example used in the previous chapter used this method. When Steve entered information in the phone number field, an active link was triggered that matched that phone number to an entry in a supporting form and then pulled information from that entry to automatically fill in several fields on the main form. As another example, supporting forms can hold information used by Search menus that are located on the main form. (Search menus are discussed in "Search menus" on page 65.)

Supporting forms can also contain information that is used to provide additional details about the values provided on a main form. For example, as shown in the following figure, an Asset Information form might have a field that names the manufacturer of an asset. A supporting form might contain additional details about the manufacturer, such as the address, email address, phone number, fax number, and contact name.

Figure 2-4: Information taken from supporting form



Asset Information

Asset Tag Number: 32 IQ
Owner: Alex Tran
Location: New York
Manufacturer: Acme Inc.

Manufacturer Details

Manufacturer: Acme Inc.
Email: acme@us.com
Phone: (516) 549-7872
Fax: (516) 549-0887
Contact: Elizabeth Whelan

Holding workflow information

Supporting forms can hold data that defines workflow rules used in an application. This kind of workflow refers to the general business processes of your organization, *not* the specific AR System workflow components themselves (filters, active links, and escalations). For example, a supporting form holding workflow information might specify the escalation chain of command and how much time can pass before the next level is notified. As another example, a supporting form can contain the work schedules and skill sets of staff members to determine who will work on requests.

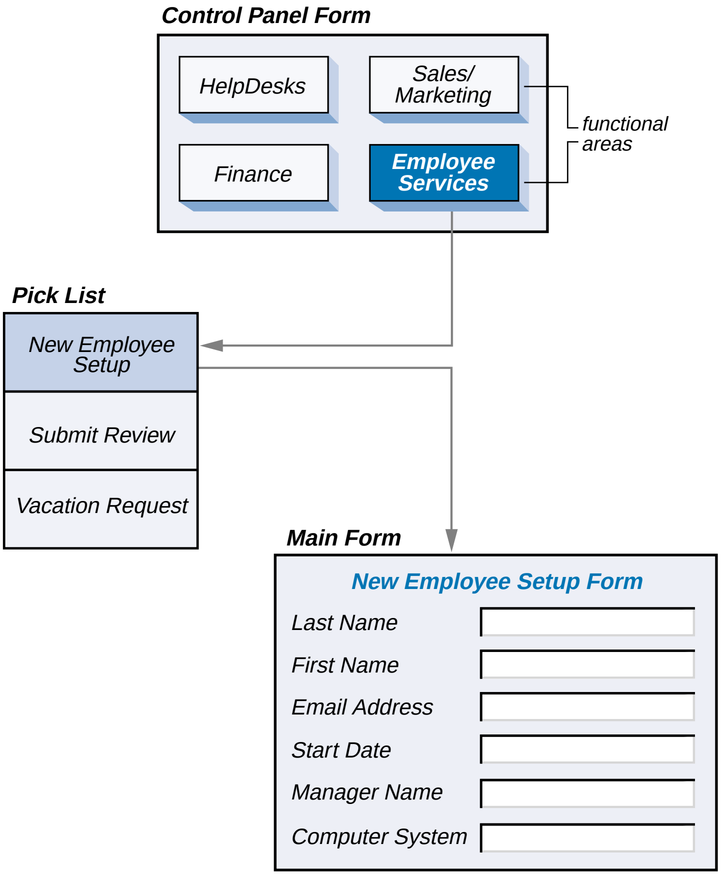
Acting as a control panel

A supporting form can be used as a launch point from which users initiate tasks. When a form is used in this way, it is usually referred to as a **control panel**. (Display-only forms are usually used to create control panels.) When users select choices from the control panel, they are presented with one or more main forms that they fill out.

The following figure shows a control panel used for a large corporate application. The application includes tasks encompassing a variety of functional areas, including HelpDesks, Finance, Employee Services, and Sales/Marketing. Users select the functional areas they are interested in and are then presented with a main form related to the task they need to do.

For example, a manager who needs to set up a new employee’s office would select Employee Services from the control panel and then New Employee Setup from the resulting pick list. The manager would then be presented with the main form related to setting up new employees, would fill out a request with all of the pertinent information about the employee, and then submit the request to the AR System.

Figure 2-5: Example of how a control panel works

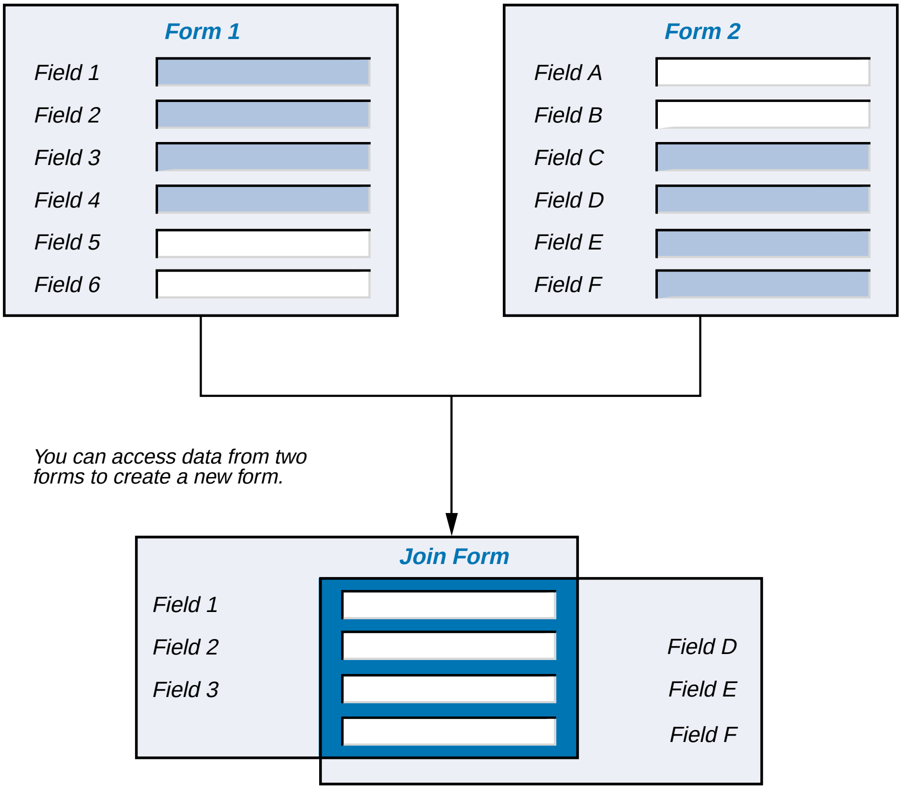


Join forms

You can combine information from two separate forms by creating a **join form**. A join form can be useful in the following situations:

- When you need to produce reports from data that exists in more than one form.
- When data is stored in multiple forms and you want to display the data in a single form.
- To eliminate the need to enter the same data into multiple forms.

Figure 2-6: Example of how a join form works



You can access data from two forms to create a new form.

The illustration in this figure shows how join forms help eliminate data redundancy. For example, creating a join form from your Support Call form and Customer Info form enables you to update a customer's phone number by modifying *one* request (in the join form) rather than two (in requests for the Support Call and Customer Info forms).

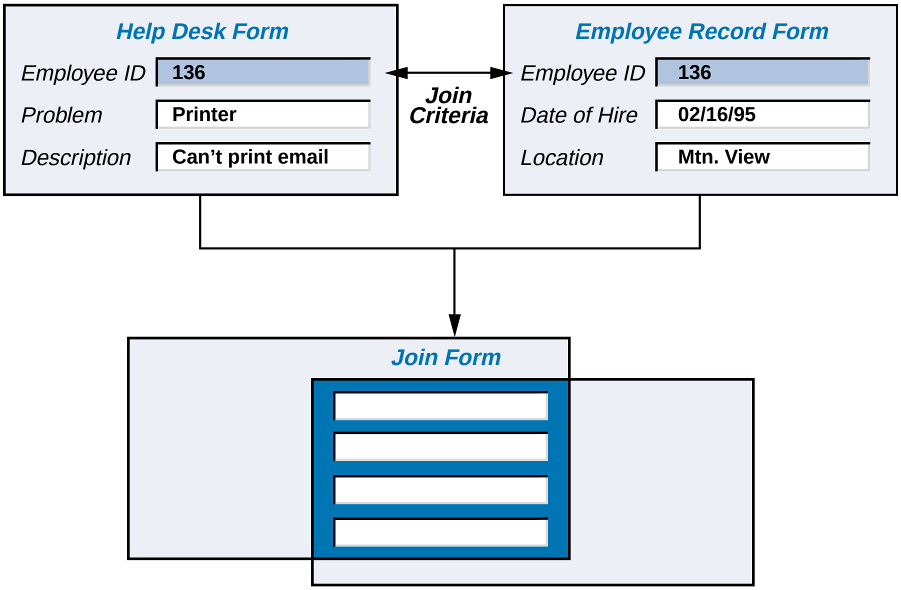
Join forms also provide data consistency. Modifications made to the phone number in the Customer Info form are automatically reflected in the join form and vice versa, decreasing the chance that the phone number is entered differently in different forms.

How join forms work

When you create a join form, you specify which forms you want to join, what the **join criteria** are, and which fields to include in the join form. The AR System then uses this information to search the database for field information from the forms that comprise your join form.

The join criteria define the link between the two underlying forms; it is a key value that the forms have in common. For example, as shown in the following figure, if a Help Desk form has an Employee ID field and an Employee Record form has an Employee ID field, you can join the two forms by linking the Employee ID field from the two forms. The Employee ID field becomes the join criteria.

Figure 2-7: Example of how a join form works



Note that a join form does not actually store data. It is a composite picture of the data you want to present from the two forms you specify. The data comprising the join form is retrieved from the underlying forms; it is then displayed, printed, or used in workflow as needed.

Example use of a join form

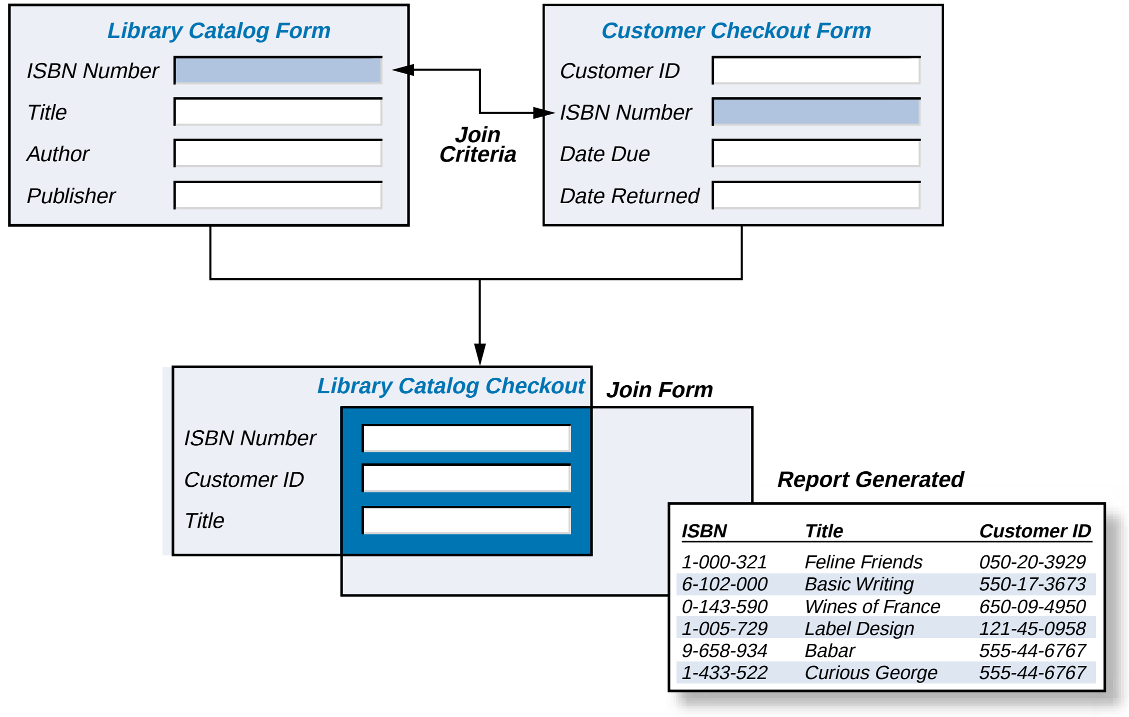
A join form example

A large university library needs to produce weekly reports of all catalog items (books, journals, and so forth) checked out by patrons. The library has an AR System application in place to manage and track various aspects of the library's inventory. Currently, the library uses the following forms:

- Library Catalog, which tracks information about all the items catalogued in the library. This form includes fields for ISBN, title of the item, author, and publisher.
- Customer Checkout, which tracks information about the items checked out of the library. This form includes fields for Customer ID, the ISBN of the item checked out to the customer, the due date of the item, and the date the item was returned.

The data needed to generate the library's report exists in these two separate forms, so the library administrator decides to create a join form containing the information needed, as shown in the following figure.

Figure 2-8: Example of how a join form works



ISBN	Title	Customer ID
1-000-321	Feline Friends	050-20-3929
6-102-000	Basic Writing	550-17-3673
0-143-590	Wines of France	650-09-4950
1-005-729	Label Design	121-45-0958
9-658-934	Babar	555-44-6767
1-433-522	Curious George	555-44-6767

Using multiple form views

A view is a visual representation of a form. You can create different views of the same form, enabling you to reuse the same forms for diverse groups while at the same time accommodating each group's unique requirements. In essence, views enable you to readily customize the interface of an AR System application so that each group sees the system as its own.

You can create as many views of a form as you need. For example, you can provide views customized according to the following criteria:

- Users' roles (requesters, managers, and so forth)
- Size of the screen (for example, laptop or desktop)
- Language or locale (for example, Brazilian Portuguese)

When creating a form view, you can:

- Change the layout of your form. For example, if your application is used by diverse groups, you might create a view for each group that places significant fields at the top of the form. An auto-layout feature allows you to precisely align and size fields in rows and columns according to a style sheet, which can be applied consistently across all forms in an application.
- Use different fields in different views. For example, you might choose to alter the data fields, page fields, and trim fields. You might also define active link control fields (buttons, menu items, and toolbar buttons) to be used in a view. You can even hide fields that are not relevant or that you do not want users of the view to see.
- Create a view that provides the best result in the target display environment, such as Windows, browsers, or both. Note that in most respects, the appearance and functionality of forms on the web is essentially the same as that in BMC Remedy User. This eliminates the need to create and maintain multiple views for different types of clients.
- Use terminology or language specific to the group using the view. For example, if you have users in Brazil and Mexico, you can create one set of views with all of the field labels translated into Portuguese and another set of views with all of the field labels translated into Spanish.

How a view is selected for the user

AR System tries to provide the user with the best possible view of a form. The choice of view is based on the user's application environment, language, and preference settings as follows:

1. The system selects the view category that has been requested by the user or by way of workflow. If no view category is requested or if the requested category does not exist, the default category will be used.
2. The system selects a view that is appropriate for the client that the user is running. If the client is on the web, the system will select a view according to the Prefer standard/windows views option in the BMC Remedy Mid Tier Configuration Tool.
3. The system selects a view that is appropriate for the user's locale. If there is not an exact match, a fallback mechanism finds the closest possible locale to the one requested. The resulting view is then displayed for use.

Guiding users through forms

You can lead a user through one or more forms by creating an **active link guide**. One way an active link guide can be used is much like a software "wizard" to help novice users complete tasks.

An active link guide consists of active links; each active link is associated with a specific form. You can set the order of forms you want the user to see. For details about the active links used in guides, see Chapter 5, "Workflow actions and firing conditions."

Bundling forms in applications

You can bundle your main forms and supporting forms into an application. When you do so, AR System automatically includes in that application all of the workflow and other components (such as menus) that are associated with these forms.

Applications developed in BMC Remedy Administrator are fully customizable and extensible. You can add your own fields, objects, and templates to any application, whether created by you, purchased from BMC, or acquired from a third party. Out of the box, AR System provides extensive authoring capabilities for applications built for Windows and web environments.

You can create two kinds of applications—*local* applications and *deployable* applications.

Local applications

Local applications are basic applications that allow you to group forms and workflow, and interact with them as an organized unit. Local applications use permissions based on groups that are defined on the local server. This tends to tie the application to the local server environment, since there can be conflicts between groups on different servers.

Deployable applications

Like local applications, deployable applications allow you to group together forms and workflow as organized units. However, deployable applications are more advanced and have additional features and options that make moving between server environments easier.

A deployable application uses permissions based on roles defined within the application. These roles are then mapped to groups on the local server to tie the application to that server. This makes deployable applications portable to a variety of target environments, since you can install the same application on different servers and simply map the application's roles to appropriate groups on each server. For more information about groups and roles, see "Role-based access in deployable applications" on page 114.

Other features are available only to deployable applications, including: gathering statistics about the application, and mapping roles to different groups depending on the development state of the application (such as Test or Production).

Localizing applications

Localization is the process of customizing an application for use in various languages, countries, and cultures. The AR System provides a complete **internationalized** environment for building, testing, and localizing your applications. A **locale** describes the language, country setting, and other characteristics that define the display of information and user interaction with the system. You can create an AR System application to be run in a particular locale, or you can make your application available in multiple locales simultaneously.

The development environment provides for full localization of interaction with the system, including:

- The language used for labels, messages, help text, reports, menus, and any other words that are part of a form.
- The separator symbol for decimal numbers and the grouping symbol for numbers over one thousand.
- The format for dates and times.
- The layout, colors, and images that are used.

Each localized version of an application can be stored as a view. Therefore, the same application can provide separate interfaces (separate views) for British English, Australian English, Mexican Spanish, and Peruvian Spanish. The data and workflow are the same for all users, even though the language and interaction for each user is tailored to their locale and preferences. Because all users of the application share the same data, you need to agree on the language for the data before the application is made available.

The localization features have been designed to be automatic for the user and easy to implement for the application builder. To localize an application for a given locale, an administrator only needs to create views for that locale and add corresponding messages to the message catalog. Utilities are available to assist with this work. For more information about localization, see the *Form and Application Objects* guide.

Accessing applications using entry points

You can create links called **entry points** that users click to start a task in an application, such as creating a new request. When users click on an entry point, such as `Create a New Help Ticket`, they are directed to the appropriate server and form, and the form opens in the mode specified—in this case, "new."

Entry points appear on a designated **home page** form, which opens by default when a user logs in to BMC Remedy User, or by entering a "home" URL in the web client. You can also list entry points on any form by including a special kind of field called an **application list field**. The entry points that the user sees are only those to which the user has access permission.

AR System fields

The fields that make up a form enable you to control how information is captured and displayed. For each field in a form, you can determine:

- If users can access the field and, if so, whether they can simply view the field or change its contents.
- The type of information the field can contain (such as characters, integers, or date/time data).
- The amount of information the field can contain (field length).
- If the field should be visible or hidden.
- If the field should be enabled or disabled.
- If the field is required, optional, or display-only (a temporary field for which no space is allocated in the database).
- Where the field appears on a form.
- How the field is displayed (for example, its label and physical appearance).
- How information is entered into the field (for example, automatically by an active link or by having users select choices from a list or menu).
- The field's default value.
- Whether fields are indexed for faster searches.

You can add as many fields as you need to your form (within the limits of your database) to capture and display the information needed for your application.

Indexing fields

To reduce the time it takes to search for information, you can index specific data fields in a form. An index on a form is much like an index in a book: it provides quick access to particular information. The fields to index are those that are searched on most frequently in user searches and in automated workflow searches. The Request ID field is indexed by default.

The AR System includes an indexing capability for fields containing small amounts of information, such as employee names, status, or problem category.

For more information about the indexing capabilities in the AR System, see the *Optimizing and Troubleshooting* guide.

Core fields

When you first create a regular form, it automatically contains a set of core fields that are predefined by the AR System as follows:

- Request ID: A unique tracking number assigned to each request by the AR System.
- Submitter: The login name of the user who submits a request.
- Create Date: The date and time a request is created.
- Assigned To: The person assigned to handle the request.
- Last Modified By: The user who last modified the request.
- Modified Date: The date and time the request was last modified.
- Status: The current status of the request.
- Short Description: A brief description of the request.
- Status History: When the request's status was last changed, and who made the change.

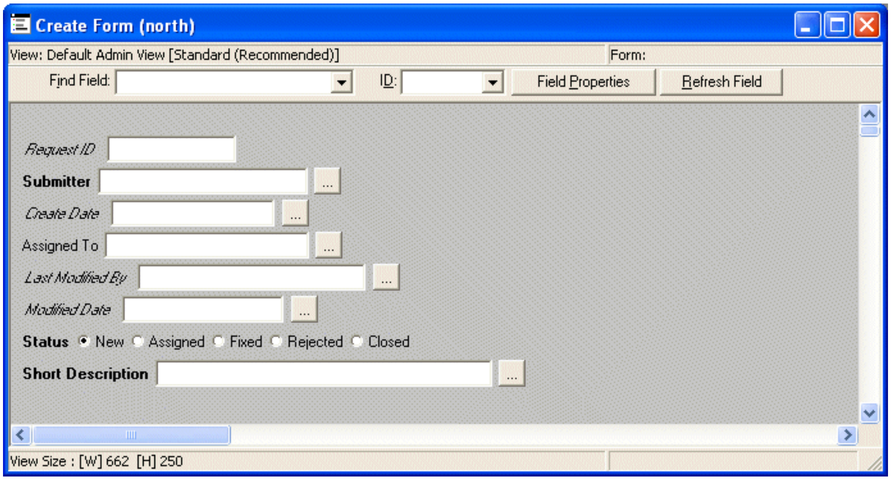
These core fields are built into the AR System to provide base capabilities that most application designers need. You can find a complete description of each of these core fields in the *Form and Application Objects* guide.

The AR System comes with templates for blank forms and forms with core fields. You cannot delete core fields from a form that was created with them, but you can hide them from a user's view and change their labels, location, and display appearance.

The following figure shows a newly created form with core fields.

- Field names shown in boldfaced text are required fields.
- Field names shown in italic text are fields that are automatically maintained and updated by the AR System.
- Field names labeled in plain text are optional for successful submission of the request.

Figure 2-9: New form with core fields



View: Default Admin View [Standard (Recommended)]

Status ● New ○ Assigned ○ Fixed ○ Rejected ○ Closed

View Size : [W] 662 [H] 250

How a field is identified

A field is identified in the AR System in three ways:

- Field ID—A unique numeric identifier that is assigned to each field. If the administrator does not assign the ID to a field when it is created, the AR System will do it automatically. Once field IDs are assigned, they cannot be changed. (The field ID corresponds to the column name in the database table.)
- Field name—A descriptive identifier for a field when it is stored in the database. Every field in a form must have a unique field name. The field name is usually referenced or displayed when defining workflow components. However, because the AR System stores the field ID within the definition of workflow components, you can change field names used in workflow without causing any conflicts.

- Field label—A name supplied by the administrator that describes the field's purpose, such as Vendor Name or Department Charged. The field label is for display purposes only; it is what users see on their forms. It is not required and if present, it does not need to be unique. This means that you can label a field differently in each layout (or view) of a form that you create. (Form views are discussed later in this chapter.)

What do all fields have in common?

All fields in the AR System share the following characteristics:

- They can be disabled (grayed out) or hidden entirely.
- They have a unique field ID and field name.
- They can be used in workflow.
- They can have context-sensitive help associated with them to help users learn more about the field.
- A field's display properties (including its location on the form and its appearance) can be changed.
- Permissions can be set to specify which users can access a field.
- The AR System automatically records the history of each field, including the field's owner, the user who last modified the field, and the date and time the field was last modified.

Field types

There are several types of fields that you can include in your forms:

- Data fields, which contain actual data values stored in database tables.
- Table fields (list view, tree view, results list, and alert list fields), which enable you to display data from other requests within the context of your current request.
- Attachment fields, which allow you to attach files to the request.
- View fields, which provide a browser window within the form. This browser can display any URL, HTML content, or file format that is compatible with a browser (including displaying contents of attachments, if desired).
- Data visualization fields, which provide a way to display flashboards or other graphics on AR System forms.
- Application list fields, which display a list of available entry points. The contents of this list are automatically generated from AR System.
- Horizontal and vertical navigation fields, which allow you to navigate to the correct screen in an application quickly and easily.
- Active link control fields (buttons, menu items, and toolbar buttons), which trigger active links.
- Page fields, which help you organize forms by using tabs.
- Trim fields, which enable you to add boxes, lines, and text to enhance the visual appearance of forms.

You manipulate the attributes of all AR System fields using workflow. This means, for example, that you can set the permissions for a group of trim fields or active link control fields so that they are not accessible to a certain group of users. Or you can add extra tabs in a page field that are visible to some users (such as managers or support staff), but not to others.

Data fields

Data fields contain actual data values that are stored in database tables. Supported types of data fields include:

- Character
- Date (a calendar pop-up dialog box allows you to easily enter date information in both the Windows and browser clients)
- Time (a control box allows you to easily enter time information in both the Windows and browser clients)
- Date/Time (a calendar pop-up dialog box allows you to easily enter date and time information in both the Windows and browser clients)
- Integer
- Real Number
- Decimal Number

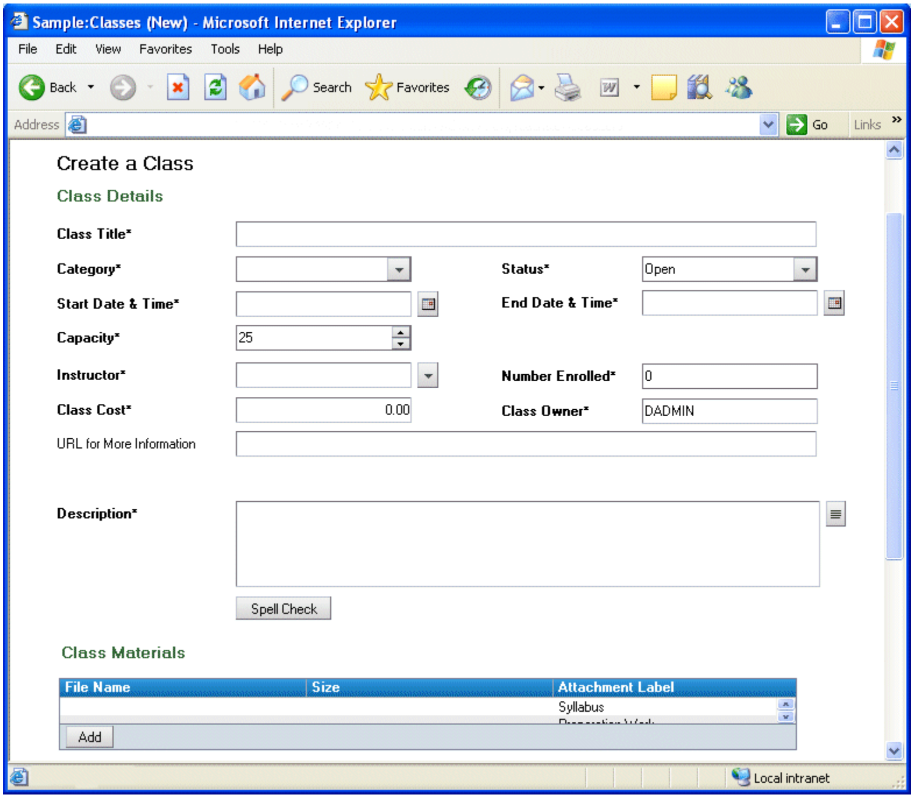
In addition, there are three special types of data fields:

- Diary—Show the history of a request over time as it changes. Each diary entry is automatically appended to the previous one and stamped with the time, date, and user's name.
- Selection—Provide mutually exclusive, fixed choices for filling in a field value. A selection field can appear as either a drop-down list or as radio buttons for multiple options, or as a check box when there is only one option. Users cannot enter choices that are not included in a list. For example, you might make the Impact field in a Help Desk form a selection list to prompt users to choose whether their requests are urgent, high, medium, or low.

  Selection list fields are useful in situations where you do not expect the choices to change over time. This is because it is easy to add new choices to the end of the list, but not to insert new ones within the list.
- Currency—Are multiple-column data fields that support currency data. A currency field displays a particular currency with the correct format, for example, 45.00 USD or 50.00 EUR.

The following figure illustrates the use of various types of data fields.

Figure 2-10: Form with several types of data fields


Sample:Classes (New) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Local intranet

Create a Class

Class Details

Class Title*			
Category*		Status*	Open
Start Date & Time*		End Date & Time*	
Capacity*	25		
Instructor*		Number Enrolled*	0
Class Cost*	0.00	Class Owner*	DADMIN
URL for More Information			
Description*			

Spell Check

Class Materials

File Name	Size	Attachment Label
		Syllabus
		Preparation Work

Add

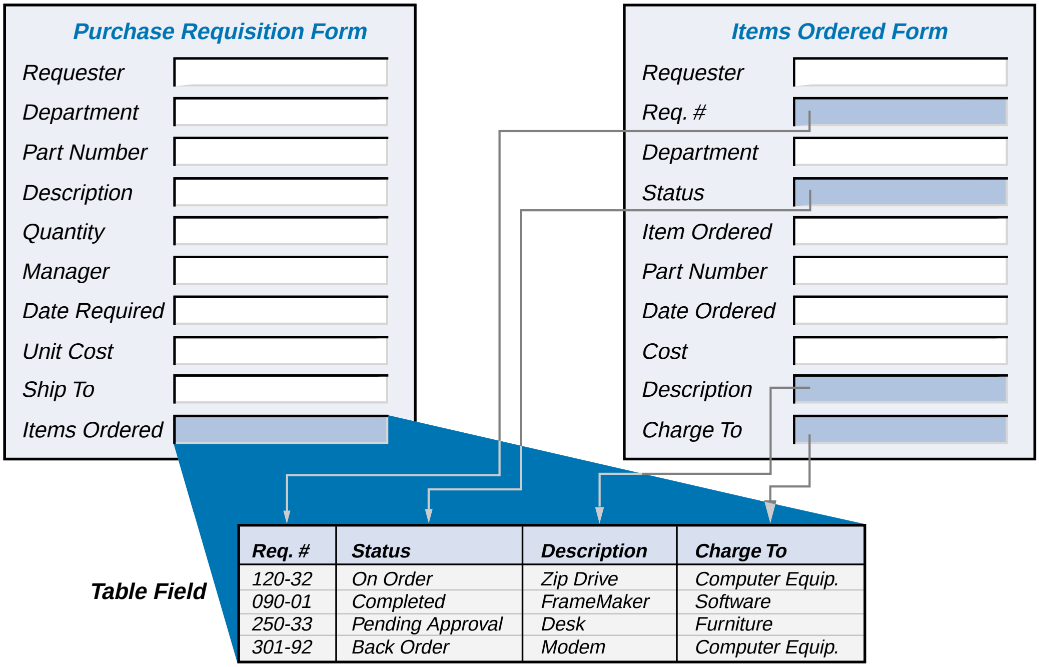
Table, results list, and alert list fields

Table (list view and tree view), results list, and alert list fields enable users to view specific fields and requests from a supporting form in a table format. These fields enable you to display data from other requests within the context of your current request. The data appears as a spreadsheet- or tree-like table. This is a useful way to view information that is related to the current request.

For example, as shown in the figure in this section, suppose you have a Purchase Requisition form, which serves as the main form for users, and an Items Ordered form, which is a supporting form containing details about the items that have been ordered. You can embed a list view table field within the Purchase Requisition form that enables users to view multiple requests from the Items Ordered form.

You decide which fields from the Items Ordered form to include in your table field. Each field that you select from this supporting form represents a column in the table, and each request appears as a row in the table. Users can open and modify any request represented in the table, if you give them permission to do so.

Figure 2-11: List view table field created from another form



Req. #	Status	Description	Charge To
120-32	On Order	Zip Drive	Computer Equip.
090-01	Completed	FrameMaker	Software
250-33	Pending Approval	Desk	Furniture
301-92	Back Order	Modem	Computer Equip.

This table field on the ***Purchase Requisition Form*** *displays selective information from the* ***Items Ordered Form****.*

You can choose to color rows in the table based on the value of a field in the row. For example, in the Status field in the illustration, Pending Approval can be displayed in red to alert a manager.

You can select table rows and create the following statistics on table columns:

- Sums
- Averages
- Maximum and minimum values
- Number of rows that contain data when a column is selected
- Number of table rows (with data and empty) when a table field is selected

You can create multiple table fields within a single form.

Attachment fields

Attachment fields enable users to attach files to their requests. Attachments can contain text, graphics, audio, video information, or other file content. The data in the attachment is stored with the request in the database. Users can open an attachment from within the request (the associated application opens automatically) or save the attachment wherever they choose.

An attachment pool contains and manages a set of one or more attachment fields. Workflow components can address attachment pools and attachment fields separately.

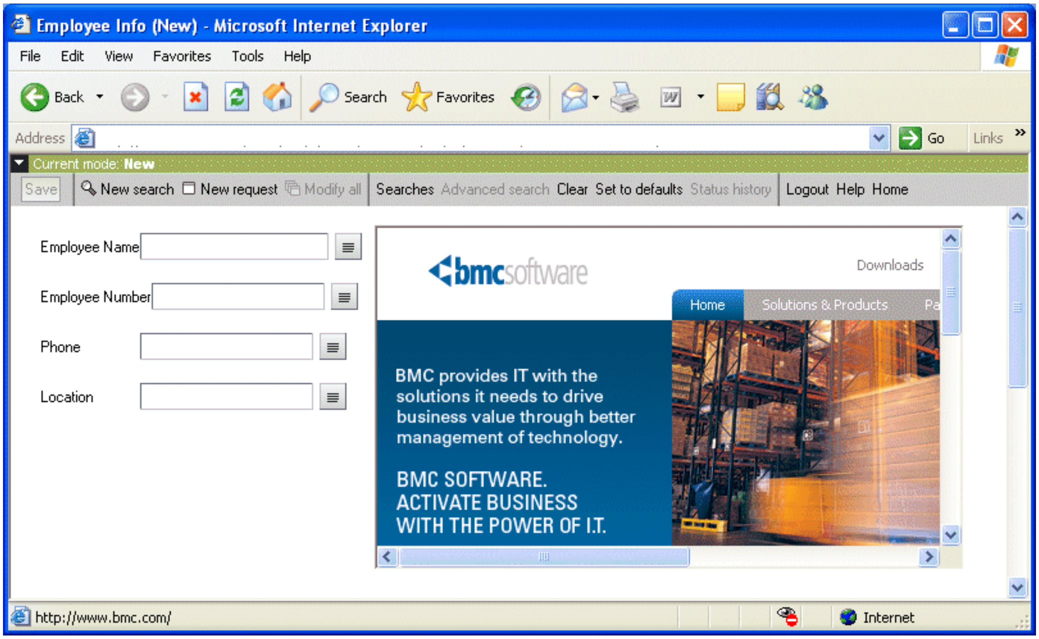
View fields

View fields provide the ability to embed a browser window within your form for use on Windows and browser clients. A view field is a fully functional browser window able to display the contents of URLs, HTML files, and other files that can be displayed in a browser. In addition, a view field can be directed to display an HTML text string or the contents of an attachment. Once something is displayed within the view field, the field behaves like a typical browser window allowing you to follow links to other pages and images.

A view field can be set to initially display some default content or URL. This content can be changed dynamically using workflow so that the contents displayed in the field can change automatically for the user based on other activity on the screen.

The following figure shows an example of using a view field as a browser within a form.

Figure 2-12: View field in a form viewed in a browser



Employee Info (New) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Current mode: New
Save New search New request Modify all Searches Advanced search Clear Set to defaults Status history Logout Help Home
Employee Name
Employee Number
Phone
Location
bmcsoftware
Downloads
Home Solutions & Products Pa
http://www.bmc.com/
Internet

BMC provides IT with the solutions it needs to drive business value through better management of technology.

BMC SOFTWARE.
ACTIVATE BUSINESS
WITH THE POWER OF I.T.

Data visualization fields

Data visualization fields provide a way to display flashboards, reports, and other graphics.

Flashboards are available for inclusion in data visualization fields on forms only if you have installed and licensed a Flashboards server. See “BMC Remedy Flashboards” on page 118 for more information.

Application list fields

Application list fields display a list of available entry points. The contents of this list are automatically generated from AR System. It is composed of the available applications, forms, and entry point guides on a given server.

Horizontal and vertical navigation fields

Horizontal and vertical navigation fields allow users to navigate to the correct screen quickly and easily. A horizontal navigation field might allow a user to move from application to application, and a vertical navigation field might give the user access to common functions and application entry points within an application.

Active link control fields

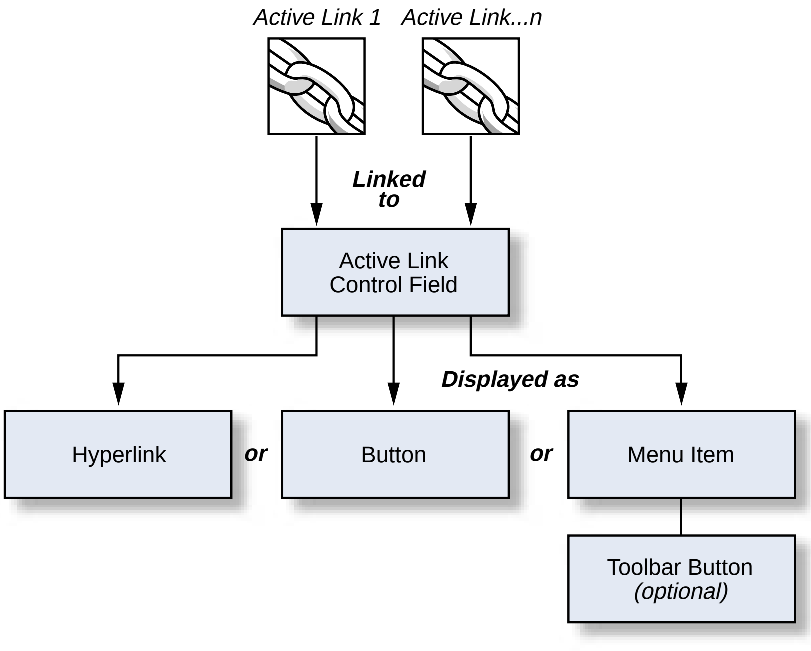
Active link control fields are buttons in the form of text and graphic buttons, hypertext links, menu items, and toolbar buttons that trigger active links.

- A button refers to a button field, which is a display-only field on a form.
- A hypertext link, or hyperlink, is a piece of text on a web page or form usually linked to another document or a different location in the same document.
- A menu item is a menu or menu item on the menu bar in BMC Remedy User.
- A toolbar button is a graphical shortcut for a menu item in BMC Remedy User.

Active link control fields in the form of buttons or hypertext links can appear anywhere on a form. Buttons are especially useful for triggering active links that are used frequently. Buttons can be marked with a label, an image, or both.

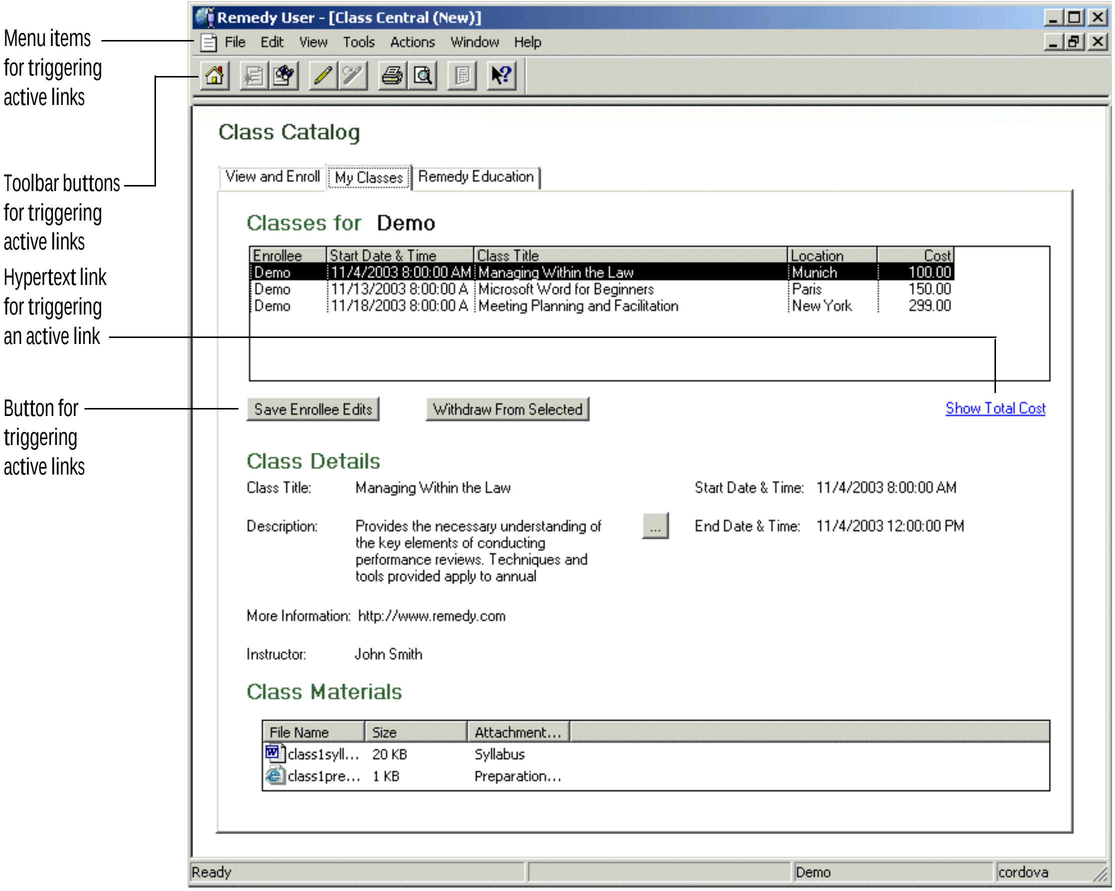
If you use a menu item to trigger an active link in BMC Remedy User, you can also choose to include a toolbar button as a shortcut to choosing the menu item. You might use a menu item when the active link is not frequently used. In addition, using a menu item frees up space on your form that you might need for other purposes.

As illustrated in the following figure, you can associate more than one active link with the same active link control field. For example, you can define an active link that executes only if the client is running on a PC. You could then associate another active link with the same active link control field, this time defining the active link so that it fires only if the application is running in a browser.



The following figure illustrates all of the possible formats for active link control fields.

Figure 2-13: Examples of active link control fields



Menu items for triggering active links

Toolbar buttons for triggering active links

Hypertext link for triggering an active link

Button for triggering active links

Class Catalog

Classes for Demo

Enrollee	Start Date & Time	Class Title	Location	Cost
Demo	11/4/2003 8:00:00 AM	Managing Within the Law	Munich	100.00
Demo	11/13/2003 8:00:00 A	Microsoft Word for Beginners	Paris	150.00
Demo	11/18/2003 8:00:00 A	Meeting Planning and Facilitation	New York	299.00

Class Details

Class Title: Managing Within the Law

Start Date & Time: 11/4/2003 8:00:00 AM

Description: Provides the necessary understanding of the key elements of conducting performance reviews. Techniques and tools provided apply to annual

End Date & Time: 11/4/2003 12:00:00 PM

More Information: http://www.remedy.com

Instructor: John Smith

Class Materials

File Name	Size	Attachment...
class1syll...	20 KB	Syllabus
class1pre...	1 KB	Preparation...

Page fields

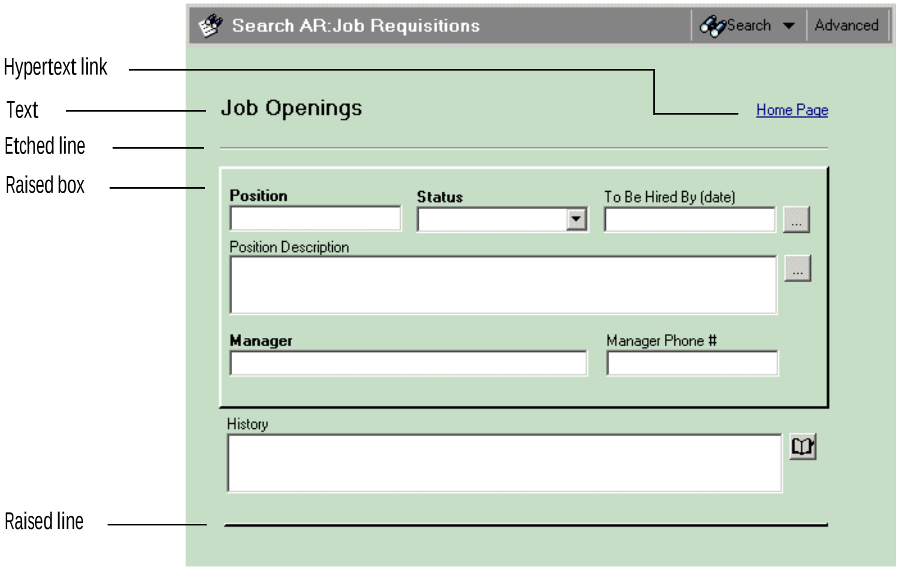
Page fields provide a way to organize fields on a form into multiple, tabbed pages. This enables users to fill out related information in manageable groups rather than having to scroll through a multitude of fields. For forms that will be viewed in a browser, you can hide the page tabs and add your own navigation method for accessing each page.

A **page holder field** contains the actual page fields for each form and defines the order of pages and which page is currently visible. You provide labels for page fields, which appear on the tabs. Workflow components can address page holders and page fields separately.

Trim fields

Trim fields, as shown in the following figure, are the boxes, lines, and text that enable you to better organize and annotate forms, and enhance their visual appearance. Boxes are useful for grouping related fields. You can choose from different border widths and styles including three-dimensional effects. Lines are useful for separating fields from one another, and, like boxes, provide options for border width and style. You can use text trim fields to add headings, instructions, or any other kind of text that you want to appear on your form. Text trim can also contain URL links displayed as hypertext.

Figure 2-14: Trim fields



Global fields

Global fields are a class of display-only fields that allow you to share data across forms or across records. You can define a number of different types of fields (such as character, diary, date/time, date, time, integer, decimal, real, and selection fields) as global fields by giving them field IDs within specific ranges.

Regular global fields are used to share data across *forms*. For example, you might include a global field for the user's name and employee ID on several forms, because this information is the same for each form in the application. When you enter the user's name and employee ID in the global field on one form, the same information appears in the matching global fields (global fields with the same field ID) on other forms.

Window-scoped global fields are used to share data across *records* within a single window. For example, on one form, you might include a window-scoped global field that contains navigation links, because you want those navigation links to appear on the form even if the rest of the fields are cleared or populated with different records. Unlike a regular global field, data is not shared across windows.

Chapter

3 Using menus as automation aids

You can attach a menu to any character field on a form to help users fill in the field. Menus can provide suggestions for entering data into a field, or they can be mandated as the only possible choices. Menus can be statically defined, dynamically built by searching AR System databases and external databases, or read from text files written by other applications.

Menus are separate objects stored independent of a form. This means that you can create a single menu and then reuse it for multiple forms. You can also use the same menu for as many fields as you want.

This chapter discusses the different types of menus you can create.

Types of menus

The AR System defines five kinds of menus:

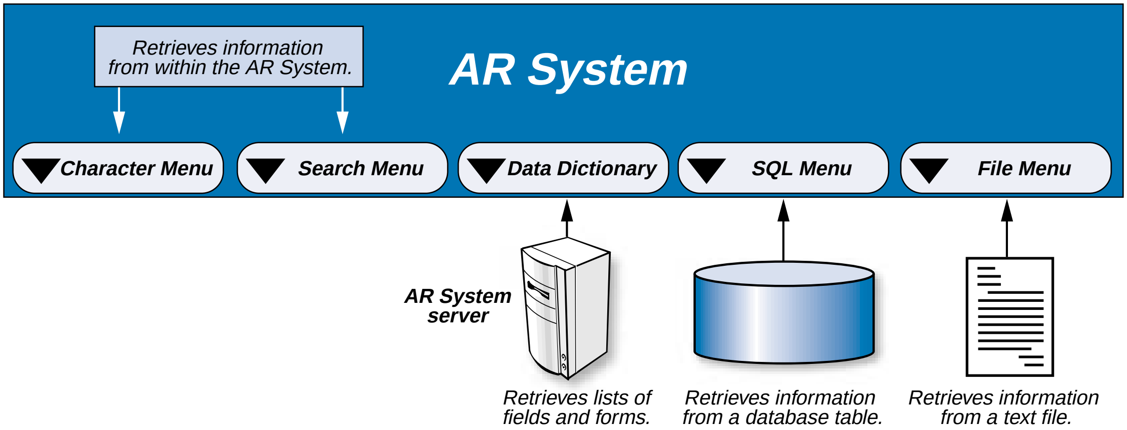
- Character menus
- Search menus
- Data dictionary menus
- SQL menus
- File menus

Although menus appear to be similar to selection list fields (described in Chapter 2, “Using forms”), there are distinct differences between the two:

- Unlike selection list fields, where the order of the items stored in the database makes it harder to make changes, menus can be readily altered without changing the meaning of data already stored in the database.
- Menus enable you to provide multiple layers of choices in the form of submenus, unlike selection list fields, which offer only one level of choice.
- The administrator can set up menus as merely recommended—rather than required—choices, allowing users to fill in the field as they’d like.

The different types of menus are discussed in the following sections.

Figure 3-1: Menu used in AR System


Retrieves information from within the AR System.
AR System
Character Menu
Search Menu
Data Dictionary
SQL Menu
File Menu
AR System server
Retrieves lists of fields and forms.
Retrieves information from a database table.
Retrieves information from a text file.

Character menus

Character menus are stored and maintained as a list of items within the AR System. They are useful for fields that have a predefined series of choices that do not change frequently. For example, a Department field is a good candidate for a character menu. Using character menus, you can provide multiple layers of choices in the form of submenus.

File menus

The items in a file menu are created, stored, and maintained as a text file outside of the AR System. The text files can be stored on either an AR System server or a Windows client. File menus are convenient when you do not want (or need) to store the data within the AR System. For example, you might use a file menu for the following reasons:

- You need to integrate with some other application that creates the data contained in the file.
- Your data file currently exists outside of the AR System and is very large (containing, perhaps, hundreds of items). In this case, it might be more efficient to simply reference the file using a file menu.
- You want to create a private file stored locally on a user's computer.

When a File menu is selected, the AR System reads the file you specify (by referencing the path name), and the items in the file are displayed as choices in the menu. When you need to change the menu, you simply update the file.

Like Character menus, File menus also provide users with multiple levels of choices in the form of submenus.

Search menus

Search menus retrieve information from forms stored in AR System databases. The information retrieved is then used to *dynamically* build a menu of choices in the current form. Search menus are often used in situations where the choices listed in the menu depend upon values entered in fields on the current screen.

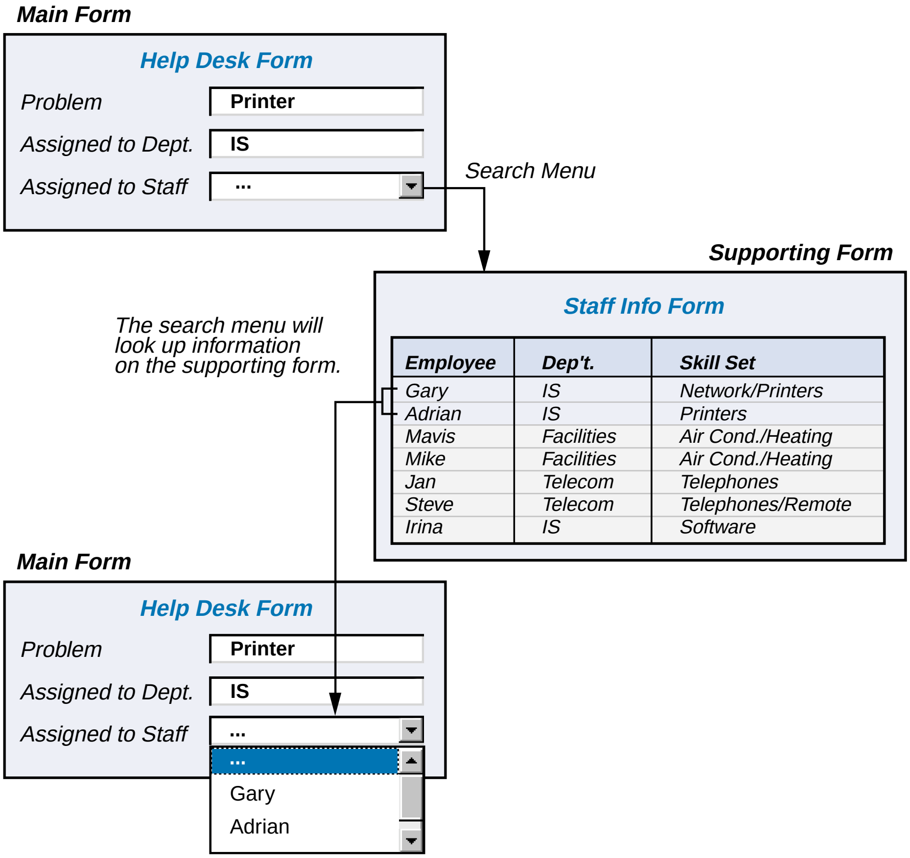
For example, suppose your company runs a Help Desk and you need to dispatch requests to different departments. Within those departments, a worker is assigned to actually handle each request.

When a user creates a Help Desk request, you want to provide the dispatchers with a menu listing the staff members they can assign the requests to, based on the staff members’ skill sets.

Currently, your company has a form called HelpDesk that captures users’ requests, and a supporting form called Staff Info that includes information about each staff member, including the staffer’s name, department, and skill set.

As shown in the following figure, to dynamically build a menu listing the appropriate staff members to assign requests to, you can attach a search menu to the Assigned To Staff field on the HelpDesk form. This search menu will retrieve information from the supporting form—Staff Info—to determine which staff members belonging to the Information Systems (IS) department are able to fix a printer problem.

Figure 3-2: Dynamic menu



Main Form

Help Desk Form

Problem: Printer
Assigned to Dept.: IS
Assigned to Staff: ...

Search Menu

Supporting Form

Staff Info Form

Employee	Dep't.	Skill Set
Gary	IS	Network/Printers
Adrian	IS	Printers
Mavis	Facilities	Air Cond./Heating
Mike	Facilities	Air Cond./Heating
Jan	Telecom	Telephones
Steve	Telecom	Telephones/Remote
Irina	IS	Software

The search menu will look up information on the supporting form.

Main Form

Help Desk Form

Problem: Printer
Assigned to Dept.: IS
Assigned to Staff: ...

In this example, the menu items listed in the Search menu (the staff members listed) depend on the values in two fields: Problem and Assigned To Dept. The search results in a list of two staff members who can handle the printer problem. The items listed in the Search menu will change as the values in these fields change. If, for example, the problem had been related to the air conditioning in the building, the Search menu would have resulted in a different set of employees from Facilities who could have been assigned to handle the request.

SQL menus

Like Search menus, SQL menus are used to retrieve information from database tables. (SQL is the language used to "talk" to relational databases.) However, SQL menus can be used to pull data from a table that is not part of the AR System.

When you access an SQL menu, the AR System uses an SQL command to query a table, extracts the data, and then generates the menu from the resulting data. You specify which column of the returned data is used as the menu label and which column to use as the menu value.

Data dictionary menus

Data dictionary menus are used to retrieve lists of fields and forms from an AR System server. This type of menu is useful for creating special configuration interfaces; it is generally not used to help users perform their work.

Chapter

4 Introduction to workflow

Forms, with the help of menus, capture the crucial data needed to run your business. Doing something with that data is the function of workflow. You use three workflow components—active links, filters, and escalations—to enforce business rules in a variety of ways, including notifying people of events, escalating problems to the next level, automatically routing information, and checking that key data is correctly entered.

This chapter explains what the workflow components have in common as well as what differentiates them.

Workflow: in general and in the AR System

Workflow can be defined as the set of processes that your company uses to run itself, for example, tracking defects or administering employee benefits. In that sense, workflow exists whether or not a single computer is present. What workflow does in the AR System is automate these processes through the use of active links, filters, and escalations. So, for example, if your organization decides that purchase orders for amounts above a certain level need director approval, you can design workflow that allows only correctly approved purchase orders to be automatically forwarded to the purchasing department.

Active links, filters, and escalations consist of descriptions (also called definitions or rules) that specify what actions to take under what circumstances. The actions can be triggered by the state of data (for example, the value in a field) or by the amount of time data has been stored. The triggers of the actions are called firing conditions.

Some of the actions that the workflow components can take to automate processes and ease data entry are:

- Changing the values in fields to values you specify, for example, to override a value a user has entered.
- Manipulating a form, for example, enabling or disabling fields, or changing menus associated with fields.
- Error checking.
- Enabling cross-application integration using OLE or DDE.
- Opening new windows for data entry or display.
- Communicating with users by means of onscreen messages or notifications sent by email, BMC Remedy Alert, or other methods.
- Running a guide as a subroutine—that is, a predefined sequence of commands.

A complete list of actions appears in “Actions: what workflow can do” on page 77.

How workflow components differ

The three workflow components can be grouped differently depending on what aspect you are comparing. Active links are client-centric—they respond directly to user input. You create an active link when you want someone to do something. For example, an active link might warn a user that the value just entered is flawed in some way.

In contrast, filters and escalations are server-centric—they respond to business rules. Filters are the true gatekeepers for the AR System. That is, they ensure that the actions taken by the system follow your business rules. For example, you can decide that only correctly approved purchase orders can be entered; all others will be returned to the requester. Or you can decide that the manager of Information Systems will be notified whenever a critical server malfunctions.

In reality, an escalation is really just a time-based filter. Use an escalation to make sure your business runs as you want it to. For example, create an escalation to warn another group of users that in one hour their manager will be notified that a problem has been unsolved for too long.

On the other hand, active links and filters are similar to one another in how they are triggered, whereas escalations and filters have the location where they are performed (the server) in common. These factors influence which components you use to accomplish your goals. The following table summarizes how and why you would use these different workflow objects.

Component	Triggered by: events or passage of time	Where it is performed: client or server
Active link	Events	Client
Filter	Events	Server
Escalation	Time	Server

Note: You can create a workflow component for one form, or you can share a component with several forms.

Firing conditions: events compared to time

Filters and active links are triggered by events such as a change in the state of some data or by some specific action. For example, a filter can be designed to notify a support manager whenever a new request is submitted to the server with a priority of High or Critical. The submission of the new request is the event. Other events that can trigger filters are the updating, deleting, or retrieving of requests. Active links can be triggered by additional mechanisms such as opening or closing a window, displaying a request, pressing a button on a form, pressing Return in a field, or making a menu selection.

Escalations are time-based, meaning that they are triggered by the passage of time. The trigger (or execution condition) can be either absolute time, such as “every day at 2:00 p.m.” or a time interval, such as “once every hour.”

You can further refine the firing condition for a workflow component by adding a **qualification**. In the previous example, the requirement that the request have a certain priority is the qualification for the filter. Qualifications are explained in more detail in Chapter 5, “Workflow actions and firing conditions.”

Client-based compared to server-based

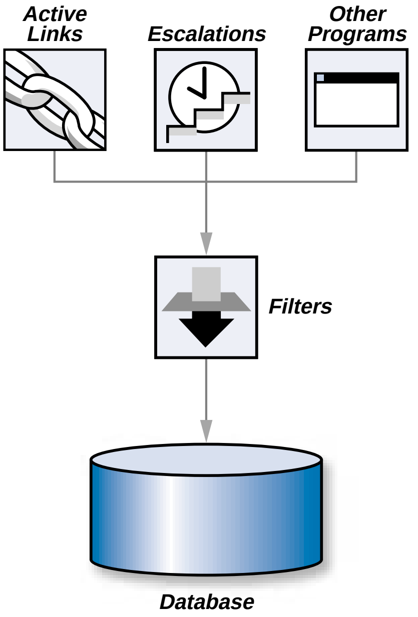
Another essential difference between filters and escalations on one hand and active links on the other is where they are executed. The definition (or logic) of an active link resides on the client. In fact, unless an active link queries the AR System server for information, it can complete its operation even if the server is down. For example, messages can still appear, the cursor can be moved to another field, and similar processing can continue. (If an active link does query the server and the server is down, then an error is generated.) The active link is like a researcher who is writing a book; all the writing (processing) can be done in his home if he has all the necessary books. Even if he travels to a library (server) to get more information, and then adds that information to the book he is writing at home, the process still is performed by the researcher.

The definitions of filters and escalations reside on the server; the tasks take place after the transaction has been transferred to the server for processing. For filters, this is analogous to the researcher giving the book to the publisher (server) for final editing before it is published. Note that the processing can return to the client where more active link actions can take place.

Active links and filters have some actions in common, for example, setting values in fields. Sometimes, part of the reason for choosing which of the two components to use is the location of the component's logic. For example, the reason that filters are used for integrity checks is that they are located on the server and the system *guarantees* that they will operate, regardless of how the server gets accessed— whether by an API call, from some other client, using AR Import, processing modify requests in bulk, or by any other means. In contrast, if active links were solely responsible for integrity checks, and the user did not perform some action (such as pressing a button), the checks would not take place. This is one reason that active links are often used as convenient, but not required, accelerators, such as automatically retrieving values for fields in response to a user action.

The importance of filters as guardians of the system is illustrated in the following figure. Note that even escalations must pass through filters if they change any data that the filters monitor.

Figure 4-1: Filters in AR System



Collecting workflow into procedure guides

Active links and filters can each be bundled into collections to be run as procedures. These collections are called active link guides and filter guides, respectively.

The workflow in guides can be organized in a defined processing sequence. Using guides allows you to name a set of workflow operations that accomplish a specific task. These named guides can be called at any point during workflow processing.

In addition to being used as procedures, active link guides can interact with users by requesting information and then waiting for the input. This feature allows you to create tasks that guide users through application processes in a way similar to software “wizards.”

Chapter

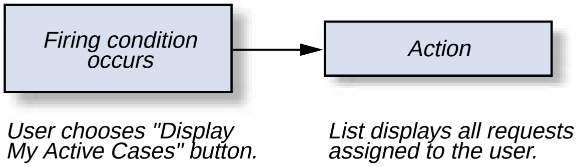
5 Workflow actions and firing conditions

The previous chapter introduced the workflow components. This chapter provides more detail about what the components can do, beginning with the actions that active links, filters, and escalations perform, and followed by the firing conditions that trigger the actions.

Mechanics of workflow: actions and firing conditions

The basic question about workflow is: “What can I do and when can I do it?” The actions that workflow can take are the *what* and the firing conditions are the *when*. For example, choosing a button called Display My Active Cases could display a list of all requests assigned to the user.

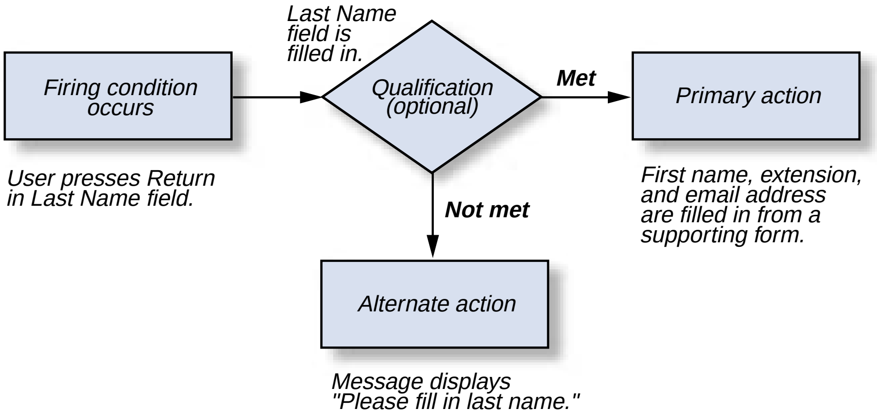
Figure 5-1: Example of workflow


Firing condition occurs
Action
User chooses "Display My Active Cases" button.
List displays all requests assigned to the user.

You can further refine the firing conditions by specifying other criteria that must be met before the action is taken; these criteria are qualifications, which are discussed in the section “Qualifications” on page 92. Carefully designed qualifications make the workflow components even more efficient and powerful.

You have the ability to control whether an action is the primary or alternate action. If an operation meets the qualification you specify, the primary (“if”) action is performed; if not, the alternate (“else”) action is performed, as illustrated with an active link in the following figure.

Figure 5-2: Example of workflow


Last Name field is filled in.
Firing condition occurs
Qualification (optional)
Met
Primary action
User presses Return in Last Name field.
Not met
First name, extension, and email address are filled in from a supporting form.
Alternate action
Message displays "Please fill in last name."

Note: When escalations are performed, the alternate action is taken only if no requests meet the qualification. Therefore, alternate actions cannot change a request, but can do other types of tasks. However, because active links and filters *always* operate in the context of a request, they can change the current request with either primary or alternate actions.

Actions: what workflow can do

The actions that active links, filters, and escalations can take are listed in the following table.

Action	Active Link	Filter	Escalation
Display a message on the screen.	✓	✓	
Log information to a file (for an audit trail).		✓	✓
Notify users of events.		✓	✓
Set fields on a form to different values.	✓	✓	✓
Set fields on a form using (“consuming”) a web service.		✓	
Push values to fields on other forms.	✓	✓	✓
Run a separate process (program).	✓	✓	✓
Run an SQL command.	✓	✓	✓
Perform an alternate action in the processing sequence (“go to,” “go to label”).	✓	✓	
Start (call) the specified guide.	✓	✓	
Suspend active link guide processing until the user responds (“wait”).	✓		
Exit the guide.	✓	✓	
Open a window.	✓		
Close the current window.	✓		
Commit changes to the current window.	✓		
Change the appearance of a field, change the menu associated with a field, or refresh a table field.	✓		

Action	Active Link	Filter	Escalation
Use DDE to connect to other applications.	✓		
Use OLE to connect to other applications.	✓		

Each of the actions is described in the following sections. The icons indicate which actions apply to which workflow components. If there are differences in how the action functions for different components, those differences are explained.

Display a message on the screen

An active link or filter can display a message to the user that prompts with advice, gives reactive information, warns of a particular situation, or presents an error message and stops the processing of current workflow.

Because active links execute on the client, you can use them to display messages immediately to the user. For example, if a user fills in a particular field, an active link could warn that a related field must be filled in first. This way, the user is directed to provide consistent data before the transaction is sent to the server. A filter, which executes on the server, is useful for checking an entire transaction and sending error messages or informational messages. For example, a filter could display a message to a user that the support staff has been notified about a problem.

Log information to a file

You can designate a file that contains an audit trail of transactions meeting the criteria you specify. Each log entry includes the date and time, the name of the user whose action triggered the filter, the name of the form, the entry's ID number, and the fields included in the transaction (that is, those fields that have changed). For example, you could create a filter that logged all vacation requests during a given time span. Or you could create an escalation that made a log entry every time a service level agreement was not met.

Escalations do not support the Log to File action as an alternate action because if no requests are retrieved, there are no requests to log.

Notify users of events

You can notify users of certain events. For example, a filter could notify support staff that they have been assigned a new request. Or an escalation could notify the service department that an asset warranty has expired. The methods for notifying users are:

- Electronic mail
- BMC Remedy Alert
- A delivery mechanism created by you, for example a Microsoft Windows service that pages users when they receive a notification

A useful feature of this action is the ability to notify groups of users about certain events. The creation of groups in the AR System is described in Chapter 6, “Controlling access to AR System.”

Set fields on a form to different values

Set Fields is the ability to fill in fields on the current form with values that you specify. For example, a filter can automatically set the Status field to Assigned every time a name is entered into the Assigned To field.

The value that you put in a field can be static (always the same), a keyword value (based on one of the many predefined variables BMC offers), or a value retrieved from another source of data.

Keywords

A keyword is a variable whose value is defined by the AR System. Keywords are uppercase and enclosed in dollar signs. For example, `$USER$` represents the name of the user that is currently logged in; `$TIMESTAMP$` represents the current date and time; and `$OPERATION$` represents the operation currently in progress.

Keywords can be used almost anywhere a qualification can be defined or a value specified. Some examples are:

- Defining qualifications for search menus and for workflow. For example, you can use the keyword `$OS$` to check that the operating system can run the process you specify in workflow.
- Specifying a value in the Set Fields action for active links, filters, and escalations.

- Defining searches and macros.

A complete list of keywords can be found in the *Workflow Objects* guide.

If the value for a field is retrieved from another data source, that source can be any of those from the following list.

- A field in the same form or another form on the same server (or different server for active links).

  If the value is assigned from a request other than the current one, you are allowed to control what the server does if there are no requests that match the qualification or if there are several requests that match the qualification. If no requests match, you can specify that the field either be set to `NULL` or that an error be generated and the workflow component stop processing. If more than one request matches the qualification, you can specify that either the field be set to `NULL`, that the first matching request be used, or that an error be generated and processing cease. Active links have the additional option of displaying a selection list of the matching requests for the user to choose from.

- The result of a mathematical calculation.
- The result of a function.
- The result of an independent system process.
- The result of a direct SQL command.

  You can use an SQL command to get values from tables other than AR System tables. Filters and escalations can run the SQL command on the same server whereas active links can run the command on the same or a different server.

- The result of a DDE request—for active links only.
- The result of a filter API plug-in service—for filters only.
- “Consume” the results of an external Web Services—for filters only.

If the Set Fields action is an alternate action for an escalation, the only fields that can be set are display-only fields.

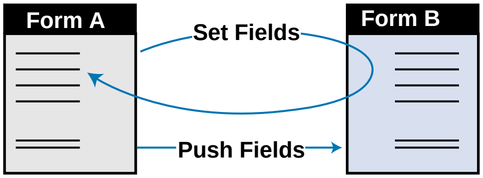
Because filters and escalations execute with administrator permissions, field values set through a filter are updated regardless of whether the user has permission to update the field.

Push values to fields on other forms

This action changes the values of fields in another form to the values in the current form; in other words, it "pushes" the values from the current form to another form. The action can also change the value to a keyword or function.

You can use Push Fields to set values in related requests or to create requests that are associated with the current one. For example, you can create multiple work orders for a telephone connection, network address, and new furniture when a new employee is hired.

Figure 5-3: How the Set Fields and Push Fields actions work



*The **Set Fields** action pulls data from another source into the current form.*

That other source can be a different form, among other options.

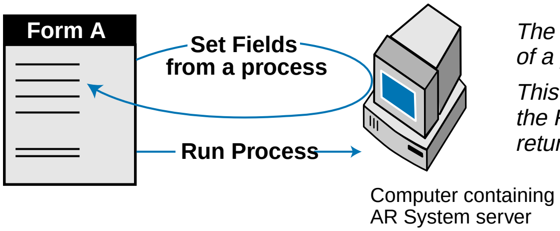
*In contrast, the **Push Fields** action sends data from the current form to a different form.*

Run a separate process

This action runs a program that you specify, either on the server for filters and escalations, or on the Windows client or server for active links. For example, a filter can send a page, or an active link can launch a browser on a user's Windows desktop.

You can have the process accept values from a request or from keywords. For example, the process could get the number of a pager from a field in the current request.

Figure 5-4: How the Run Process action contrasts with the Set Fields action



*The **Set Fields** action can insert the results of a process into the current form.*

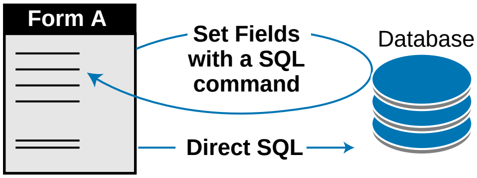
This "out and back" operation contrasts with the Run Process action, which does not return results.

Run an SQL command

All three workflow components can execute SQL commands to directly manipulate the database; the commands do not return data from the database.

As an example, suppose the AR System is integrated with another product and they share the same database. You could use this action to change data in the other product's database tables. In contrast, the Set Fields SQL command could *read* data *from* the other product's tables.

Figure 5-5: How the Direct SQL action contrasts with the Set Fields action



*The **Set Fields** action can insert the results of a SQL query into the current form.*

*The **Direct SQL** action, in contrast, affects the database without returning data.*

Perform an alternate action in the processing sequence

All workflow is performed in a specific order—the execution order. However, there are times when it is useful to jump to a different place within the flow and continue executing. The Goto and Go To Guide Label actions are similar to the Goto command in programming languages. These actions allow you to jump to a fixed location or to a location indicated by a field value. With this functionality, you can jump out of the execution flow (for example, skip to order 1000). This is useful if you encounter an error and want to skip the remaining processing or if you have found the answer and do not need to perform further steps that search other places.

You can set up a loop that will prompt the user, check the value, and then jump back to prompt again if the value is out of range or incomplete. The workflow will continue to loop until a legal value is specified.

Start the specified guide

Start execution of a specified guide. The active links or filters in the guide are performed, and when completed, return control to allow the next action to be performed.

Suspend guide processing until the user responds

On Windows, an active link guide can pause until users indicate that they want to continue, by pressing the tab key, selecting the Continue Guide menu option, or clicking the Continue Guide button.

Exit the guide

This action causes the current guide to terminate. “Nested” guides (guides within guides) can also be terminated with this action.

Open a window

This action opens a new window of any type. The action specifies what data or qualification is to be passed to the new window to determine its contents upon opening. The action can open a new window to allow submission with some data loaded as default information. Or, it can open a Modify window with entries matching a specified qualification loaded for modification.

This action can also be used to open a special type of window called a dialog box. A dialog box is a window that displays information or asks for data that you need the user to handle immediately before continuing other work. The user must complete the requested operation and is not allowed to work with any other window belonging to the application until the dialog box has been dismissed. Data can be passed between the dialog box and the window that calls it. For this type of window, processing of active links from the calling window is suspended until the dialog interaction is complete.

Close the current window

This action closes the current window.

Commit changes to the current window

This action performs the current window operation. For Submit and Modify windows, this action saves the data to the database. For Search windows, this action performs a search. For dialogs, this action sends the return values to the window that called the dialog box.

Change characteristics of a field

An active link can change the following characteristics of a field:

- Move the cursor or keyboard focus to the field. For example, after a user fills in the Employee ID field, two more fields (first and last name) could be automatically filled in by using a Set Fields action and the cursor then moved to the Problem Category field.
- Allow the field to be visible or hidden. For example, you can design an active link that hides all fields related to telephone problems if a user is reporting a network problem.
- Change how the field can be accessed—whether it is read only, read/write, or disabled (grayed out). For example, you can make all fields in a request read-only after the request is closed.
- Change the color of the field label or trim text.
- Change the character menu attached to a data field. For example, a form for scheduling a meeting could have a field for the building. Depending on which building you are going to meet in, the menu of meeting rooms attached to the meeting room field would change.
- Refresh the data in a table field.
- Change the label of a field.

Use DDE to connect to other applications

Dynamic Data Exchange (DDE) is a method for exchanging data between applications on the Windows platform.

An active link can:

- Direct another application to execute a specified command. For example, a support person who uses BMC Remedy User most of the day could have buttons on the form that are labeled Open Email and Open Word-Processing Program. Choosing a button would open the correct application in an appropriate state.
- Send data to another application. For example, the number of support calls related to desktop applications, network problems, or hardware could be sent to a spreadsheet application to be formatted as a pie chart.

Use OLE to connect to other applications

Object linking and embedding (OLE) is a method for exchanging data between applications on the Windows platform. OLE and DDE provide similar benefits, but programs that support OLE usually have more capabilities.

Firing conditions: what causes workflow to act

Firing conditions determine when workflow actions take place. In the case of active links and filters, you specify what event triggers the component; in the case of escalations, you specify the points in time that trigger the component. For all three workflow components, you can further refine the firing condition by adding a qualifying statement that tells the system which set of actions to run if the additional criteria are met and which set to run if the criteria are not met. In this section, filter and active link firing conditions are described first (along with several important considerations related to firing conditions), then qualifications, and then the escalation firing conditions.

Active link and filter firing conditions: event-based

The following table summarizes the firing conditions for active links and filters.

Firing Condition	Active Link	Filter
Window Open	✓	
Window Close	✓	
Copy To New	✓	
Window Loaded	✓	
Display	✓	
Un-Display	✓	
Set Default	✓	
Return/Table Dbl-Clk	✓	
Menu/Row Choice	✓	
Button/Menu Item	✓	

Firing Condition	Active Link	Filter
Interval	✓	
Gain Focus	✓	
Lose Focus	✓	
Search	✓	
Get Entry		✓
Submit	✓	✓
Modify	✓	✓
Delete		✓
After Submit	✓	
After Modify	✓	
Merge		✓
Event	✓	

Each of the firing conditions is described in the following sections. The icons indicate which firing conditions apply to which workflow components.

Window Open

An active link with this firing condition executes when a user opens a form or a dialog box, or changes a form to a different mode. This is especially useful for establishing initial environments. It executes *before* any data is loaded into the window.

Window Close

An active link with this firing condition executes when a user closes a window or a dialog box or changes the mode of a form. If an error is returned, the window will not close.

Copy to New

An active link with this firing condition executes after the data has been sent to the new window, so that the administrator has the ability to use this data in the workflow.

Window Loaded

An active link with this firing condition executes after a New or Search window has been opened after all data that is going to be loaded on that window has been loaded.

Display

An active link with this firing condition executes when a user displays a request—that is, after data for an entry is loaded into the window.

Un-Display

An active link with this firing condition executes when an entry is removed or unloaded from the window.

Set Default

An active link with this firing condition executes when a user selects the Set to Default option or when the AR System loads defaults after a New or Search mode has started.

Return/Table Dbl-Clk

An active link with this firing condition executes when a user presses Enter (or selects a radio button if the field is a selection list field) in a specified field or when a user double-clicks a row, or selects a row and presses Enter, in a table field to “drill down” to the source form.

Menu/Row Choice

An active link with this firing condition executes when a user selects a choice from a character menu associated with the field you specify or chooses a new row in a table.

Button/Menu Item

An active link with this firing condition executes when a user selects either the button, menu item, or toolbar button associated with the active link.

Interval

An active link with this firing condition executes repeatedly in BMC Remedy User when an associated interval value has been chosen and the interval counts down to zero.

Gain Focus

An active link with this firing condition executes when a user moves the cursor to a field or when a Change Field action does the same.

Lose Focus

An active link with this firing condition executes when a user moves the cursor out of a field or when a Change Field action does the same.

Search

An active link with this firing condition executes when a user performs a search operation. If you choose this option, the active link executes *before* the search operation; therefore, if the active link criteria has not been met, the search operation will not be performed. You can use this firing condition to prevent users from running inefficient searches.

Get Entry

A filter with this firing condition executes when a request is retrieved by the system.

Submit

Active links and filters with this firing condition execute when a user submits a new request but *before* the request is sent to the AR System server (in the case of active links) or *before* the request is sent to the database (in the case of filters).

Modify

Active links and filters with this firing condition execute when a user modifies an existing request but *before* the request is sent to the AR System server (in the case of active links) or *before* the request is sent to the database (in the case of filters). An active link will *not* execute during a Modify All operation.

Delete

A filter with this firing condition executes when the user chooses to delete a request but before the request is actually deleted from the database.

After Submit

An active link with this firing condition executes after a user submits a new request *and* after the request is written to the database. If the submission fails, the active link will not be executed.

After Modify

An active link with this firing condition executes when a user modifies an existing request *and* after the request is written to the database. An active link will *not* execute during a Modify All operation. If the modification fails, the active link will not be executed.

Merge

A filter with this firing condition executes when a request is merged into the database. A request can be merged with, for example, BMC Remedy Import, Distributed Server Option, or API programs.

Event

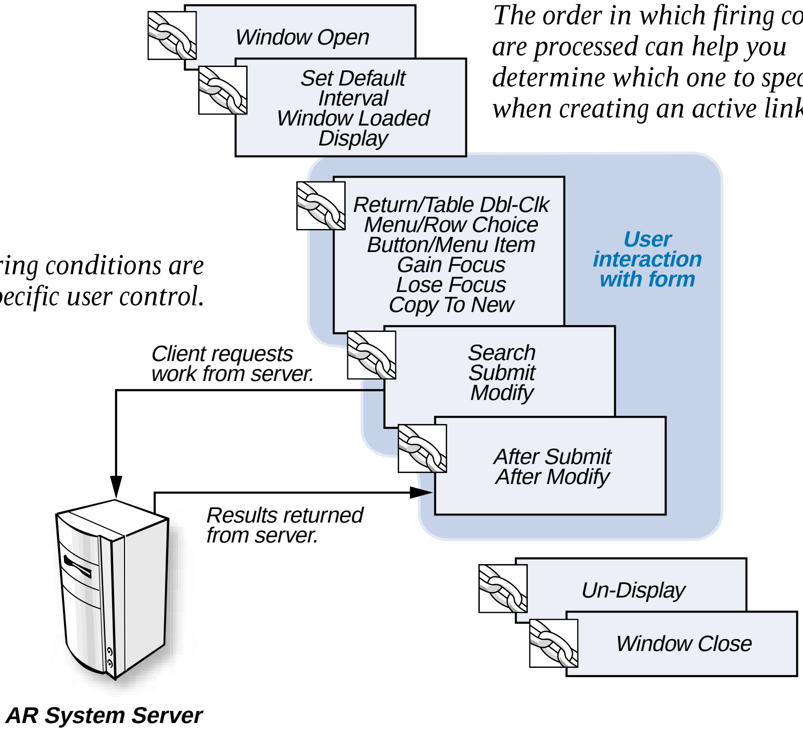
An active link with this firing condition executes when an event is received by a window. An event is a message sent from another window that generally indicates that the sending window has finished some activity, such as modifying data. Workflow on the sending window specifies what activity triggered the event, and which window should receive the event. For example, many applications include multiple windows containing interrelated data; data associated with one window might appear in another. In such cases, you could create workflow to send an event from some other window to the current window to signify that data has changed. The current window can react to that event by updating its contents or refreshing a table field.

Processing of active link firing conditions

The active link firing conditions have a certain order in relationship to one another and to the interaction between the client and server, as shown in the figure in this section. You can use this implicit order to decide at what point you want the active link to execute. For example, you would choose to set field values on Window Open if those values were required for workflow processing *before* the request is actually displayed. However, any values you want the user to see when the request is displayed should be set with the Display firing condition. For example, an active link that fires on Window Open could check that the user has permission to open a Modify All window and could generate an error message, thus preventing the window from opening if the user does not have permission.

As the diagram illustrates, certain firing conditions are dependent on how the user interacts with fields on the form. For example, if the user does not press a particular button, that active link will not fire. That is what it means for the user to “control” whether the active link fires. Active links that are not under the user’s control fire whenever the user pursues a task to its finish. That is, if the user follows the normal steps, from opening a window through closing the window, the active links not under explicit user control will *always* fire. (Of course, if a user decides not to submit or modify the request, the active links that fire on Submit or Modify will not be executed.)

You can use the “user-controlled” firing conditions to provide additional helpful information, such as a list of nearby printers. The other firing conditions can be used to ensure that consistent, complete data is maintained regardless of whether the user performs certain actions. For example, you can use an active link to retrieve the host name of the client and copy it into a field in the form whenever a new request is submitted to guarantee that every submitted request includes the host name.



The order in which firing conditions are processed can help you determine which one to specify when creating an active link.

These firing conditions are under specific user control.

Execution order of active links and filters

More than one active link can fire on the same firing condition. The same is true for filters. For each component, you can specify what order you want it to fire in relation to the other active links or filters. One reason to do this is that the output of one active link can affect another active link. By carefully ordering a group of active links, you can perform very complex operations.

When the active links and filters are bundled into active link guides and filter guides, execution order within the guide is ignored. Instead, workflow executes in positional order within the guide. This allows the guide procedure to run without worrying about affecting the order of workflow outside the guide.

Workflow and join forms

There are aspects of using workflow with join forms that should be noted.

Filters can be affected by join forms. Usually, you attach filters to the underlying forms if they affect data validation, notification, or integration. However, a filter that uses a qualification that relies on fields in both underlying forms must be attached to the join form. If filters are attached to both the join form and its underlying forms, they fire in this order:

1. Filters attached to the join form.
2. Filters attached to the primary form.
3. Filters attached to the secondary form.

Qualifications

Specifying a qualification when you create an active link, filter, or escalation allows you to define the data condition that causes that workflow component to take action. You can use qualifications to check values of fields, amount of time passed since a certain event, and many other factors. For example, a qualification could check if the priority of a request is High or Critical or if the day is a weekend day.

Qualifications have a different impact on active links and filters than they do on escalations.

For active links and filters, the qualification simply refines the firing condition further. For example, an active link can fire whenever a specific field is filled in (firing condition), or it can fire whenever the field is filled in *and* whenever the client machine is a Windows platform (qualification). In contrast, escalations fire whenever the specified time arrives, and, without a qualification, *every* request in the form to which the escalation is attached will result in the escalation actions being performed. The qualification is an essential part of most escalations, not simply a refinement.

For example, if an escalation simply sends a notification every eight hours (firing condition), the notification would be meaningless. However, a meaningful escalation could check every eight hours (firing condition) if twelve or more hours have elapsed since a request was entered, *and* no solution found (qualification), and then send a notification.

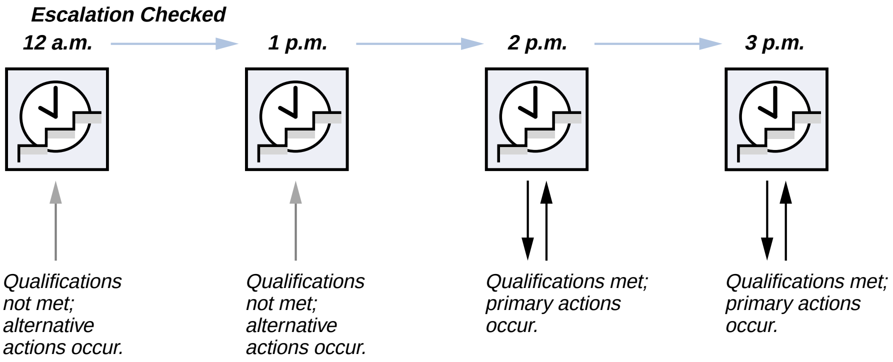
For filters, the qualification can check the value of a field in the database, in the current transaction, or both. This makes it possible to check whether the value of the field is changing. For example, if you have a business rule that help desk requests can only be closed if they have been fixed, a filter could check all transactions that change a request to Closed. If the database value of the request is Fixed, the request can be modified; otherwise the change is not allowed.

Escalation firing conditions: time-based

In contrast to active links and filters, which react to events on the client or server, escalations respond to the passage of time. You can trigger an escalation at a specific time, such as every Monday, or at a time interval, such as every eight hours.

When the specific time or time interval that you have specified arrives, the server determines if the escalation qualification has been met. If it has been met, that means that one or more matching requests have been found in the database, so the primary actions are executed in order. The server then waits for the next interval. If the qualification has not been met, no requests have been found, so the alternate actions are executed in order. Again, the server then waits for the next interval. This process is illustrated in the following figure.

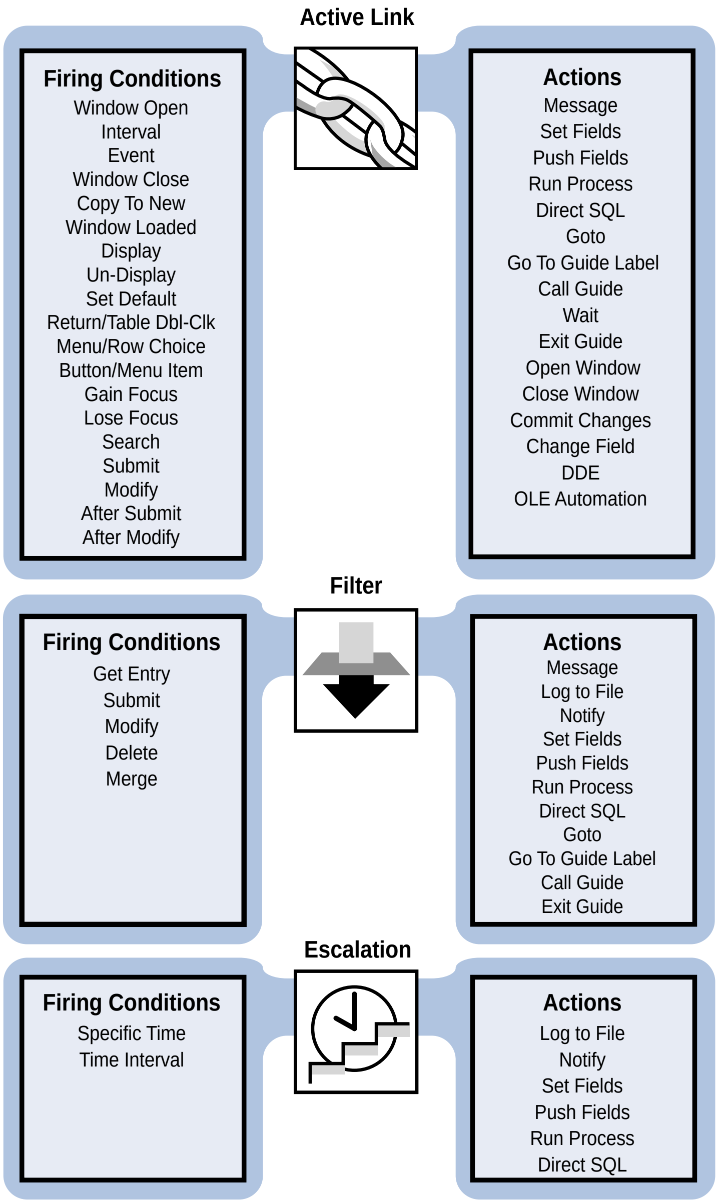
Figure 5-6: Example of how escalations work


Escalation Checked
12 a.m.
1 p.m.
2 p.m.
3 p.m.
Qualifications not met; alternative actions occur.
Qualifications not met; alternative actions occur.
Qualifications met; primary actions occur.
Qualifications met; primary actions occur.

Summary

The following figure summarizes the workflow actions and firing conditions for each of the components.

Figure 5-7: Workflow firing conditions and actions



Chapter 6 Controlling access to AR System

Keeping information secure can be a major consideration in client/server environments. And it is sometimes a balancing act for administrators. You want to rigorously control who can access data, yet you do not want security to become so complex that it becomes an intrusive burden on your user community or difficult for you to implement.

AR System enables you to meet these seemingly opposing security goals. It provides a rich set of features that protect your data from unauthorized access. And it has a logical, multi-tiered access control structure that is straightforward for you to implement and for users to understand.

This chapter discusses access control in the AR System; it describes the purpose of groups and roles, the multi-tiered access control model, and how licensing affects access.

Understanding access control in AR System

AR System enables you to control which users can access data and perform certain actions such as modifying a request or triggering an active link. The access users have is determined by:

- Which groups users belong to.
- The type of licenses users have been granted.

AR System uses a multi-tiered approach to control access at the following points:

- Server
- Form (or table)
- Request (or row)
- Field (or column)
- Active link and active link guide

This hierarchical approach provides you with a wide range of control over data access, enabling you to restrict access at the highest levels (server and form) and more finely at the request and field levels. Because you are able to refine your data access criteria so precisely, you can use a single form for many different purposes by simply setting the appropriate permissions.

The user and group model

Individuals who need to access AR System are registered as users by an administrator. The administrator then assigns these users to access control groups.

Each access control group has specific permissions associated with it that determine whether the members of the group can access particular application components such as forms, requests, fields, active links, and active link guides. Groups further determine the *kind* of access group members have to these components—change, view, or no access.

Users are assigned to groups based on their common need to access information. For example, you might create a group called Employee Services Staff in which members are permitted to view and change only certain fields in an Employee Information form. You might have another group called Employee Services Managers in which the group is permitted to view and change *all* fields in the Employee Information form, including Salary information.

If you choose, you can permit unregistered users to access AR System as **guests**. Guests are members of the Public group, as described in the section “Types of access control groups” on page 101.

You can create any number of groups within AR System to enforce access control. In addition, AR System defines seven special groups that perform specific functions, as described in the section “Types of access control groups” on page 101.

In deployable applications, access permissions are based on **roles**. In these applications, roles have specific permissions to forms, fields, active links, and so on, similar to the access permissions for groups. Users are not assigned directly to these roles; instead, users are assigned to groups and then the groups are “mapped to” (or associated with) the various roles in the application. Roles make deployable applications easy to install on a variety of servers. For more information, see “Deployable applications” on page 46 and “Role-based access in deployable applications” on page 114.

Note: For simplification, the following sections discuss user access in terms of *group* permissions and user membership in those groups. In a deployable application, which uses *role* permissions, user access is ultimately determined by which groups are mapped to the roles in the application.

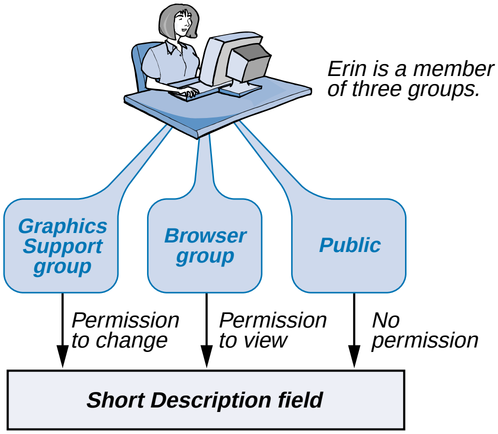
Membership in multiple groups

Users often belong to multiple groups in an organization. They inherit permissions from each of the groups they belong to.

An important concept to understand is that if you give a group permission to access a form, field, request, active link, or active link guide and a user is a member of that group, then the user *has access*, even if the user belongs to other groups that are not given access.

For example, if Erin is a member of the Graphics Support group, which has *change permission* for the Short Description field in the Graphics Tools form, and is also a member of the Browser group, which has *view permission* to the field, Erin will be able to change the field by virtue of her membership in the Graphics Support group (see the following figure).

Figure 6-1: Example of how permissions work



Because Erin belongs to a group with permission to change the Short Description field on the Graphics Tools form, she can change the field even though she belongs to other groups that do not have change permission.

The additive nature of permissions

Access control in AR System is *additive*. Each user starts out with no access permissions; administrators then add permissions as needed. In this way, AR System implements strict access control. Administrators must make a conscious decision to grant access to specific groups on a case-by-case basis.

Note: Administrators can change AR System's default permissions (no access) so that each time they create new forms, fields, active links, and active link guides, groups will automatically have the default settings they specify.

Types of access control groups

There are seven predefined groups within AR System that confer specific access rights to their members. These groups fall into two categories:

- Explicit groups
- Implicit groups

An **explicit group** is one that you must assign users to. The predefined explicit groups are:

- Administrator
- Subadministrator
- Customize

In addition to these predefined explicit groups, any **regular** and **computed groups** that you create are explicit groups.

An **implicit group** is one that a user is automatically (or implicitly) a member of by virtue of the contents of certain fields in a request. You cannot assign users to implicit groups. The predefined implicit groups are:

- Public
- Submitter
- Assignee
- Assignee Group

In addition to these predefined implicit groups, any dynamic groups that you create are implicit groups.

Explicit groups

The following sections describe each of the predefined explicit groups and computed groups.

Predefined explicit groups

There are three predefined explicit groups in AR System.

Administrator

Administrators are the persons who manage AR System. Because their functions are so wide-ranging, members of the Administrator group are allowed unlimited access to AR System. Members of this group can do the following tasks:

- Create, modify, and delete AR System objects (applications, forms, fields, active links, and so forth).
- Add and delete users and groups (including the Administrator group).
- Configure and assign preferences for users and groups.
- Assign access privileges to users and groups.
- Access any data in any form on the AR System server.
- Perform administrative functions on the AR System server.

Note: Each person assigned to the Administrator group must have a fixed write license. See the section “How licensing affects access control” on page 115 for more information about licensing.

Subadministrator

Some organizations have a structure in which individuals manage *portions* of AR System. These individuals are referred to as subadministrators. A subadministrator’s responsibilities can vary from managing just a single form to managing all of the forms in several departments.

Members of the Subadministrator group can do the following tasks:

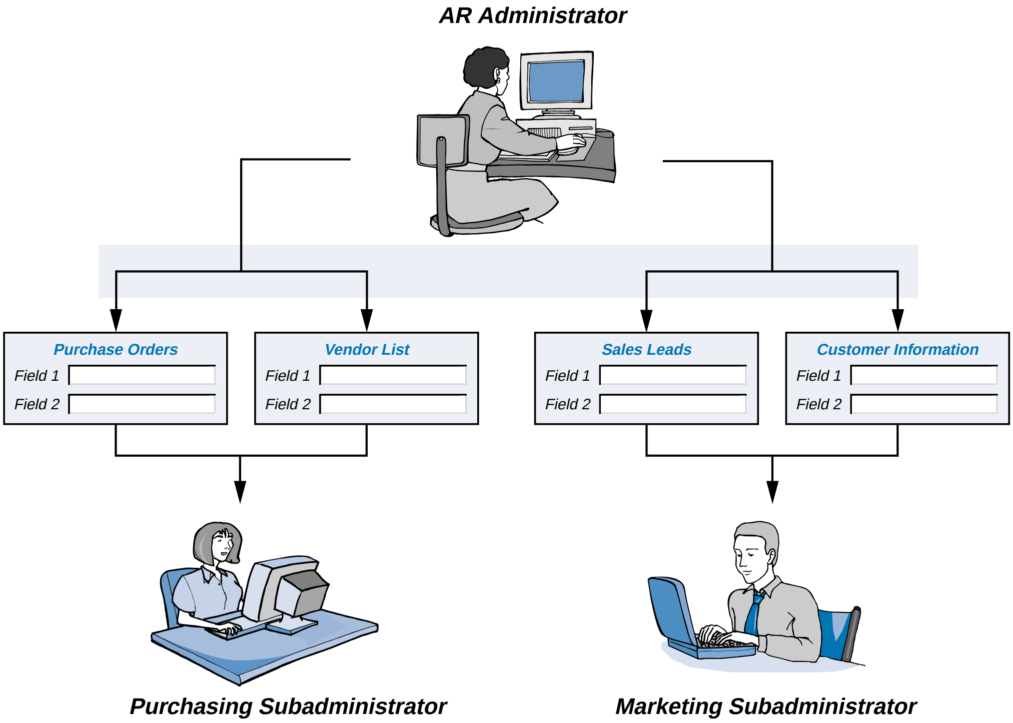
- Administer forms to which they have been given access.
- Create, modify, and delete filters, active links, and escalations connected to forms to which they have been given access.
- Create, modify, and delete menus.
- Create new forms.
- Create applications (using forms to which they have been given access).
- View server information settings.

Members of the Subadministrator group cannot do the following tasks:

- Change server information settings.
- Administer forms to which they have not been given specific access.

Note: Each person assigned to the Subadministrator group must have a fixed write license. See the section “How licensing affects access control” on page 115 for more information about licensing.

Figure 6-2: Subadministrators in AR System



Customize

Members of this group have the right to customize the layout of their forms. If users do not belong to this group, they cannot change the form layout set up by the administrator or subadministrator. Those in the Administrator group are automatically part of the Customize group.

Computed groups

A computed group is a group whose membership is based on which users belong to, or do not belong to, explicit groups specified in a Boolean expression. For example, you can create a computed group with an expression such as

```
(“Sales” OR “Support”) AND NOT “IT Staff”
```

The membership of this computed group would be all of the users who are members of either the group `Sales` or the group `Support`, excluding members of the group `IT Staff`.

Computed groups make groups easier to manage, because they allow you to create new groups based on existing groups, without the need to update the lists of users.

Implicit groups

The following sections describe each of the predefined implicit groups and dynamic groups. For more information about accessing requests and the groups Submitter, Assignee, and Assignee Group, see the section "Controlling access to requests" on page 110 later in this chapter.

Predefined implicit groups

There are four predefined implicit groups.

Public

All AR System users (registered users and guests) are automatically members of the Public group.

Submitter

When a user creates a request (that is, the user's name appears in the Submitter field of the request), that user automatically belongs to the Submitter group *for that request*. The Submitter group provides you with a way to limit access to requests and fields to the *individual* who submitted the request.

Assignee

When a user is assigned to a request (that is, the user's name appears in the Assigned To field of the request), that user automatically belongs to the Assignee group *for that request*. The Assignee group provides you with a way to limit access to requests and fields to the *individual* who is assigned to handle the request.

Assignee Group

When a user belongs to a group that appears in the Assignee Group field, or when a user's name appears in that field, that user automatically belongs to the group Assignee Group *for that request*. This group provides you with a way to limit access to requests and fields to specific *groups* of people. All users who belong to the group specified in the Assignee Group field of a request will be able to access the request.

Dynamic groups

Dynamic groups extend the Assignee Group functionality to include other specially designated fields. In addition to groups, individual users can be named as members of a dynamic group. For example, when you create a dynamic group, the user specified—or all users who belong to the group specified—in the field associated with that dynamic group will be able to access the request.

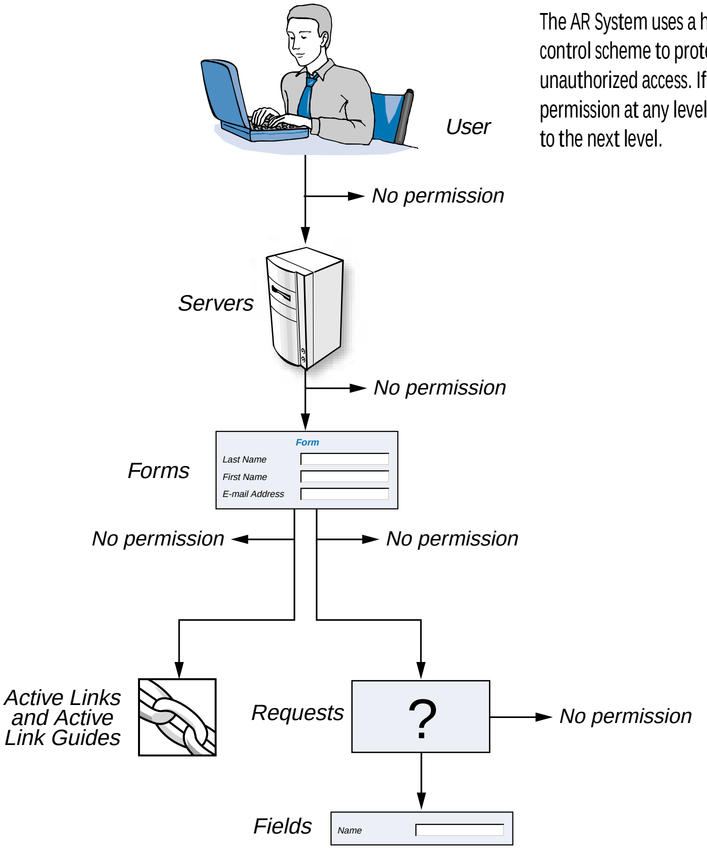
Multi-tiered access control model

As mentioned earlier, AR System uses a multi-tiered approach to control data access. You can think of this as a hierarchical trail that users traverse that determines whether they can access AR System and whether they have been granted permission to specific forms, fields, requests, active links, and active link guides.

At each level in this hierarchy, a user's permissions are checked. If access is permitted, a conceptual "gate" opens that allows the user to proceed to the next level. If access is denied, the user cannot proceed.

For example, if a user is denied access to a form, that user cannot see any fields associated with the form. As another example, a user's ability to access a request hinges upon whether the user is a member of a group that has been granted access to a key core field—Request ID. (For more information about accessing requests, see the section "Controlling access to requests" on page 110.)

Figure 6-3: Access control in AR System

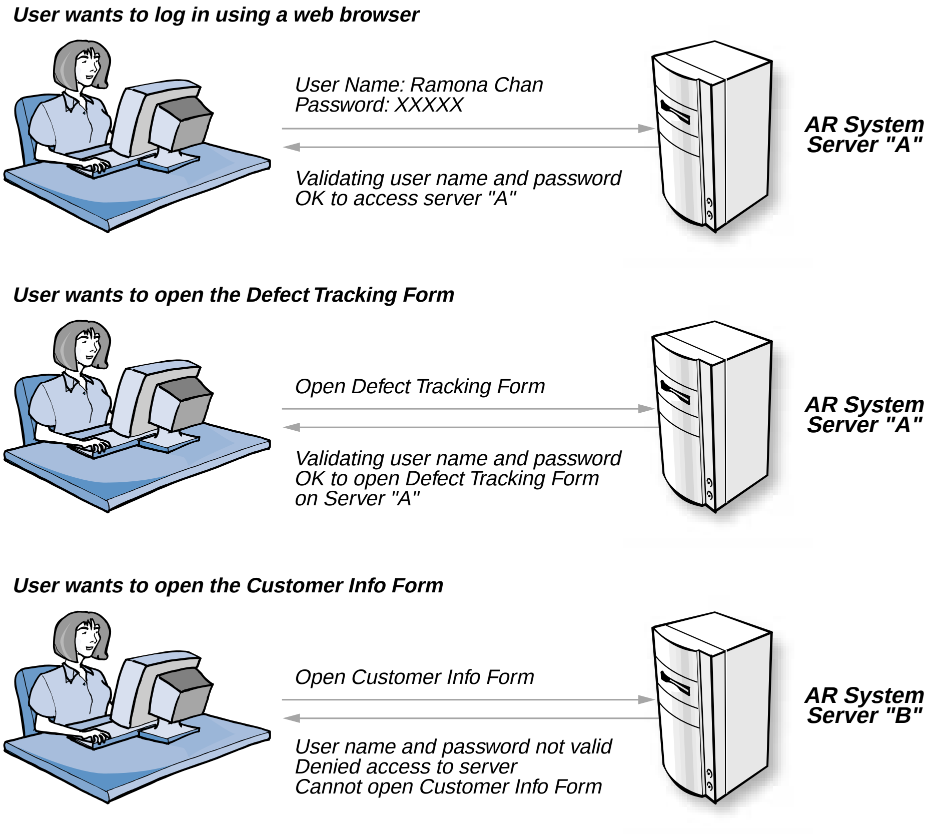


The AR System uses a hierarchical access control scheme to protect information from unauthorized access. If users are denied permission at any level, they cannot proceed to the next level.

Controlling access at AR System server level

At the top-most level, users must pass an initial checkpoint when they start an AR System client tool (such as BMC Remedy User or a browser). At this point, users must enter a valid name, a password, and optionally, an authentication string which might be used by your site. AR System servers check the user name, password, and authentication string each time a client requests a transaction, such as opening a form or changing a field.

Figure 6-4: Server validation for each request


User wants to log in using a web browser
User Name: Ramona Chan
Password: XXXXX
Validating user name and password
OK to access server "A"
AR System Server "A"
User wants to open the Defect Tracking Form
Open Defect Tracking Form
Validating user name and password
OK to open Defect Tracking Form on Server "A"
AR System Server "A"
User wants to open the Customer Info Form
Open Customer Info Form
User name and password not valid
Denied access to server
Cannot open Customer Info Form
AR System Server "B"

Controlling access to forms

As the administrator, you give groups access to forms based on their need to view or change information. If a group is not allowed access to a form, members of that group will not be able to view or access the form.

Once you have granted a group permission to access a form, you then determine whether you want the form to be visible or hidden. Why would you choose to grant access and then hide a form? If you want the group to access the form through workflow only, for example, through an active link action, then you would specify hidden access at the form level. This prevents users from opening a form directly.

After granting permissions for a form, you then specify which fields (and if applicable, active links and active link guides) the group can access. Remember: you specify access individually at each level of access control.

Controlling access to fields

You can control access to every field on a form (including non-data fields such as trim fields, table fields, and active link control fields). At the top level, you determine if a group can access a field at all. If a group cannot access a field, the field does not appear when members of the group open the form.

For data fields, you also need to determine whether a group should be able to simply view the information in the field or also change the information. In addition, you can make a field visible to users or hide the field so that it is accessible through workflow only.

The field permission you grant to an explicit group applies to the field for *all* requests associated with the form. For example, if the Customer Support group has change access to the Customer Contact field in the Customer Info form, that group can change the Customer Contact field for all requests associated with the form.

Controlling access to active links

In addition to being able to control access to data, you can also control access to active links, which can trigger a variety of workflow actions. For example, you can allow one group to trigger an active link, while denying access to another group. Or you might want your support staff to trigger several active links appropriate to their work, but not allow users to trigger those same active links.

Note that just because a group can access an active link, the group does not automatically have access to the field associated with the active link. You must grant access to fields independently of the active links associated with them. For active links that fire when the user selects a button or menu item, the user must have permission to both the button or menu item *and* the active link to trigger the active link.

If a user has access to an active link associated with a button or menu item, but not to the button or menu item itself, the button or menu item will not be displayed on the form. If a user has access to the button or menu item, but not to the active link associated with it, the button or menu item will not trigger the active link.

Controlling access to active link guides

When you create an active link guide, you define the groups you want to have access to it. To access an active link guide, a user must have permission to each active link in the active link guide and to the active link guide itself. If a user has access to all active links associated with a guide, but not to the active link guide itself, the guide will not appear at all. If a user can access an active link guide, but cannot access all of its active links, only the active links to which the user has permission will fire, and the active link guide might not perform as designed.

Controlling access to requests

There are times when you want to strictly control who can access requests associated with a form. For example, you might want managers to be able to access requests with confidential employee information, but not want to make the information available to anyone else. Or, maybe you provide an outsourcing service in which you use AR System as the central help desk for several companies, and you do not want one company to see the requests from another company.

The key to controlling access to requests is defining who can access the Request ID field (a core field with field ID 1). The Request ID field is the gatekeeper to all requests in the form. If a group does not have access to this field, the group cannot access any requests associated with the form regardless of how you set permissions on other fields in the form.

Using implicit groups to control access to requests

Three key implicit groups in AR System—Assignee, Submitter, and Assignee Group—and dynamic groups that you create are particularly useful when you want to control access to requests.

The first two groups—Assignee and Submitter—allow *individuals* to access specific requests and fields. A user belongs to the Assignee group if that user's name appears in the Assigned To field (a core field with field ID 4). If you give the Assignee group permission to access the Request ID field, then a user will be able to access all requests assigned to him or her. Similarly, a user belongs to the Submitter group if that user's name appears in the Submitter field (a core field with field ID 2). If you give the Submitter group access to the Request ID field, then a user can access all requests he or she submitted.

But what if you want a *group* of people (or a list of users) to access specific requests associated with a form? If you give an explicit group, such as Marketing Managers, access to the Request ID field, members of the group will be able to access *all* of the requests associated with the form. However, if you use the Assignee Group designation, you can restrict a group's access to just the requests you specify.

The group Assignee Group and dynamic groups work a little differently from the other implicit groups in that you have to add an associated field to the form—Assignee Group (field ID 112) for the Assignee Group, or fields in the dynamic field ID range (field IDs 60000 to 60999) for dynamic groups.

The following examples illustrate how you might use implicit groups to control access to requests and fields.

Example: using the group Assignee Group to control access to requests

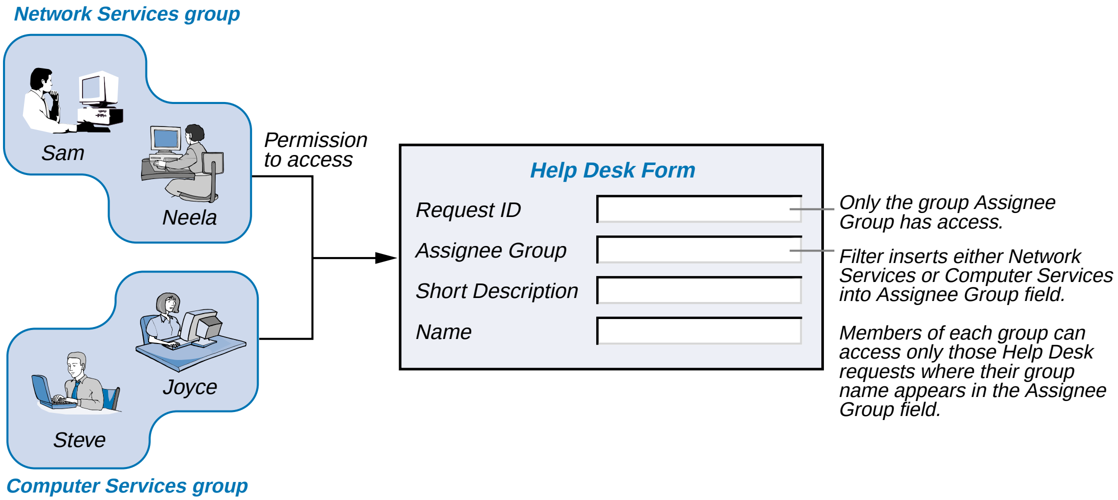
In this example (shown in the figure in this section) there are two departments, Network Services and Computer Services, that use the same form for tracking help desk requests. However, each department should only be able to access their own requests. The administrator does the following to accomplish this goal:

- Creates two groups: Network Services and Computer Services. Assigns each user to only one of these groups.
- Gives the two groups access to the Help Desk form.
- Adds the Assignee Group field (with field ID 112) to the Help Desk form.
- Restricts access to the request by only allowing the group Assignee Group to access the Request ID field.
- Gives the group Assignee Group permission to appropriate fields in the form.
- Creates workflow (using a filter, for example) that inserts one group name—either Network Services or Computer Services—into the Assignee Group field, as appropriate.

Members of each group—Network Services and Computer Services—can access only those help desk requests in which their group name appears in the Assignee Group field.

If you wanted someone to access requests for both departments, you would make the person a member of both groups, Network Services *and* Computer Services.

Figure 6-5: Assignee Group access to requests


Network Services group
Sam
Neela
Joyce
Steve
Computer Services group
Permission to access
Help Desk Form
Request ID
Assignee Group
Short Description
Name
Only the group Assignee Group has access.
Filter inserts either Network Services or Computer Services into Assignee Group field.
Members of each group can access only those Help Desk requests where their group name appears in the Assignee Group field.

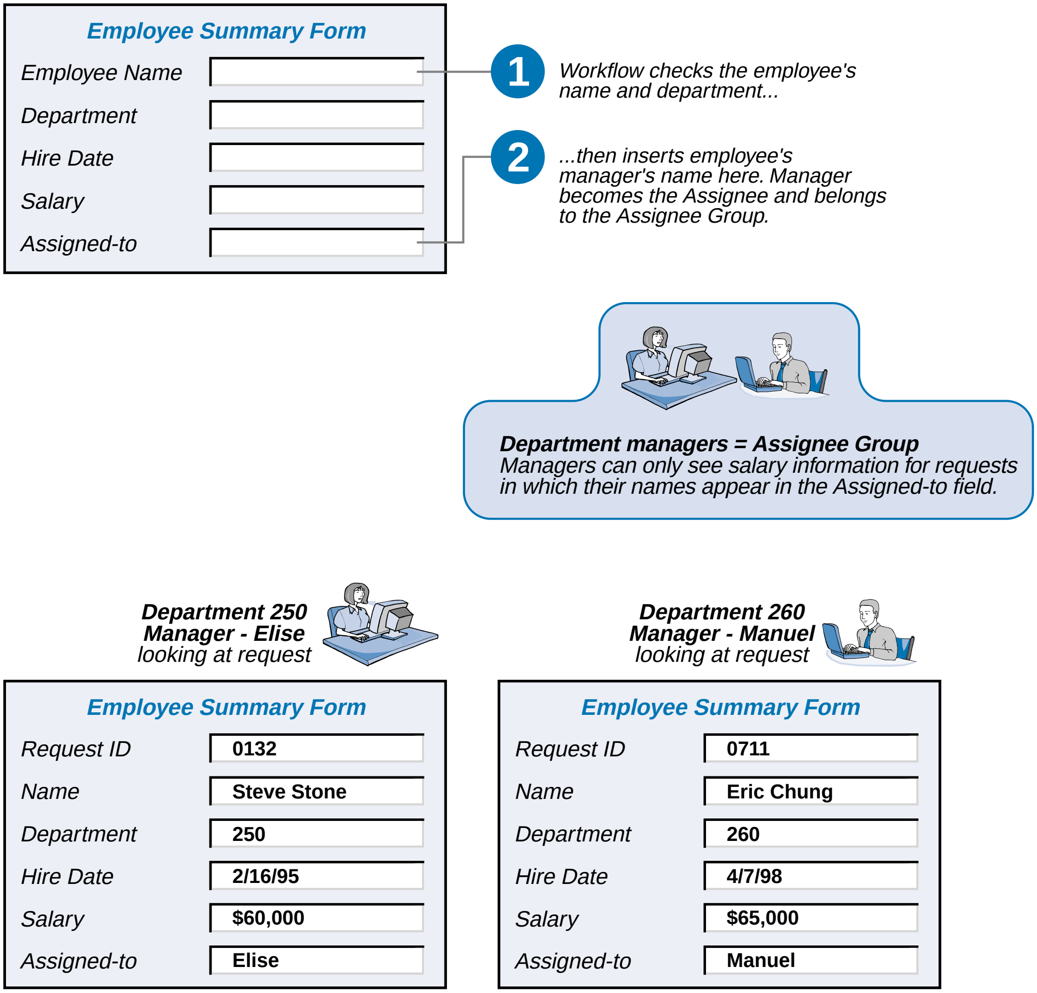
Example: using the group Assignee to control access to fields within a request

This example illustrates how the implicit group Assignee is used to control access to a specific field—Salary—within requests. As shown in the figure in this section, there is one form, called Employee Summary, that is used to track information about employees. The administrator needs to control access to the form so that department managers can see only the salary information for the employees who report to them. To accomplish this, the administrator completes the following tasks:

- Creates workflow that verifies an employee's name and department and then automatically places the employee's manager's name into the Assigned To field when the request is submitted. Each department manager then becomes the Assignee for that request and belongs to the Assignee Group. Employees, in essence, are assigned to their respective managers.
- Makes sure that the Assignee Group has view permission to the Request ID field and change permission to the Salary field. (Note that if the managers already have access to the Request ID field by virtue of their membership in another group, the administrator does not need to give the Assignee group access to the Request ID field also.)

When department managers display requests for the Employee Summary form, they will have permission to the Salary field for *only* those requests where their name appears in the Assigned To field.

Figure 6-6: Assignee access to fields



Employee Summary Form

Employee Name	
Department	
Hire Date	
Salary	
Assigned-to	

Workflow checks the employee's name and department...

...then inserts employee's manager's name here. Manager becomes the Assignee and belongs to the Assignee Group.

Department managers = Assignee Group
Managers can only see salary information for requests in which their names appear in the Assigned-to field.

Department 250
Manager - Elise
looking at request

Employee Summary Form

Request ID	0132
Name	Steve Stone
Department	250
Hire Date	2/16/95
Salary	$60,000
Assigned-to	Elise

Department 260
Manager - Manuel
looking at request

Employee Summary Form

Request ID	0711
Name	Eric Chung
Department	260
Hire Date	4/7/98
Salary	$65,000
Assigned-to	Manuel

Example: using the group Assignee Group to control access to fields within a request

This example is similar to the previous one in that there is one form, called Employee Summary, that is used to track information about employees. In this case, the administrator needs to control access so that the managers of each division can see the salary information for all of the employees in their own divisions, but not see the salaries of employees in other divisions.

The administrator decides to use the implicit group Assignee Group to accomplish this by completing the following tasks:

- Creates a group for each division: Marketing Managers, Engineering Managers, Operations Managers, and so forth.
- Adds the Assignee Group field (with field ID 112) to the form.
- Gives the group Assignee Group permission to access the Request ID field and the Salary field (or makes sure that all the appropriate users have access to the Request ID field by virtue of their membership in other groups).
- Creates workflow that automatically places the appropriate division group name (Marketing Managers, Engineering Managers, Operations Managers, and so forth) into the Assignee Group field when an Employee Summary request is submitted.

Role-based access in deployable applications

Deployable applications use permissions based on roles. Roles are server objects similar to groups, except that roles are defined individually for each application, while groups are defined for the entire server.

Role permissions are assigned to each object in an application. Like groups, roles can be given change permission, view permission, or no permission as their highest potential access. Roles are “mapped” to (or associated with) appropriate groups on the server at the time the application is installed. This allows the same application to be installed on different servers that have different groups, without the need to re-define all of the object permissions in the application in every installation.

Different role-group mappings can also be defined depending on the development state of an application, such as Test or Production. This allows you to restrict access to an application while it is being developed or upgraded.

For more information about applications and permissions, see the *Form and Application Objects* guide.

How licensing affects access control

Although licensing is not a component of access control, licensing can affect whether users can actually perform an operation that you have granted them permission to perform. For example, if a user is a member of a group with Change permission to a field but you have not given the user an appropriate write license, the user will not be able to change the field.

There are four general classes of licenses you can assign to users: Read license, Restricted Read license, Fixed Write license, and Floating Write license. The base AR System product comes with three Fixed Write licenses, and unlimited Read and Restricted Read licenses. (You can purchase additional Fixed Write licenses and Floating Write licenses from BMC Software, Inc. or from an authorized reseller.)

A complete discussion of licensing is in the *Configuring* guide, but the elements related to access control are summarized in the following descriptions.

Read licenses

Within their assigned permissions, users with Read licenses can search for requests and display requests. In addition, administrators can configure the AR System server to enable users with Read licenses to do the following tasks:

- Submit requests
- Modify requests that they submitted

The ability to allow those with Read licenses to submit requests allows you to reserve write licenses for users who need to modify requests that they did not submit.

Restricted Read licenses

Within their assigned permissions, users with Restricted Read licenses can also search for requests and display requests. As with Read licenses, administrators can configure the AR System server to enable users with Restricted Read licenses to submit requests. Unlike those with Read licenses, users with Restricted Read licenses cannot modify *any* requests, including their own.

Also, unlike Read or Write licenses, a Restricted Read license allows users to access the AR System from multiple IP addresses (multiple machines) at the same time using the same user login account.

Fixed Write licenses

Fixed Write licenses provide users with all the capabilities of Read licenses plus the ability to modify existing requests that they did not submit (according to the user's assigned privileges). A Fixed Write license is associated with a user name, and is always "reserved" for that user. Those who have a Fixed Write license can use the AR System server at any time. AR System administrators and subadministrators must have a Fixed Write license. Users who need to frequently modify requests are also good candidates for Fixed Write licenses.

Floating Write licenses

Floating Write licenses are ideal for users who occasionally access the AR System and need to modify requests. These licenses provide the same capabilities as Fixed Write licenses; however, they are available on a first-come, first-served basis.

A user with a Floating Write license is temporarily logged in to the AR System with a Read license. When the user tries to search, modify, or submit a request, the AR System checks for an available Floating Write license token. If a token is available, the user is granted write access to requests. If no tokens are available, the user is notified and continues to use the Read license until a token becomes available. When a token becomes available, the user is automatically upgraded to a write license. Tokens are released when a user logs out or after a specific time interval passes with no activity since the token was acquired, whichever comes first.

License pools

Generally, Floating Write licenses are shared by all AR System users. However, administrators can define license pools to reserve a set of Floating Write licenses for a specific group of users. This gives you a way to prioritize the availability of Floating Write licenses. For example, you can allocate a number of licenses to department managers to ensure that they are not delayed in being able to approve essential requests. Users who do not belong to this group cannot acquire any of the reserved licenses.

Chapter

7 Expanding beyond the core AR System product

The core AR System product—the client tools (Windows and web-based), the mid tier, and AR System server—serves as the foundation for the BMC Remedy product line. BMC also offers add-on products that provide additional services and capabilities. Beyond the core environment, BMC provides solutions for IT service and customer relationship management. This chapter provides brief overviews of these products.

In addition to these products from BMC, a wide range of products has been developed by third parties for integration with AR System. Some of the most popular integration areas are discussed in this chapter, as well as the “hooks” used to integrate these products. For the latest details about products that have been integrated with AR System, see the product partners and product integration information about the BMC Remedy web page (referenced in the appendix).

BMC products

The following sections describe other products from BMC that are based on the AR System foundation.

BMC Remedy Approval Server

BMC Remedy Approval Server is a powerful and flexible approval engine that lets you link your approval, agreement, or acknowledgment process to any BMC Remedy solution. It automatically routes requests for approval to the right approvers, in the right order, and in the fastest possible time. Built-in contingency handling ensures that approvals are completed quickly and properly within the system. Requesters and others can check the status at any time. Customization of your approval processes such as changing approval cycles, models, or approvers does not require programming.

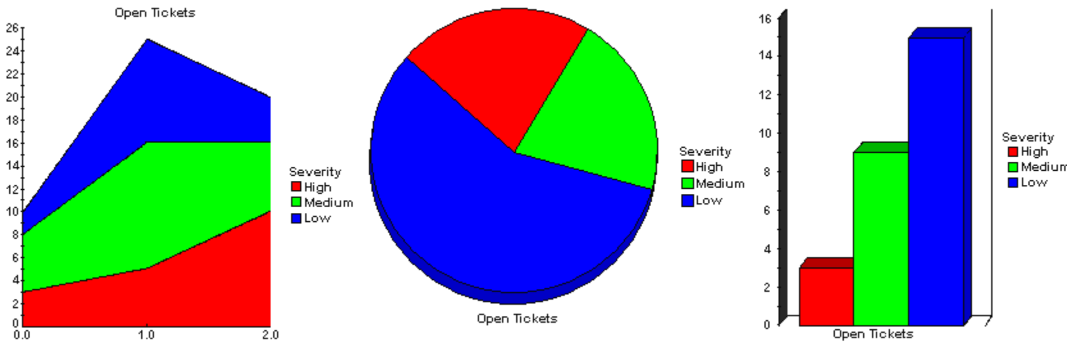
Distributed Server Option

The BMC Remedy Distributed Server Option (DSO) lets you send and receive data from both similar and dissimilar forms on physically separate servers. See "Distributing your AR System servers" on page 29 for more information about DSO.

BMC Remedy Flashboards

BMC Remedy Flashboards® works with AR System as a visual monitoring tool, enabling you to collect and display graphical information about the status and trends of AR System requests and events. Using meters and charts, you can get a snapshot view of requests and events or view them over a period of time to gather historical data. BMC Remedy Flashboards has an alert service that can be set up to automatically alert you to potential problems—for example, when an important threshold is reached—through BMC Remedy Alert, email, or pager calls.

Figure 7-1: BMC Remedy Flashboards



Development tools

BMC Remedy Migrator

BMC Remedy Migrator provides a fast and easy way to move forms and applications between two or more AR System servers. This tool helps you transfer your data and workflow definitions from a development environment to a production server, while ensuring the integrity of all migrated changes.

BMC Remedy Application Explorer

BMC Remedy Application Explorer works with BMC Remedy Administrator to help you identify relationships among objects in an application and explore the activities of workflow, given specific conditions. You can use this tool to understand an application's logic and structure, assess the impact of making modifications to the application, and simplify application troubleshooting.

IT Service Management

IT Service Management offers a complete, integrated solution to technology lifecycle management. These applications compress business cycles with custom routing of approvals and consistent enforcement of business rules.

BMC Remedy Help Desk

The BMC Remedy Help Desk application integrates the functions of problem management, problem resolution, asset inventory, change tasking, measurement, and reporting. You can automate skill-based assignments so that submitted requests are routed to designated help desk personnel, matching the problem with the specific staff expertise required to solve it. The application's adaptable workflow capabilities, escalation features, complete logging of information, and flexible problem categorization provide for highly efficient case management.

Tracking assets and their components with accurate configuration information accelerates help desk problem diagnosis. Intelligent links to help desk cases or change requests provide a complete history of assets. Change tasking keeps change requests on track through automated scheduling of assets and people as well as automatic notification and escalation. BMC Remedy Help Desk also provides ways to easily link changes to help desk cases or assets across the enterprise.

BMC Remedy Asset Management

The BMC Remedy Asset Management application tracks and manages enterprise assets and their changing relationships throughout the entire asset life cycle. It consolidates all costs related to each asset—including contract, asset depreciation, and service costs—into a single, integrated view of your distributed computing environment.

You can integrate information from third-party tools, such as Intel LANDesk, Microsoft SMS, and Tally Systems' NetCensus, into the BMC Remedy Asset Management application. You can also view asset statistics and cost data in a variety of ways using Flashboards.

BMC Remedy Change Management

The BMC Remedy Change Management application enables you to analyze the impact, risks, and resource availability associated with changes, and then create plans and automate approval functions to implement and respond to those changes. Its sophisticated tools facilitate scheduling and task assignment as well as performance review and process improvement.

BMC Remedy Service Level Agreements

Service level agreements (SLAs) are contracts that enable organizations to better manage service goals and commitments. The BMC Remedy Service Level Agreements application enables organizations to create these SLAs, incorporate them into existing business workflow, and then automate SLA tracking using notifications, escalations, and reassignments.

Any SLA that you create can be used with AR System applications, including BMC Remedy Help Desk, BMC Remedy Asset Management, or any third-party application.

Customer Service and Support

BMC Remedy Customer Service and Support Solutions can be implemented individually or as an integrated suite. Integration enhances collaboration among your sales, marketing, support, engineering, and quality assurance departments because they can share important and up-to-the-minute information.

BMC Remedy Customer Support

The BMC Remedy Customer Support application streamlines your customer support processes by logging issues, problems, suggestions, and requests for information and tracking them through to resolution—continually updating your customer database to reflect up-to-the-minute status. It automatically escalates unresolved problems and sends notifications to the appropriate people.

BMC Remedy Quality Management

The BMC Remedy Quality Management application improves your product development cycle by adding critical customer input directly into the development process. It automates your quality assurance tracking functions and provides the link that connects quality assurance, customer support, and engineering to your customers.

Integration with third-party products

A basic design philosophy of AR System is that it will frequently be used in conjunction with other tools and products to create an integrated solution. Some popular areas in which third parties have integrated their products include:

- Web services
- Network and system management
- Computer telephony, including automated call distribution (ACD) and integrated voice response (IVR)
- Asset/inventory management
- Groupware
- Legacy management
- Report writers
- Remote access
- Fax/pager/email
- Knowledgebases
- Accessibility for disabled AR System users

AR System is an *open system* with several well-defined interfaces for linking to other products. Third parties can use any of the following methods or "hooks" to integrate their products with AR System. For an in-depth discussion about integrating with third-party products, see the *Integrating with Plug-ins and Third-Party Products* guide.

Integration Method	Description
Accessibility for AR System Users with Disabilities	AR System 7.0 is compliant with Section 508 of the Rehabilitation Act of 1973. Web clients displaying AR System applications that are integrated with assistive technology like JAWS (Job Access with Speech) are accessible to people with disabilities.
Application Programming Interface (API)	The AR System API on the server is the most technical of the methods. However, it is very powerful and provides access to all AR System server functionality. Using the API creates tightly linked, high-performance integrations. The API is available for both the "C" and Java environments. The Java API classes and methods allow developers to quickly build enhanced applications to enable forms to be viewed on the web.
AR System Database Connectivity (ARDBC)	AR System Database Connectivity allows you to access and manipulate data that is not stored in AR System. Using the ARDBC API, you can create plug-ins that are used by AR System server to manage data. These plug-ins are loaded at run-time and implement calls that are analogous to the set, get, create, delete, and get-list API calls for entries in a form.
AR System External Authentication (AREA)	External authentication gives you a way to connect to a data source outside AR System, such as LDAP or Kerberos, to validate users.
Command Line Interface (CLI)	A command-line interface is available on most of the AR System client tools. It allows the tools to be launched and controlled by a third-party application or process.
Dynamic Data Exchange (DDE)	BMC Remedy User supports DDE. It allows AR System to pass data to other Windows applications and to be started "in context" by other applications.

Integration Method	Description
Email Messaging	AR System 7.0 Email Engine provides the following functionality: ■ Outbound email can be used to send email messages. Includes support for formatted text, customizable email header fields, priority, HTML templates, and MIME attachments. ■ Inbound email includes creating new requests in AR System and retrieving data or multiple records for the server. Inbound email can be parsed and can execute specific actions. ■ Processes notifications about AR System requests (such as the status of the request). ■ Supports multiple protocols including Messaging Application Programming Interface (MAPI), Internet Message Access Protocol (IMAP), Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and mBox. Also supports Multipurpose Internet Mail Exchange (MIME) types for attachments.
Enterprise Integration Engine (EIE)	A set of AR System forms (the Data Exchange application) can be used to define how to transfer data between AR System and another application, such as an Enterprise Resource Planning (ERP) system.
Extensible Markup Language (XML)	AR System can export and import object definitions in XML. AR System clients can convert AR System objects to XML and vice versa without making calls to AR System server. When the server exports the file in XML format, it adds a header required to make it a valid XML document. This same header is required for the server to correctly import an XML file, otherwise the file is assumed to be in the standard AR System definition format.
Filter Application Programming Interface (Filter API)	The filter API allows you to use filters to permit other applications to make calls back into AR System.
Hypertext Markup Language (HTML)	AR System can import and export form definitions in HTML. Web pages can be imported to AR System for use as templates by specifying the Uniform Resource Locator (URL) for the page. A form view created with AR System web editor in BMC Remedy Administrator can be saved in HTML format. The resulting HTML file can be edited using a simple text editor or industry-standard third-party HTML tools such as Microsoft FrontPage and Macromedia Dreamweaver.

Integration Method	Description
Object Linking and Embedding (OLE)	BMC Remedy User also supports OLE, a method for linking to other applications on the Windows platform. DDE and OLE provide similar functionality, but programs that support OLE usually have more advanced capabilities.
Open Database Connectivity (ODBC)	ODBC (Open Database Connectivity) is an SQL-based communication standard developed by Microsoft that enables the AR System to communicate with ODBC clients. BMC provides the BMC Remedy ODBC driver, which acts as a gateway to access data defined in AR System forms. Seagate Crystal Reports is an ODBC client that has full functionality with the AR System. Crystal Reports enables you to create custom reports with wide-ranging capabilities and provides additional flexibility in report design.
Running External Processes	The Run Process workflow action in AR System can be used to start other applications.
Simple Network Management Protocol (SNMP) Monitor and Management Interface	The SNMP interface enables system administrators and IT personnel to use SNMP-compliant management consoles (such as BMC Patrol) to monitor AR System statistics and state. In addition, users can stop or start AR System using the BMC Remedy SNMP agent.
SQL Database Access	Because the AR System database is fully open and documented, third-party applications can read data from the AR System database. AR System can also read and write to third-party databases.
Vendor Forms	Vendor forms allow direct read and write access to data located in database tables and external data sources, such as text files and spreadsheets, which are not owned by AR System. This allows direct access to this data without database replication; however, some programming is required to link to the external data source.

Integration Method	Description
View Forms	View forms allow direct read and write access to data located in database tables that are not owned by AR System. This allows direct access to these tables, as if they were owned within AR System, without programming, database replication, or synchonization.
Web Services	Integration technology (XML, WSDL, UDDI, and SOAP) that easily allows you to build B2B or A2A distributed applications without programming. You can now perform the following tasks: ■ Use the Set Fields workflow action and a Web Services object to “consume” third-party web services in AR System applications. ■ Use AR System to create and “publish” a Web Services object.

Chapter 8 Putting it all together

In the previous chapters, you have learned about the important concepts of AR System—forms, menus, workflow components, and integrations with other products. Now that you have the pieces of the puzzle, you need to understand how to put them together to form a useful whole.

This chapter brings together those concepts by showing how a sample organization, in this case a wild animal park, might approach planning and designing an AR System application. Like any business, the park needs to take a systematic approach to automating its processes. This chapter walks you through the planning process that the park staff uses to evaluate and address its business needs.

Overview

The wild animal park has been growing successfully with the tried-and-true methods of paper-based record keeping combined with isolated computer-based procedures. Now, however, a number of redundant or inefficient processes have been noticed by employees, and the park has decided to use AR System to improve these processes. The processes that they would like to automate using AR System include:

- Tracking and managing animals associated with the park.
- Tracking and managing public relations aspects of the park, such as attendance statistics, public inquires, membership renewals, and group tour information.
- Tracking and managing the maintenance of on-site visitor facilities, including snack bars, rest rooms, first-aid stations, and park transit systems.
- Managing the botanical gardens adjacent to the park.

For the purpose of this chapter, we will focus on the first application, managing and tracking the animals associated with the park.

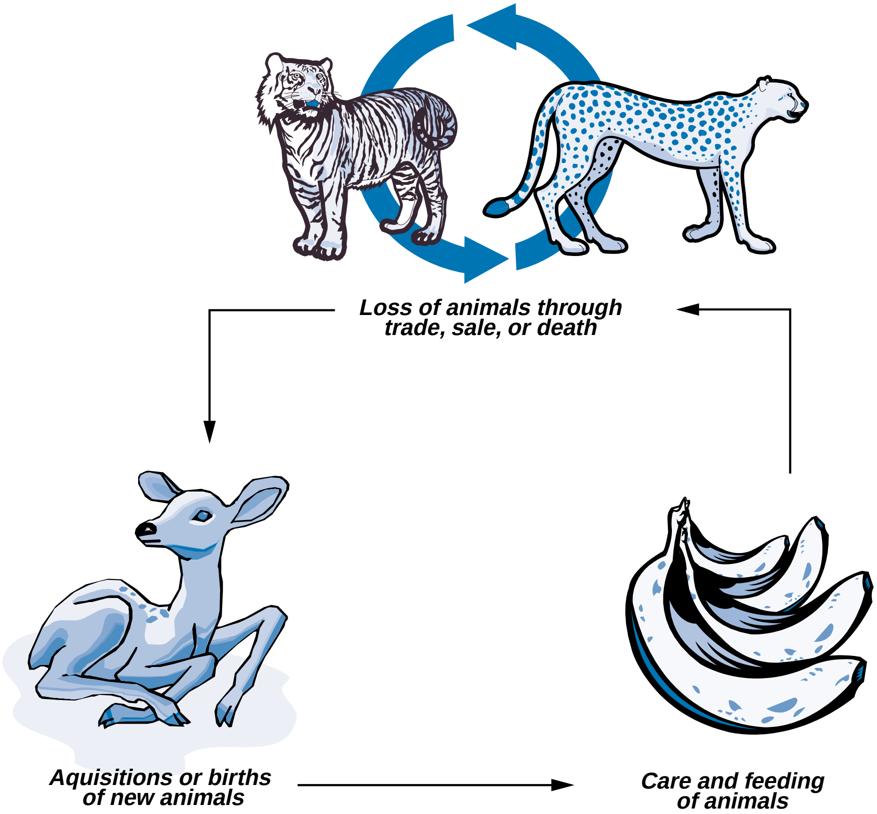
High-level view of an animal tracking application

The park needs to accomplish the following goals with the animal tracking application:

- Track the type and number of animals grouped together in enclosures.
- Track births, deaths, acquisitions, trades, and sales.
- Track daily observations of each animal, including behavior and medical condition.
- Track the complete medical history of each animal, including preventive care (such as dental work, vaccinations, and parasite checks).
- Track animal feeding.
- Immediately alert the veterinary staff of medical emergencies.
- Alert the animal handlers if any animals escape their enclosures.

All of these goals relate to keeping track of animals throughout their entire life at the park, as indicated in the following figure.

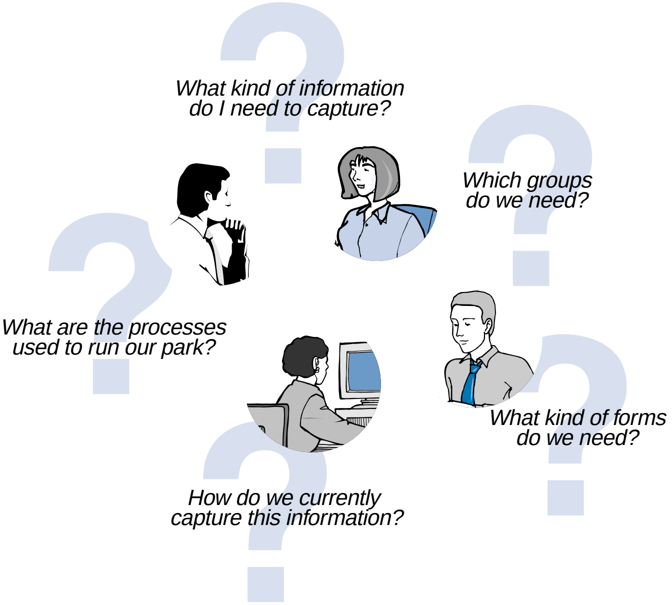
Figure 8-1: Animal tracking overview


Loss of animals through trade, sale, or death
Aquisitions or births of new animals
Care and feeding of animals

Planning and design questions

After defining its application goals, the staff begins more detailed planning. The following sections describe some of the points that any organization needs to consider if it wants to create a useful application. (The planning and design process is thoroughly addressed in the “AR System Requirements Analysis, Design, and Development” course offered by BMC.) This section suggests various questions the staff asks, as shown in the following figure. The section called “Planning and design decisions” on page 132 describes the answers to these questions.

Figure 8-2: Questions for planning the animal tracking application



Data analysis

As the park staff members begin to plan their animal management application, they think about the type of data they need to capture. They also ask themselves how this data is captured in their current system (for example, in a legacy database or by paper forms).

After determining the kind of data they need to capture and how the data interrelates, the park can determine what forms (main and supporting) and fields need to be created. They also need to think about whether they want to include menus on the forms and, if so, which kind are most appropriate to help users fill in fields.

Workflow/process analysis

Next, the staff considers the organizational processes that are currently in place at the park.

- What are these processes?
- What are the various stages or steps associated with each process?
- Who are the specific groups of people involved in these processes?
- What specific information do these groups need to manage, access, or track?

Defining business rules

After examining their business processes, the staff members also consider their business rules, which are the fundamental policies that govern day-to-day life at the park. The rules frequently provide the basis for making important decisions. For example, one of the rules might be that every animal must be checked by a veterinarian within 24 hours of arrival. If the rule is broken, that might indicate a need to hire more medical staff or to increase clinic capacity.

The questions about business rules include:

- What should happen when various conditions occur?
- What is the flow of information through the existing systems?

Mapping business rules to workflow components

Next, the park needs to determine how to translate its business workflow (rules and processes) to AR System workflow components:

- Which processes can be accomplished using active links?
- When would it make more sense to use a filter?
- What kinds of escalations are needed to enforce business rules? For example, an escalation might be used to enforce the rule that animals must be checked within 24 hours of arrival.

What are the integration considerations?

The staff needs to consider what other products or databases must be integrated into the application initially and what future integrations are desirable:

- It must be possible for park staff to enter data while they are out in the park, perhaps using hand-held devices.
- Future integration with a sister zoo must be possible.
- They want to integrate with an internationally maintained database on endangered species, partly to get new individuals contributing to the gene pool at the park.
- They eventually might want to integrate information about the botanical gardens at the park (although this could be a different application entirely).

Planning and design decisions

Based on their research, the park staff makes the following decisions about the types of forms they need to capture data, the groups of people who need to use AR System, and the workflow components they would put in place to automate their processes.

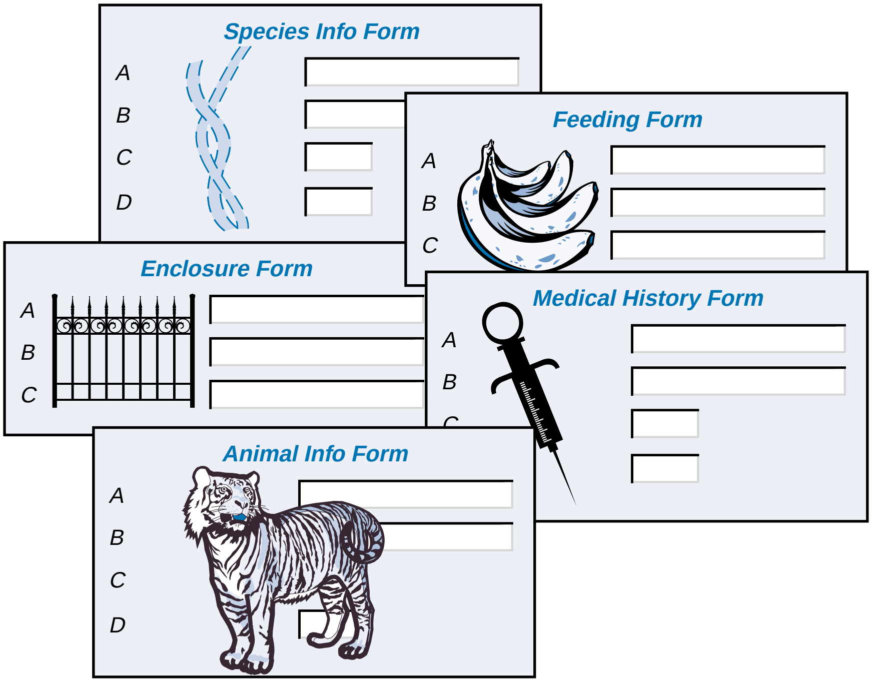
Decisions about forms

The park staff determines that some of the forms they need to capture information include:

- An Animal form, which contains detailed information about each animal. (They consider using page fields to divide the form up into manageable chunks.)
- A Species Information form, which contains details about a particular species such as feeding requirements, life span, medical needs, and whether it is endangered. (This would be a supporting form.)
- A Feeding form, which contains information about the feeding schedule for each animal.

- An Enclosure form, which includes information about the number of animals each enclosure can hold, the types of animals, and so forth.
- A Medical History record form, which contains a complete medical history of each animal.

Figure 8-3: Forms for animal tracking application



Decisions about access control

Some of the AR System groups or roles the park needs include:

- Veterinarians
- Keepers (those who care for the animals)
- Curators
- Horticulturists (who maintain the animals' naturalistic habitats)
- Researchers

Appropriate permissions will be assigned to each group or role, based on the information they need to access.

Decisions about business rules

Some sample business rules for the park include:

- Animals will be not be kept in temporary enclosures longer than 48 hours.
- Specially trained animal handlers will be notified immediately if a dangerous animal escapes.
- Every animal must be checked by a veterinarian within 24 hours of arrival.

Decisions about workflow components

Some of the workflow components the park needs include:

- A filter to notify the animal handlers whenever an animal needs to be moved.
- Active links that make filling out requests easier.
- An escalation to notify keepers if animals have not been fed within one hour of their scheduled feeding time.

Scenarios

After the planning and design process, the park came up with an application that met its diverse needs. The following three scenarios are examples of the application in use. Each scenario contains a description and illustration along with "Development Notes" that describe some of the changes the developers made to the application based on feedback from users. In addition, each scenario includes a reference to similar processes in various business applications. Some of the problems the wild animal park staff faces are not unlike the ones you face.

Each scenario illustrates a complete process, but within each discussion not every component is discussed in detail. For example, enough detail might be given about how one workflow component functions to enable you to make assumptions about other components.

Scenario 1: A new tiger is acquired

As shown in "Scenario 2: Tiger is injured" on page 136, when a new Sumatran tiger named Karuna is obtained, a staffer fills in the Animal form, and presses a button called "List Enclosures." An active link opens a dialog box displaying the Enclosure form with a table field showing enclosure information, including the availability and habitat. (The user could double-click the enclosures to get more information.) Next, he selects an appropriate choice—in this case, enclosure 16—and creates a request. A filter notifies the Animal Handlers group and sends a message to the staffer who created the request that the appropriate people have been informed. Also, the Status field changes from New to Move Pending.

Gary from Animal Handlers receives email that a new tiger must be moved from the temporary cages to enclosure 16.

After the tiger is transferred, Gary changes the Status field from Move Pending to Permanent. After he modifies the request, workflow components create new requests in related forms and notify the veterinarian group and the animal handlers group to begin the care and feeding of the new animal. These requests and notifications are one way of handling work orders in the AR System.

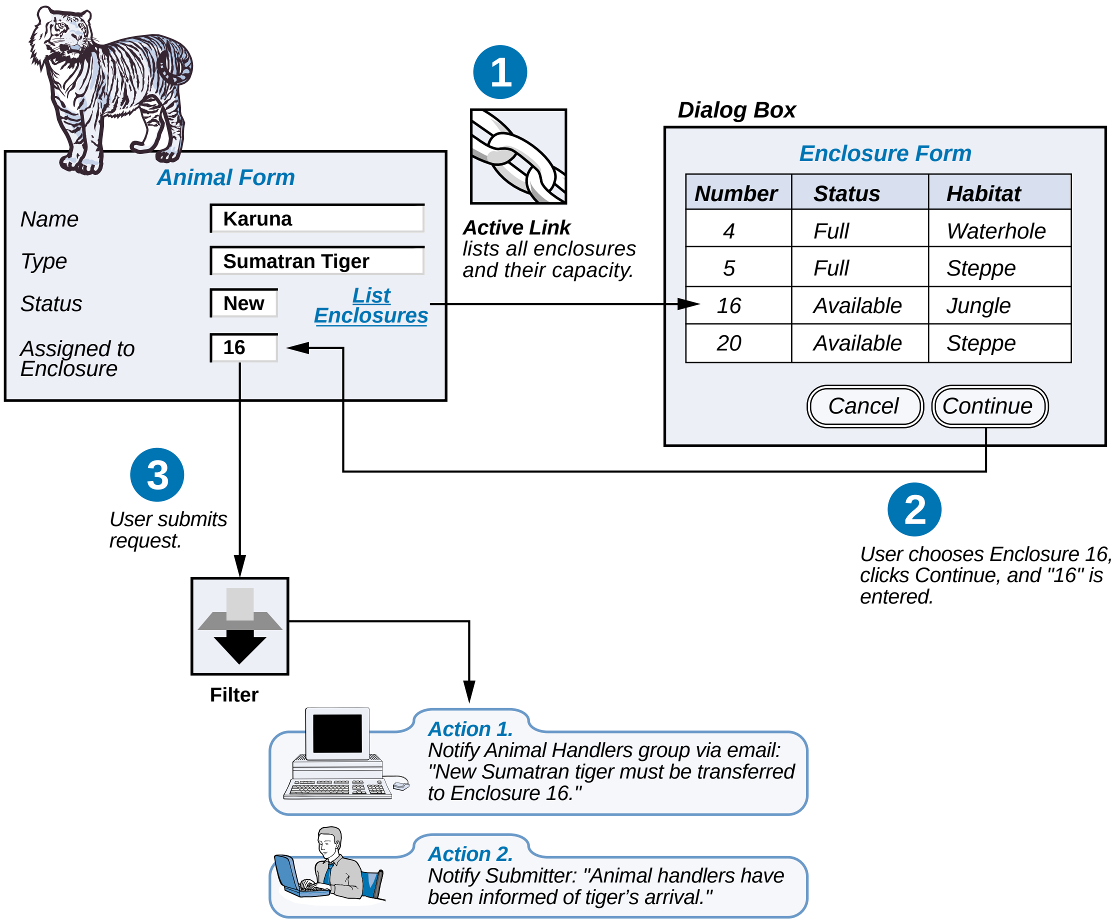
Similar process, different application

This process is similar to *moves, adds, and changes (MAC)* in an Employee Services application.

Development notes: During trial runs of the system, the application developer realizes that the animal handlers are frequently away from their computers and rarely check email. The developer integrates the application with a pager, and has the filter notify the handlers of new animals with a page. Handlers can then use their cell phones to get information about their assigned tasks.

Scenario 2: Tiger is injured

Figure 8-4: Example scenario



Name	Karuna
Type	Sumatran Tiger
Status	New
Assigned to Enclosure	16

Active Link lists all enclosures and their capacity.

Number	Status	Habitat
4	Full	Waterhole
5	Full	Steppe
16	Available	Jungle
20	Available	Steppe

User chooses Enclosure 16, clicks Continue, and "16" is entered.

User submits request.

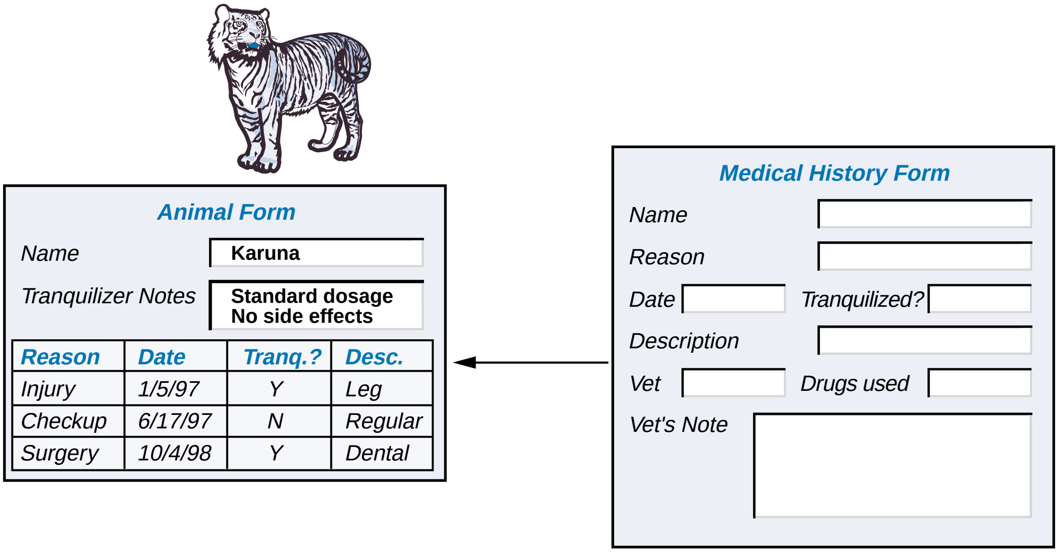
Action 1. Notify Animal Handlers group via email: "New Sumatran tiger must be transferred to Enclosure 16."

Action 2. Notify Submitter: "Animal handlers have been informed of tiger's arrival."

One morning when the keepers are doing their daily rounds, they notice that Karuna has been injured. This is reported immediately to the veterinarians. The veterinarian looks at the Animal form and checks a table field that contains data from the Medical History form. She discovers that Karuna has no history of serious injury or illness.

For the injury reported, Karuna must be tranquilized and moved to the veterinarian hospital for surgery. He has been tranquilized before without problem as indicated by the Tranquilizer Notes field on the Animal form, so the veterinarian computes the dosage and sets out with several keepers to bring in the tiger.

Figure 8-5: Table in the animal tracking application



Animal Form

Name: Karuna

Tranquilizer Notes: Standard dosage
No side effects

Reason	Date	Tranq.?	Desc.
Injury	1/5/97	Y	Leg
Checkup	6/17/97	N	Regular
Surgery	10/4/98	Y	Dental

Medical History Form

Name

Reason

Date

Tranquilized?

Description

Vet

Drugs used

Vet's Note

Development notes:

The Tranquilizer Notes field on the Animal form did not always exist. During the prototyping phase, the user would have had to open the Medical History form separately to learn about Karuna's record with tranquilizers. However, the veterinary staff pointed out that they would want that important information readily available during an emergency. So, the Tranquilizer Notes field was added to the Animal form and a filter that executes on Submit was added to post a message to the veterinarians reminding them to update the Tranquilizer Notes if necessary.

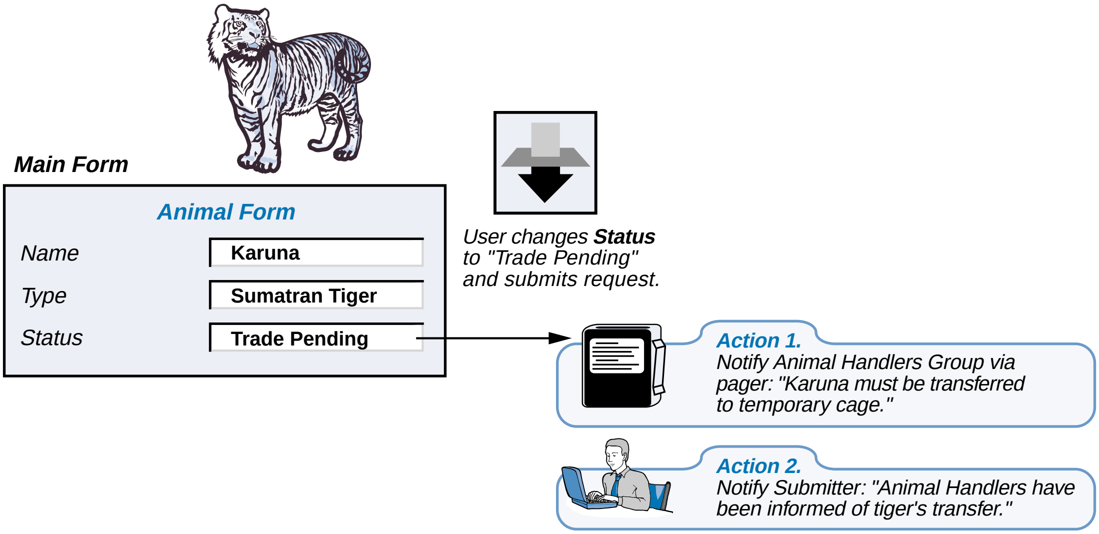
Similar process, different application

This scenario is similar to handling a customer call in a technical support application. The technical support representatives could decide that they need important information about a customer available on a main form rather than on a supporting form.

Scenario 3: Tiger traded to another zoo

After several years, the animal park determines that it should have a different male tiger to maintain genetic diversity in its tiger population. By examining a database maintained by zoos worldwide, the staff discovers a tiger that is available and has no common ancestors with Karuna or with the park's female tigers. After the decision is made to trade Karuna, a staffer changes Karuna's status from Permanent to Trade Pending, thereby triggering the same notification filter that was used in the first scenario, this time to notify the animal handlers to move Karuna to a temporary cage.

Figure 8-6: Notifications in the animal tracking application



Name	Karuna
Type	Sumatran Tiger
Status	Trade Pending

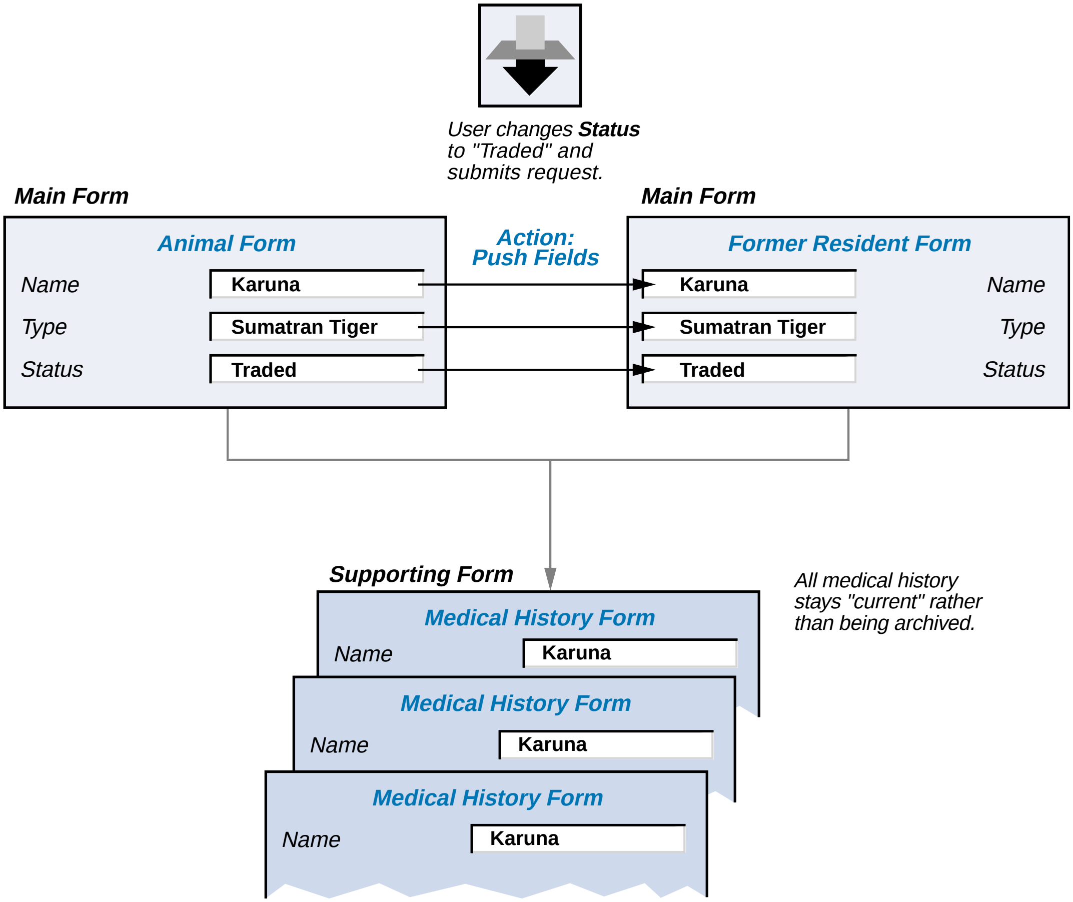
User changes **Status** to "Trade Pending" and submits request.

Action 1. Notify Animal Handlers Group via pager: "Karuna must be transferred to temporary cage."

Action 2. Notify Submitter: "Animal Handlers have been informed of tiger's transfer."

After Karuna has actually left the park, his status is changed to Traded. When the changed request is submitted, a filter moves all of Karuna's data from the Animal form to the Former Resident form using a Push Fields action.

Figure 8-7: Push Fields action used in the animal tracking application



Development notes:

The Medical History form does not get archived or changed because the staff might, at any point, want information from the medical records. For example, they might want to find out about all tiger surgeries performed at the park.

Similar process, different application

This scenario is similar to retiring an asset in an Asset Management application. One commonality is that you need to track the problem history of an asset both during its active use and after its retirement.

As you can see, AR System applications can be created to track any kind of asset allocation or business process. With sufficient planning, even workflow intensive procedures can be automatically maintained in an orderly manner.

Appendix

A For more information

This appendix points you to additional sources of information about BMC and AR System. Areas of discussion include the BMC Remedy website, a list of relevant publications and resources from BMC, and information about electronic forums, BMC Remedy user groups, training courses, customer support, and consulting services.

BMC Remedy website

The BMC Remedy website contains a wealth of detailed information about Remedy and its products and services. In addition, you will find software demos and applications that you can download directly from the site.

You can visit the BMC Remedy website at `http://www.remedy.com`.

AR System product documentation

The following table lists the documentation available for AR System products.

Unless otherwise noted, online documentation in Adobe Acrobat (PDF) format is available on AR System product installation CDs, on the Customer Support website (`http://supportweb.remedy.com`), or both.

You can access product Help through each product's Help menu or by clicking on Help links.

AR System documents

Title	Description	Audience
Concepts	Overview of AR System architecture and features with in-depth examples; includes information about other AR System products as well as a comprehensive glossary for the entire AR System documentation set.	Everyone
Installing	Procedures for installing AR System.	Administrators
Getting Started	Introduces topics that are usually only learned when first starting to use the system, including logging in, searching for objects, and so on.	Everyone
Form and Application Objects	Describes components necessary to build applications in AR System, including applications, fields, forms, and views.	Developers
Workflow Objects	Contains all of the workflow information.	Developers
Configuring	Contains information about configuring AR System servers and clients, localizing, importing and exporting data, and archiving data.	Administrators

Title	Description	Audience
Installing and Administering BMC Remedy Mid Tier	Contains information about the mid tier, including mid tier installation and configuration, and web server configuration.	Administrators
Integrating with Plug-ins and Third-Party Products	Discusses integrating AR System with external systems using plug-ins and other products, including LDAP, OLE, and ARDBC.	Administrators and Developers
Optimizing and Troubleshooting	Server administration topics and technical essays related to monitoring and maintaining AR System for the purpose of optimizing performance and troubleshooting problems.	Administrators
Database Reference	Database administration topics and rules related to how AR System interacts with specific databases; includes an overview of the data dictionary tables.	Administrators
Administering BMC Remedy DSO	Server administration and procedures for implementing a distributed AR System server environment with the BMC Remedy Distributed Server Option (DSO).	Administrators
Administering BMC Remedy Flashboards	Flashboards administration and procedures for creating and modifying flashboards and flashboards components to display and monitor AR System information.	Administrators and Programmers
C API Reference	Information about AR System data structures, C API function calls, and OLE support.	Administrators and Programmers
C API Quick Reference	Quick reference to C API function calls.	Administrators and Programmers
Java API	Information about Java classes, methods, and variables that integrate with AR System.[a]	Administrators and Programmers
Administering BMC Remedy Email Engine	Procedures for installing, configuring, and using the BMC Remedy Email Engine.	Administrators
Error Messages	List and expanded descriptions of AR System error messages.	Administrators and Programmers
Master Index	Combined index of all books.	Everyone

Title	Description	Audience
Release Notes	Information about new features list, compatibility lists, international issues, and open and fixed issues.	Everyone
BMC Remedy User Help	Procedures for using BMC Remedy User.	Everyone
BMC Remedy Import Help	Procedures for using BMC Remedy Import.	Administrators
BMC Remedy Administrator Help	Procedures for creating and modifying an AR System application for tracking data and processes.	Administrators
BMC Remedy Alert Help	Procedures for using BMC Remedy Alert.	Everyone
BMC Remedy Mid Tier Configuration Tool Help	Procedures for configuring the BMC Remedy Mid Tier.	Administrators

[a] A JAR file containing the Java API documentation is installed with the AR System server. Typically, it is stored in `C:\ProgramFiles\AR System\Arserver\Api\doc\ardoc70.jar`.

Other BMC documentation

- A wide variety of technical and troubleshooting documents are available to customers through the BMC Remedy Customer Support website at `http://supportweb.remedy.com/`. You will need a support ID and password.
- Integration notes list products that have been integrated with AR System. These are available through the BMC Remedy website (Partner page) at `http://www.remedy.com/ppp/integration.htm`.

Sample application

You can optionally install the Sample application with the AR System 7.0 server to see examples of various kinds of forms, fields, and workflow actions. The application implements a class-registration system and includes forms to add classes, register for a class, and view enrollments. Forms in the sample application are enabled for both BMC Remedy User and web client access. Help text in the forms and workflow explains how the application works.

In BMC Remedy User, log in as Demo on a server where the Sample application is installed, and explore the Sample:Class Central form. In BMC Remedy Administrator, open the Sample application and explore the objects listed in the application window.

On the web, enter the home page URL (`http://<server_name>/arsys/home`). Log in as Demo with no password. Links for the Sample application appear in the left pane. For example, click Enroll in Classes to explore the functionality of the sample application.

ARSList electronic communication

Exchanging electronic messages with other AR System administrators and users can be an excellent way to solve problems and learn about how other people are using BMC products. The most popular email forum for BMC products is *ARSList*; for more information, visit the AR System Developer Community at `http://www.remedy.com/customers/dev_community`.

To subscribe to ARSlist, use the link on the Developer Community website.

BMC Remedy user groups

BMC Remedy user groups provide an open and dynamic forum for exchanging ideas and strategies on a wide variety of topics related to AR System products. There are a number of established user groups around the world. You can also start your own user group by joining the BMC Remedy Local User Group Program.

The BMC Remedy User Group (RUG) is an annual meeting of customers and partners that offers a chance for customers to mingle with each other and BMC Remedy employees and discuss how they are implementing solutions based on BMC Remedy technology. RUG also allows BMC to relate the latest company directions and strategies.

For information about the BMC Remedy user groups nearest you, about becoming a member of the BMC Remedy User Group (RUG), or about starting your own group, visit the website at `http://www.remedy.com/customers/user_groups.htm`.

BMC Remedy developer community

Interested in the latest AR System product information, enhancing your development skills, collaborating with peers? The AR System Developer Community has been created to support your development needs. You can collaborate with peers on the Message Board, download and submit applications and utilities to the Applications Gallery, check out the Community-recommended links from your peers, product updates, documentation and downloads, patches, the ISV program and help the shape the future of the Developer Community by using the Community Suggestion Box. Visit the AR System Developer Community at http://www.remedy.com/customers/dev_community.

Training

BMC provides customers with a comprehensive range of classes that span such diverse topics as designing an AR System application, using BMC Remedy User, administering AR System, performance tuning and troubleshooting, and API programming.

For more information about training provided by BMC and by BMC Remedy Alliance Partners around the world:

- Visit the BMC Remedy website at http://www.remedy.com and navigate to the Education page.
- Contact your local BMC Remedy sales representative.

Customer support

BMC provides a variety of Customer Support Services for all BMC Remedy products including maintenance and updates, telephone-based support from the BMC Remedy Response Center, and problem resolution.

Your BMC account team helps you identify the correct combination of support services you need. They will also provide you with information about how and when to contact technical support and response centers according to your needs.

The BMC Remedy Response Center serves as the focal point for all of your support needs. The Response Center answers your technical questions and performs an initial diagnosis when you experience a problem. When necessary, the Response Center enlists additional technical resources to help you solve your problem.

For more information about BMC Remedy Customer Support:

- Visit the BMC Remedy website at `http://www.remedy.com` and navigate to the Support page.
- Send an email to `support@remedy.com`.

Consulting services

The Consulting Services organization consists exclusively of systems professionals with extensive AR System experience to help you develop solutions in workflow automation and application design.

For more information about Consulting Services:

- Visit the BMC Remedy website at `http://www.remedy.com` and navigate to the consulting page.
- Contact your local BMC Remedy sales representative.

Master Glossary

access control

A security feature in which AR System administrators limit the access users have to forms, to specific fields or records within a form, to workflow, and to specific functions within the system. See also *group*, *permission*, *role*, *user*.

access point

A form or guide in an application that is designated for use as an interface by other applications, such as in Push Fields or Call Guide actions. When creating workflow that references forms and guides, a developer can use the Restricted List check box to limit the list of available forms and guides to only those in the current application and those designated as access points in other applications.

action request

A collection of information that describes something, such as a problem or a service request.

active link

A workflow component that causes an AR System client to perform specific operations in response to specific user actions. They are generally used to assist users with interactions with the system. Active links run on the client machine.

active link guide

An ordered sequence of active links that together accomplish a specific operation. Active link guides can help lead users through a task (like a wizard) or can be used as subroutines to accomplish common tasks. Compare with *filter guide*.

administrator

An individual responsible for the management of the AR System including setting up forms, setting access rights for users, and designing the workflow process.

administrator default

The value that AR System administrators assign to a field while designing a form. When users set defaults, this value is automatically entered in the field. Users can override the administrator default by assigning their own default or by entering a different value. See also *user default*.

Administrator group

One of several special access control groups that the AR System provides. Members of this group have full and unlimited access to the AR System including unlimited ability to create definitions and access and modify data. See also *explicit group*, *Subadministrator group*.

advanced search bar

The row of buttons, the Search Criteria field, and the Fields menu list that appear at the bottom of the Detail pane when you click the Advanced button in BMC Remedy User. You can use this bar to specify complex search criteria.

alert

A notification from an AR System server or other program to the user that certain conditions have occurred, such as a request has been submitted or progress has been made in resolving a request.

alert list

The list of alerts belonging to a user that can be displayed in BMC Remedy User or on a web client.

allowable currency

A currency type that appears in the currency field menu. Users can only use an allowable currency type when entering currency values. See also *currency data type*, *functional currency*.

API

See *application program interface (API)*.

application

A group of forms and the associated workflow collected by an administrator that are related to a business function; for example, Employee Care or Help Desk. An application is also a server object in BMC Remedy Administrator. See also *deployable application*, *local application*.

application list field

A field that displays entry points. See also *entry point*, *entry point guide*, *form entry point*, *home page*.

application state

The development state of a deployable application, such as Test or Production. You can map roles to different groups depending on the application state to limit access to the application during testing or modification. See also *deployable application*, *group*, *role*.

application program interface (API)

A set of functions that provides application programmers with access to the full functionality of a product. The AR System C API is documented in the *C API Reference*, while the Java API is documented in the Java API HTML documentation.

ARDBC

See *AR System database connectivity (ARDBC)*.

AREA

See *AR System external authentication (AREA)*.

AR System

See BMC® Remedy® Action Request System®.

AR System client

The subset of AR System software that enables users to access an AR System server. The AR System client tools are BMC Remedy Administrator, BMC Remedy User, BMC Remedy Import, and BMC Remedy Alert.

AR System database connectivity (ARDBC)

A mechanism by which the AR System server uses a plug-in to access data stored externally to the AR System database as if it were within the AR System.

AR System external authentication (AREA)

A mechanism by which the AR System server can access and use authentication services from outside the AR System environment. A plug-in is developed to allow access to the external subsystem.

AR System Mail Server

An AR System client tool that is used to submit and search using email. It also enables you to send mail notifications.

AR System ODBC driver

A connectivity solution that enables the AR System to communicate with ODBC clients. ODBC, or Open Database Connectivity, is an SQL-based communication standard developed by Microsoft.

AR System server

The subset of AR System software that provides the data storage and main processing environment for the AR System.

Assignee group

One of several special access control groups that the AR System provides. Users automatically belong to this implicit group for requests for which they have been assigned responsibility (that is, their name is in the Assigned To field). See also *Assignee Group group*, *dynamic group*, *implicit group*, *Submitter group*.

Assignee Group group

One of several special access control groups that the AR System provides. Users automatically belong to this implicit group for requests for which they are a member of a group assigned responsibility (that is, they are a member of a group whose name is in the Assignee Group field). A single user can also belong to this group when that user's name is in the Assignee Group field. See also *Assignee group*, *dynamic group*, *implicit group*, *Submitter group*.

attachment data type

The data type used for fields that need to hold files. This type enables you to store text, graphics, audio, or video attachments in the database.

attachment pool field

A field that contains a set of one or more related attachment fields to organize attachments associated with an *action request*.

BMC® Remedy® Action Request System® (AR System)

Adaptable client/server software that provides a foundation for creating applications that can automate, track, and manage a wide variety of business processes.

BMC Remedy Administrator

The AR System client tool used by AR System administrators and subadministrators to set up the system for users. Use BMC Remedy Administrator to create and change structure definitions. BMC Remedy Administrator is also used to set access permissions that determine which users and groups can view or modify each form or specific parts of a form.

BMC Remedy Administrator Analyzer

A tool within BMC Remedy Administrator that analyzes forms for conditions that can lead to performance or maintenance issues within the definition of the form.

BMC Remedy Alert

The AR System client tool through which an alert can be sent to a user. See also *notification*.

BMC Remedy Application Explorer

A tool that works with BMC Remedy Administrator to help you identify relationships among objects in an application and explore the activities of workflow, given specific conditions.

BMC Remedy Approval Server

A module within the AR System that routes forms to generate the appropriate signoff signatures. Remedy Approval Server also creates an audit trail for authorizing AR System application forms.

BMC Remedy Mid Tier Configuration Tool

The tool used to configure and manage the mid tier portion of the AR System.

BMC Remedy Email Engine

A server-based application that communicates with both the AR System server and an email server. The Remedy Email Engine receives emails, and can parse and interpret these messages to execute specific instructions on an AR System form. It also sends emails to the AR System and directs notifications as a result of filters and escalations.

BMC Remedy Flashboards

A real-time visual monitoring tool that shows you the state of your service operations, warns you about potential problems, and collects and displays trend data.

BMC Remedy Import

The AR System client tool that enables AR System administrators to transfer data records from an archive file into a form.

BMC Remedy User

The AR System client tool in which users enter and track requests through the resolution process. Users can also search the database, generate reports, and modify existing requests.

broadcast operation

A distributed operation in which one source server simultaneously transfers data-only copies of entries to two or more target servers.

button field

A field on a form that a user can click to execute an active link. A button, image, or hyperlink can represent a button field. See also *menu item*, *toolbar button*.

chained operation

A distributed request operation in which one source server transfers a request to a target server, and the target server in turn transfers the request to another server. This operation is used in environments containing three or more servers.

change flag

A status flag set when the contents of a field or form have been altered. A change flag can be polled or disabled by workflow.

character data type

The data type used for fields that contain alphanumeric text.

character menu

A menu that the AR System administrator can create to provide assistance with filling in values for a field. You can attach a menu to any character field. See also *dynamic menu*.

client

See *AR System client*.

client tier

The architecture level where AR System clients operate within the multitier system.

command-line option

A parameter used to specify an operation or option to AR System tools when they are run.

computed group

An explicit group whose membership is based on the memberships of other explicit groups. See also *explicit group*, *group*.

container

The underlying data structure for guides and applications. A component of the AR System used to store collections of objects. Used as the basic storage structure for applications, active link guides, filter guides, and packing lists.

control data type

The data type for fields that execute active links. These fields do not contain data.

control panel

A form used as a centralized entry point from which users choose the business tasks they want to accomplish.

core field

One of a set of basic fields common to all AR System regular forms. AR System administrators cannot remove these fields from a regular form.

currency code

The three-letter code that represents a currency type, such as `USD` for United States Dollars.

currency data type

The data type used for fields containing currency data. Currency data is stored in four parts: a decimal value, a currency code, a conversion date, and one or more functional (converted) currency values. Each part can be viewed or searched by users, or accessed by workflow. See also *allowable currency*, *functional currency*, *currency code*.

Customize group

One of several special access control groups that the AR System provides. This group grants users the right to customize their form layout in BMC Remedy User. See also *explicit group*.

data field

A field that stores data in the database. Data fields include character, date/time, date, time, diary, integer, real, decimal, selection, attachment, and currency.

data tier

The architecture level that contains data and communicates with the AR System server. The data can be stored in a text file, spreadsheet, or database internal or external to the AR System.

data type

The property of a field that determines the characteristics of the field and what type of information (if any) the field contains.

data visualization field

A field that provides an efficient way to add graphical fields such as Flashboards to AR System forms.

data-only copy

In a distributed environment, a read-only copy of a request that is not part of an active ownership chain and cannot create a new ownership chain. See also *independent copy*, *ownership chain*, *distributed server*.

date data type

The data type used for fields containing date values. Date values are stored as the number of days from the beginning of the date field's range. Date values can range from January 1, 4713 B.C. to January 1, 9999 A.D. See also *date/time data type*.

date/time data type

The data type used for fields containing date/time timestamp values. Values can range from January 1, 1970 to January 1, 2038. See also *date data type*, *time data type*.

decimal data type

A field that accepts and contains fixed-point numbers. This type of field enables you to store quantitative information in a request.

default

See *administrator default*, *user default*.

default mapping

The mapping selected by the Distributed Server Option if the AR System finds multiple mappings that apply to the specified transfer criteria.

deployable application

An application that uses permissions based on roles that is easily portable to other servers. See also *application*, *local application*, *role*.

Details pane

The part of the main window in BMC Remedy User that displays fields for entering or viewing data.

dialog box

A message displayed to users that must be responded to before users can continue filling out a form. The administrator creates a dialog box by using an active link action.

diary data type

The data type used for fields that enables you to capture a history of the actions taken for a request. The field stores character data. It is an append-only field, and each addition is stamped with the time, date, and name of the user who entered the item.

display-only field

A temporary field for which no space is allocated in the database. See also *global field*.

display-only form

A type of form containing display-only fields. Display-only forms are used to create control panels and dialog boxes.

distributed delete

Used to remove a request from the AR System. In a distributed environment, a copy of a request can be deleted on one server if the master copy is deleted on another server.

distributed fields

Fields you can add to an AR System form that allow you to manipulate certain mapping controls. The amount and type of fields added to a form are determined by the mapping type selected.

distributed mappings

Objects on the DSO server that allow the user to define how data from one form is transferred to another. Users define how data is to be transferred by specifying the forms and fields used, how frequently updates are processed, the return values, if any, and how conflicts in distributed operations are resolved. Distributed mappings are used in conjunction with the DSO filter and escalation actions.

distributed operation actions

These operations control the action that is taken when filter or escalation criteria is met within a given form. The three distributed operation actions are distributed transfer, distributed return, and distributed delete.

distributed pools

Objects on a server that allow the inter-related operations of pending requests to occur in the correct order, especially during periods when the rate of incoming requests exceeds the processing rate.

distributed return

A distributed operation that returns the latest copy of a request, with ownership, to the requesting server.

distributed server

An AR System server that exists within a multiserver, distributed environment. See also *Distributed Server Option (DSO)*.

distributed transfer

Used to transmit information from one server to another. In a distributed environment, a request can be transferred on one server when it is created on or modified on another. The transfer can include the entire request or just specific data.

Distributed Server Option (DSO)

An AR System server option to send and receive data from both similar and dissimilar forms on physically separate servers. DSO requires a separate usage license. See also *distributed server*.

distributed update

A distributed operation that refreshes a copy of a request on one server when the master copy, on another server, is modified.

Distributed User

The name of the user performing all operations for the Distributed Server Option.

duplicate request IDs

A condition that can occur in a distributed operation when a request transferred from the source server has the same request ID as a request on the target server. The Distributed Server Option provides several options to handle this issue.

dynamic data exchange (DDE)

An interapplication communication feature used in Windows applications. For more information, see your Windows documentation.

dynamic group

One of several special access control groups that developers can create in the Group form using a group ID in the range of 60000 to 60999. Users automatically belong to this implicit group for requests when they are a member of a group whose name is in the dynamic group field (a field whose field ID is the same as the dynamic group ID). A single user or role can also belong to this group when that user or role name is in the dynamic group field. See also *Assignee Group group*, *group*, *implicit group*, *role*.

dynamic menu

A menu that causes a search to be performed when a user selects the menu button. The results of the search are used to build the list of items from which the user can choose. See also *character menu*.

email engine

See *BMC Remedy Email Engine*.

entry

A row in the database that represents a request.

entry point

A link on the home page that users click to start a task, such as creating a new request. See also *entry point guide*, *form entry point*, *home page*.

entry point guide

An entry point that starts a guide, so that a user can complete a task. See also *entry point*, *form entry point*, *home page.*

escalation

A workflow component that searches at specified times or at regular intervals for requests matching a specified condition and performs specified operations on all matching requests. They are generally used to find records that have exceeded desired business rules or processes and take appropriate action. Escalations run on the AR System server.

event

An occurrence in the AR System that can serve as a “trigger” for other events or workflow. Examples include user interactions with an AR System form (such as opening windows, tabbing from field to field, switching row focus, and so on), state transitions of requests, or any condition that arises while manipulating a request.

explicit group

A group to which users are assigned, such as Administrator or Customize. See also *group*, *implicit group*.

export

1. The BMC Remedy Administrator command that transfers object definitions in a file; for example, forms, filters, active links, and mail templates. 2. The ability to transfer one or more data entries to a file for archive or transfer by using the reporting feature in BMC Remedy User. See also *import*.

field

In the AR System, the main entity of a form. All of the following are AR System fields: data, table, page, active link control (buttons, menu items, and toolbar buttons), attachment pool, view, and trim.

field ID

A unique numeric identifier that is assigned to each field. Once assigned, it cannot be changed.

field label

A name supplied by the administrator that describes the field's purpose. Intended for display to the user.

field name

A unique character identifier that is assigned to each field. The name can be changed at any time as long as the new name is unique.

file menu

A menu with items retrieved from a file that contains a formatted character menu.

filter

A workflow component that tests every server transaction for certain conditions and responds to the conditions by taking specific actions. They are generally used to check and enforce business rules and processes. Filters run on the AR System server.

filter API

An AR System plug-in API that allows in-line access during filter and escalation processing to extended servers.

filter guide

An ordered sequence of filters that together accomplish a specific operation. Filter guides can be used as a subroutine to accomplish common tasks. Compare with *active link guide*.

fixed license

A license permanently assigned to a user, enabling access to the licensed features of the AR System at any time. See also *floating license*, *write license*.

floating license

A license that is temporarily allocated to a user who requests a license and who is defined as using a floating license. If no floating license is available at the time of the user request, the user must wait until a license becomes available. See also *fixed license*, *write license*.

form

A collection of fields that represents a record of information in the AR System. AR System administrators can define and change the fields and workflow associated with a form. An AR System application can include many forms.

form action field

One of a set of special fields that provide the standard functions of the web client interface. Some of these include submit, query, modify, and the advanced search bar.

form entry point

An entry point that opens a form in a particular mode, such as Search or New, so that a user can complete a task. See also *entry point*, *entry point guide*, *home page*.

form view

The screen layout for a form, which appears in the Details pane of BMC Remedy User or in a browser. AR System administrators can create and name multiple form views, which can be further modified by users with Customize permission. Administrators can include or hide different fields in various form views, and they can create views for particular locales or user roles.

function

Named procedure that performs a distinct service in the AR System. The AR System API has a set of function calls used to accomplish AR System tasks. Additionally, there are table functions that you can use to perform mathematical operations on table data.

functional currency

An alternate currency type to which the value in a currency field is converted. Values for functional currencies are calculated based on currency conversion ratios maintained in a form on the server. Functional currency values are stored as part of the data in the currency field, and can be viewed or searched by users, or accessed by workflow. See also *currency data type*, *allowable currency*.

global field

A display-only field whose value is the same across all forms that contain the field, as long as the user is logged in. See also *window-scoped global field*.

group

A category in the AR System used to define user access to form fields and functions. The AR System defines several special groups: Public, Administrator, Subadministrator, Customize, Submitter, Assignee, Assignee Group, and Flashboards Administrator. You can define additional groups through the Group form. See also *access control*, *explicit group*, *implicit group*, *computed group*, *dynamic group*, *permission*, *role*, *user*.

Group form

The form in which you add new groups, delete groups, and modify group permissions.

guest user

Unregistered user with a limited set of capabilities (submit requests and possibly review those requests). The administrator can specify whether unregistered users are allowed at your site.

GUID field

A field that is automatically populated with a globally unique identifier (GUID) when a request is saved.

guide

See *active link guide* and *filter guide*.

hidden field

A field that exists but is not visible in a user's view of the form.

home page

A form that lists entry points. The entry points are displayed within an application list field and include all of the entry points to which the user has access. In BMC Remedy User, the home page form can be configured to open by default on user login. In web clients, users enter a home page URL. See also *entry point*, *application list field*, *entry point guide*, *form entry point*.

implicit group

A group to which users are automatically assigned, based on the contents of certain fields in a request, such as Submitter and Assignee. See also *explicit group*, *group*.

import

1. The BMC Remedy Administrator command that transfers object definitions from an export file to the current server. 2. The BMC Remedy Import command that transfers one or more data entries from an archive file to a form. See also *export*.

independent copy

In a distributed environment, a modifiable copy of a request that is not part of the active ownership chain. See also *data-only copy*, *ownership chain*, *distributed server*.

integer data type

Data type used for fields that contain numeric values between –2147483647 and 2147483647. The range for a field can be limited by the AR System administrator.

join form

A type of form that contains information from two or more AR System forms. Although join forms function similarly to regular AR System forms in most respects, they do not store independent data. Join forms point to the data stored on the AR System forms that were used to create the join form.

keyword

A variable whose value is defined by the AR System. For example, `$USER$` represents the name of the user that is currently logged in. Keywords can be used in defining qualifications for searches, search menus, workflow, and macros; or for specifying a value in the Set Fields action for active links, filters, and escalations.

license

See *fixed license*, *floating license*, *read license*, *restricted read license*, *write license*.

local application

An application that uses permissions based on groups that is intended for use on a limited number of servers. See also *application*, *deployable application*, *role*.

macro

A set of operations within BMC Remedy User recorded for later execution. Macros are useful for automating frequently used or complex operations.

macro bar

The row of buttons below the menu bar in BMC Remedy User that enables easy access to commonly used macro commands. Macro commands are also available from the Tools menu.

mail template

A template that enables you to submit a request by using electronic mail. The AR System administrator generates templates from an existing form by using the `export` command.

main form

A form that users interact with directly. An application can use more than one main form. Main forms are sometimes called primary forms.

main window

The BMC Remedy User window that displays a form in the Details pane and, optionally, the results of a search in the Results pane and prompt text in the prompt bar. The main window includes a menu bar and optional status bar, toolbar, and macro bar.

mapping

The parameters for a particular distributed operation, such as To and From servers and forms, data control issues, and field-to-field mapping definitions.

mapping history

A history tracking record created when a distributed operation occurs. This record includes the date and time of transfer, source request ID, source form, source server, and the name of the specific mapping used.

master copy

The copy of a distributed request that currently has ownership.

master flag

A Yes/No flag indicating whether a distributed request is the master (has ownership).

menu item

A command accessed from a menu.

mid tier

The architecture level consisting of add-in services that communicate between AR System servers and various clients. BMC Remedy User can communicate directly with AR System servers. However, browser clients must use the mid tier add-in services to communicate with AR System servers.

navigation field

A field that allows users to navigate through screens in an application quickly and easily.

notification

A message to a user from workflow. Notifications can be alerts, email messages, or other methods using integrations.

ODBC

See *AR System ODBC driver*.

operator

One of a number of functions that enable you to define advanced searches or build qualifications.

ownership

In a distributed environment, having the ability to update copies of a request that are in the ownership chain. Ownership can be transferred from the original request to a copy, and ownership can be returned. See also *distributed server*.

ownership chain

In a distributed environment, the series of requests representing the history of transfers from the master copy of an original request, and updates to the original request. See also *data-only copy*, *independent copy*, *distributed server*.

packing list

A set of associated server objects that can be viewed as a work space in the server window or used in external utilities.

page field

A type of field that provides an area to group related fields. See also *page holder field*.

page holder field

A field that contains one or more page fields that allows display of one of the grouped field sets at a time in a given area of the screen.

pending operation

A distributed operation that is waiting to occur. Pending operations develop typically due to a specified transfer interval that has not yet been met, or because there are network or server-related problems within the distributed environment.

permission

The property setting that enables AR System administrators to control who can view and change individual fields of a form. Administrators also set permissions for forms, active links, active link guides, and applications. See also *access control*, *group*, *role*, *user*.

plug-in

An auxiliary program that works with the AR System server to enhance its capability. A plug-in is a dynamically linked library (DLL) on Microsoft platforms and a shared object on UNIX platforms. The plug-in service loads the plug-in at runtime.

plug-in service

A server that loads plug-ins. The plug-in service is a companion server to the AR System server that loads the ARDBC, AREA, or filter API plug-ins at runtime.

pools

See *distributed pools*.

preference server

An AR System server that stores user preferences centrally in one or more preference forms.

prompt bar

The part of the main window in BMC Remedy User that displays instructions or useful information about the form it is attached to.

Public group

One of several special access control groups that the AR System provides. Every user is automatically a member of this group.

qualification

A search criteria including field references, values, and arithmetical and relational operators used to find a set of data that matches the conditions specified.

query-by-example (QBE)

A method for describing a database search visually. An empty form is displayed and the search conditions are typed in or selected in their respective fields. The AR System turns the visual query into the proper command language, such as SQL, necessary to interrogate the database.

read license

A license that allows a user to search the AR System forms and to submit new requests but does not allow the user to modify existing requests. See also *restricted read license*, *write license*.

real data type

The data type used for fields that contain floating-point numbers. The range and precision can be set by the AR System administrator.

regular form

A form used to manipulate and display data. These forms and their contents are stored in database tables created, owned, and managed by the AR System server.

related workflow

Workflow objects that are related to a field.

request

See *action request*.

request ID

A unique numeric identifier for each request that is generated by the AR System in the Request ID field.

Request ID field

Core field on a form that contains the unique numeric identifier for a request or entry in the AR System. In join form entries, the Request ID field contains the Request ID field contents for each of the underlying forms, separated by a vertical bar. Previously called an entry ID.

reserved field

One of a set of fields with specific rules and interpretations that are defined by the AR System.

restricted read license

A license that allows a user to search the AR System forms and submit new requests but does not allow the user to modify existing requests under any conditions. It does, however, allow the same login to access the AR System from multiple machines simultaneously. See also *read license*.

results list

1. A list of requests that match a search. 2. A type of table field for displaying search results in web views. Multiple rows (i.e. requests) in the results list that meet specified criteria can be selected for further processing.

Results pane

The part of the form window in BMC Remedy User that displays the results of a search.

return

See *distributed return*.

return mapping

The specific field-to-field mappings used when a request is returned.

role

In a deployable application, defines access to form fields and functions. You define roles through the Role Mapping form, and then map the roles to groups on the server where the application is installed. You can map different groups depending on the development state of the application, such as Maintenance, Test, or Production. See also *access control*, *application state*, *deployable application*, *group*, *permission*, *Role Mapping form*, *user*.

Role Mapping form

The form in which you create roles and map them to groups for each application state. See also *application state*, *group*, *role*.

search

The process in which users display a subset of requests according to search criteria that they define.

search menu

A menu whose items are based on data retrieved from a search of an AR System form.

Section 508

A law that requires United States federal agencies' electronic and information technology to be accessible to people with disabilities (visual, motor, and auditory impairments).

selection data type

The data type used for fields with a set of mutually exclusive choices. Multiple selections are displayed as option buttons or as a list of items. A single selection can be displayed as a check box.

selection list
A list that appears when an active link performs a search that returns more than one request.

server
See *AR System server*.

server group
A group of AR System servers configured to share the same database. See also *AR System server*.

server tier
The architecture level consisting of the AR System server that controls access to the data and any supporting services called by or used by the AR System server.

Simple Object Access Protocol (SOAP)
The primary transport protocol for the messages shared by applications in web services. SOAP is an XML-based packaging format for the information being transferred, and also contains a set of rules for translating applications and platform-specific data types into XML.

source control
Managing server objects in AR System application development by controlling object access.

SQL menu
A menu whose items are based on data retrieved from a direct SQL command to the AR System database.

status bar
The part of a main window in an AR System client that displays instructions or useful information to the user.

Status field
The core field in which the AR System tracks the various stages of the resolution process for a request.

status history
Information that shows what progress has been made on a request. Users can view status history from any display showing the contents of an entry.

subadministrator
An individual with limited administrative access to the AR System as defined by an administrator.

Subadministrator group
One of several special access control groups that the AR System provides. Members of this group have limited administrative access to the AR System as defined by an administrator. See also *Administrator group*, *explicit group*.

Submitter group
One of several special access control groups that the AR System provides. Users automatically belong to this implicit group for requests they have submitted (that is, their name is in the Submitter field). See also *Assignee group*, *Assignee Group group*, *implicit group*.

supporting form
A form that supplies information needed by the main form.

table field
A field that displays data from other requests within the context of the current request. You can display a table field in a list view format (with rows and columns) or in a tree view format.

task
A shortcut or link created in BMC Remedy User that enables users to quickly open a specific form, search, application, or active link guide.

time data type

The data type used for fields containing time values. Time values are stored as the number of seconds from 12:00:00 a.m. See also *date/time data type*.

toolbar

The row of buttons below the menu bar in BMC Remedy User that enables easy access to commonly used menu commands.

toolbar button

In BMC Remedy User, an icon for a menu item that triggers an active link. See also *button field, menu item*.

transfer

A distributed operation that sends a copy of a request, with or without ownership, from one server to another.

transfer mapping

In a distributed environment, the specific field-to-field mappings used when a request is sent from one server to another. See also *distributed server*.

transfer mode

In a distributed environment, one of four types of transfer mappings that determines whether the copy of the request is sent with ownership, and whether the original is deleted. See also *distributed server*.

trim data type

The data type of fields that enhance a form's usability and appearance. Trim includes lines, boxes, text, and URL links. These fields do not contain data.

update

See *distributed update*.

user

Any person with permission to access the AR System. See also *access control, group, permission*.

user default

A value or set of values that a user can predefine. A user default overrides an AR System administrator default. See also *administrator default*.

User form

The form in which administrators add users to the AR System and specify each user's type of access and login information.

variable

A data element that changes according to the conditions.

vendor form

Forms that allow users to connect to external data sources, such as text or spreadsheet data, that reside on local or remote servers. See also *AR System database connectivity (ARDBC)*.

view

See *form view*.

view field

Field that provides a browser window within a form. Can be used to display a URL, the contents of an attachment, or direct HTML text.

view form

Forms that allow users to connect to external database tables accessible by the AR System database user.

view user interface (VUI)

A structure that contains information about a single view.

web client

An AR System client that runs in a browser to provide a user interface to AR System applications.

Web Service

Provides an interface that enables messages to be sent to and received from an application over a network (Internet or intranet), using standard Internet technologies. It uses a combination of protocols such as HyperText Transfer Protocol (HTTP) and Extensible Markup Language (XML), which are platform independent.

Web Service Description Language (WSDL)

An XML-based language used to define web services and how to access them.

wildcard

A character that users can enter to represent other characters in a search. For example, in search statements in character and diary fields, users can specify wildcards to match single characters, strings, or characters within a range or set.

window-scoped global field

A display-only field whose value remains the same across requests in the same window, as long as the window is open. See also *global field*.

workflow

1. A set of business processes used to run a company. 2. The automation of business processes through actions performed by active links, filters, and escalations.

work space

A subset of all objects on the server displayed in BMC Remedy Administrator. This subset of server objects is defined in a packing list.

write license

A license that allows a user to modify and save data in existing requests as field and form permission settings allow. See also *fixed license*, *floating license*, *read license*, *restricted read license*.

XML Schema or XML Schema Definition (XSD)

Provides a means for defining the structure, content and semantics of XML documents.

Index