

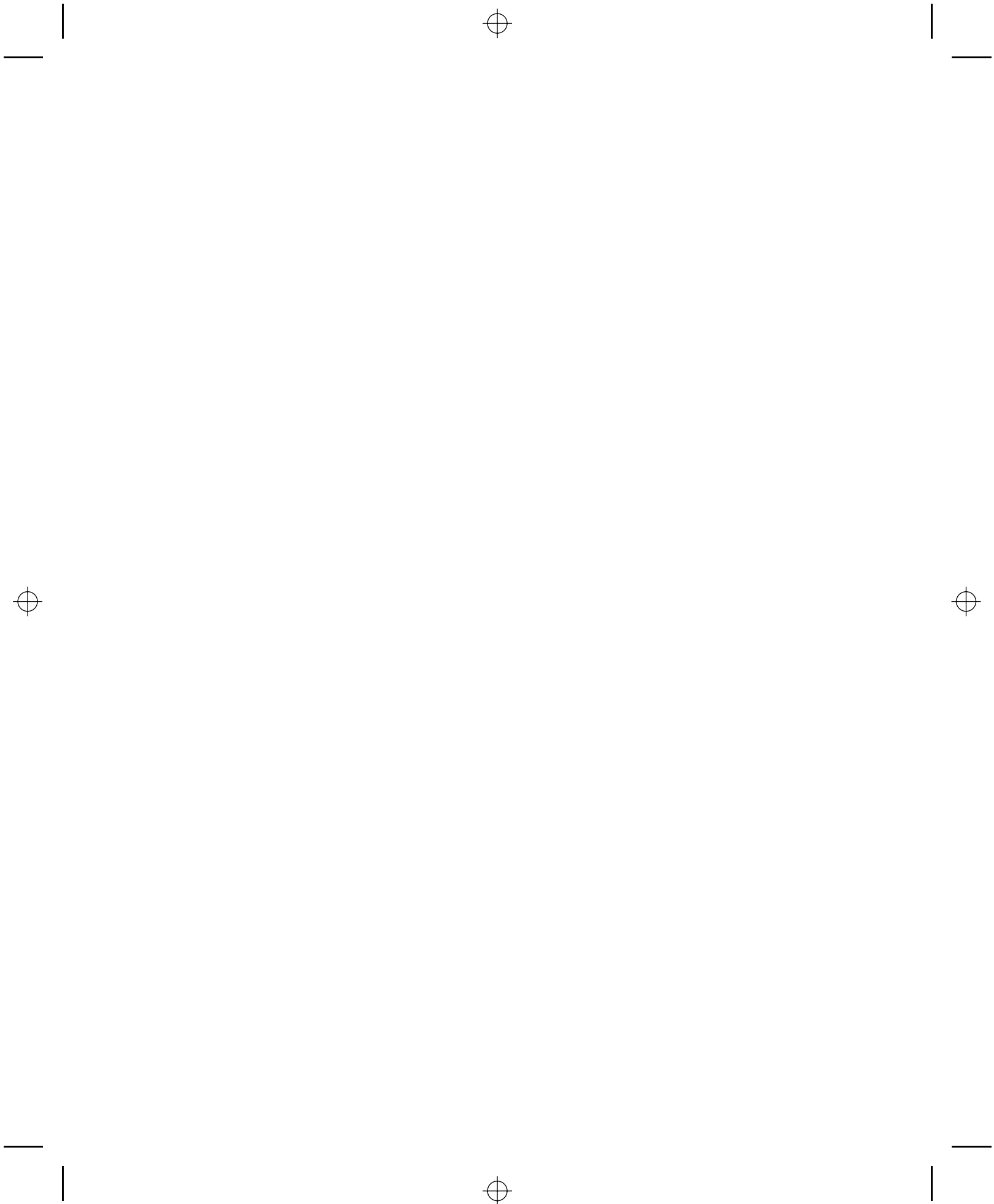
# PART I

## Getting to Know Alfresco

---

- ▶ **CHAPTER 1:** Introducing Alfresco
- ▶ **CHAPTER 2:** Architecture
- ▶ **CHAPTER 3:** Setting Up an Alfresco Environment

COPYRIGHTED MATERIAL



# 1

## Introducing Alfresco

### WHAT'S IN THIS CHAPTER?

---

- ▶ Understanding Alfresco and its uses
- ▶ Looking at the origins of Alfresco and its place in the ECM industry
- ▶ Using Alfresco in different scenarios
- ▶ Considering factors when implementing an Alfresco content application
- ▶ Exploring the importance of open source and community for Alfresco

Alfresco is an open source Enterprise Content Management (ECM) system. It was originally created in 2005 by a team from Documentum, including its co-founder, as an open source alternative to proprietary vendors in the \$4 billion ECM market.

Alfresco manages all the content within your enterprise: documents, images, photos, Web pages, records, XML documents, or any other unstructured or semi-structured file. What makes Alfresco stand out are the services and controls that manage this content and features, such as metadata management, version control, lifecycle management, workflow, search, associations to other content, tagging, commenting, and much more. This allows you to find the content you are looking for in the mountain of information accumulating in enterprises and to ensure that it is accurate. It also enables you to present and publish information through the Web or any other channel appropriate to allow users to access that information.

### For the End User

For end users, Alfresco appears as a suite of applications or extensions to their existing tools that manages their content. Alfresco exposes itself as though it were a shared drive to replace networked shared disk drives that have no organizational, search, or control mechanisms in place. Alfresco can replace networked shared drives with a store that organizes and controls information and provides a portal interface for searching and browsing content. By emulating the

SharePoint protocol, Alfresco also helps users manage their office documents from within Microsoft Office by using the tools in the Office Suite designed to be used for Microsoft SharePoint. More importantly, Alfresco provides an out-of-the-box suite of applications to browse, search, manage, and collaborate on content in the repository. These applications include document management, Web content management, content collaboration, records management, and email integration. These applications can supplement and can be supplemented by new applications developed on the Alfresco platform.

## For the Business

For the business, Alfresco is designed to support the content requirements of a number of business-critical processes and uses. The document management tools, applications, and interfaces support general office work, search, and discovery. The workflow management capabilities support numerous business processes, including case management and review and approval. The collaboration applications and services support the collaborative development of information and knowledge in the creation and refinement of content and documents. The scalable Web content management services support the delivery and deployment of content from the enterprise to its customers. The records management capability provides an affordable means to capture and preserve records based upon government-approved standards. The standards-based platform also provides access to applications that use these standards, such as publishing, image, and email management.

## For the Developer

For the developer, Alfresco provides a full-featured, scalable repository and content management platform to simplify the development of the content-centric applications. Based on content management and Internet standards, Alfresco exposes the content management capabilities as services that can be accessed from REST-based or SOAP-based Web services, the new OASIS Content Management Interoperability Services (CMIS) standard Web-based services, or the PHP programming language. It can also be incorporated directly into a Java-based application with core Java services. In addition, Alfresco incorporates lightweight scripting languages that can access these services and provide a lightweight-programming model when speed of development is important. These services provide patterns similar to those used with databases, repositories, or user interface components, but have been extended for the unique challenges of content-centric applications (such as full text search and hierarchical content structures). Being open source, the platform is transparent, and the developer can peer into the internal repository patterns. Alfresco also provides a framework application that delivers much of what end users need, but can be extended by the developer for unique application logic and a customized user interface through Surf, CMIS, Web scripts, and Core Services.

## For the IT Organization

For the IT organization, Alfresco provides a low-cost alternative to closed-source, proprietary systems from IBM, EMC, Open Text, Oracle, and Microsoft. Alfresco fits within the enterprise IT governance standards by working with virtually any database, application server, operating system, and system-monitoring infrastructure. By being a 100 percent Java application, the system is portable to virtually any hardware. A multi-tenant capability built into the core applications allows the IT

organization to provision virtual instances of Alfresco systems on-demand and maximize the use of existing hardware. The Alfresco system's small size also means that it works well within existing virtualization platforms such as VMware and Xen. Alfresco can also be configured into clusters with built-in redundancy to provide high availability and disaster recovery. Most importantly, Alfresco's open source and open standards approach means that users are not locked into a proprietary platform.

## WHAT IS ENTERPRISE CONTENT MANAGEMENT?

According to the Association for Information and Image Management (AIIM) — the leading professional group devoted to ECM — Enterprise Content Management is the collection of strategies, methods, and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes. ECM systems use a repository, a number of different applications, and application development platforms to enable this control, access, and delivery of content. Content can be any unstructured information, such as documents, Web pages, images, video, records, or simple files. (Unstructured information refers to computerized information that does not have a data model. The term distinguishes such information from data stored in fielded form in databases or semantically tagged in documents.) The ECM system manages the content and its lifecycle the way a database management system manages data in a database.

An ECM system manages the actual digital binary of the content, the metadata that describes its context, associations with other content, its place and classification in the repository, and the indexes for finding and accessing the content. Just as important, the ECM system manages the processes and lifecycles of the content to ensure that this information is correct. The ECM system manages the workflows for capturing, storing, and distributing content, as well as the lifecycle for how long content will be retained and what happens after that retention period.

According to the research group Forrester, there are five main application areas of ECM: document management, Web content management, records management, image management, and digital asset management. These areas of content management have similar requirements of storage, organization, access, and processing, but each has a different organizational focus and different end users.

- Document management tends to deal with the capture, editing, and distribution of office documents and files.
- Web content management organizes an enterprise's Web site, Web pages, and Web publishing processes.
- Records management deals with the long-term archival or disposal of important documents and records as well as any compliance or regulatory action.
- Image management may deal with documents or records, but handles content in the form of scanned images. Image management manages the process of scanning, quality control, metadata capture, and storage.
- Digital asset management is used primarily by creative and marketing professionals to handle the capture, creation, and editing of photos, videos, and illustrations.

By managing content in an ECM system, organizations are generally able to reduce costs of manual processing, increase the accuracy of information, and aid the search and discovery of important documents and information. Some of the benefits of using an ECM system include:

- Reduction of paper handling and error-prone manual processes
- Reduction of paper storage
- Reduction of lost documents
- Faster access to information
- Improved online experience for customers
- Online access to information that was formerly available only on paper, microfilm, or microfiche
- Improved control over documents and document-oriented processes
- Streamlining of time-consuming business processes
- Security over document access and modification
- Improved tracking and monitoring with the ability to identify bottlenecks and modify the system to improve efficiency

## The Origins of ECM

The ECM market started in the late 1980s with image management vendors like FileNet (now owned by IBM) and in the early 1990s with electronic document management such as Documentum (now owned by EMC). Document management married the then–relatively new relational database systems with electronic file management, integration with scanning equipment, and workflow management tools. These companies built platforms to develop applications for managing electronic documents in the emerging client-server enterprise application space and enterprise systems that were large in both scale and price. In the mid 1990s, as the Internet became important, Web content management vendors, such as Interwoven and Vignette, entered into the Web space adjacent to document management.

With the collapse of the dot com bubble in 2001, healthier companies started to acquire smaller players with overlapping functionality. As document management and Web content management vendors started to compete with each other and acquire new technologies, such as archival and records management, the ECM market was born. It was in this period that Microsoft entered with SharePoint. With the intervening years, there are now fewer, but much larger, companies in the growing ECM space. The ECM market is now estimated to be \$4 billion according to the analyst firm Gartner.

## Enter Alfresco

In 2005, Alfresco started using the open source development and distribution model to spread ECM globally. This enabled them to address underserved parts of the market where existing ECM systems were either too expensive or too complex. At the time, existing ECM vendors put more effort into consolidating disparate product sets than creating new technology, resulting in more difficult development, deployment, and usability, which adversely impacted scalability and performance of these systems. Older technology design, built with no reusable components, meant that these systems were very large, incurred a high overhead for managing information and storing data, and were very expensive to

develop and maintain, which affected the cost of the systems. Extending these systems became a major integration problem and applications built upon these systems could cost as much as ten times the sale cost of the core repository and system.

Alfresco built a different kind of system using open source development and incorporating open source components, such as Spring, Lucene, Hibernate, jBPM, FreeMarker, and POI. The Alfresco system incorporated the major applications of ECM — document, image, Web content, record, and digital asset management — but in an easier, more deployable package.

By adopting appropriate open standards, Alfresco generally fits in any enterprise environment. Alfresco’s Content Application Server provides a platform for developing content applications in a number of different development styles suitable for the programming task at hand. By being scalable, Alfresco can adapt from small departmental solutions to large-scale Internet solutions.

The end result is a system that is lightweight, flexible, and easy to deploy, with a powerful set of development interfaces. Alfresco recently exceeded 2 million downloads of the ECM system with a large and growing community. Using the professional open source model, Alfresco now has over 1,000 enterprise customers.

## ALFRESCO OVERVIEW

The Alfresco system in many ways looks similar to other ECM systems. (See Figure 1-1.) At the core is a repository supported by a server that persists content, metadata, associations, and full text indexes. There is a set of programming interfaces that support multiple languages and protocols upon which developers can create custom applications and solutions. Out of the box applications provide standard solutions such as document management, records management, and Web content management.

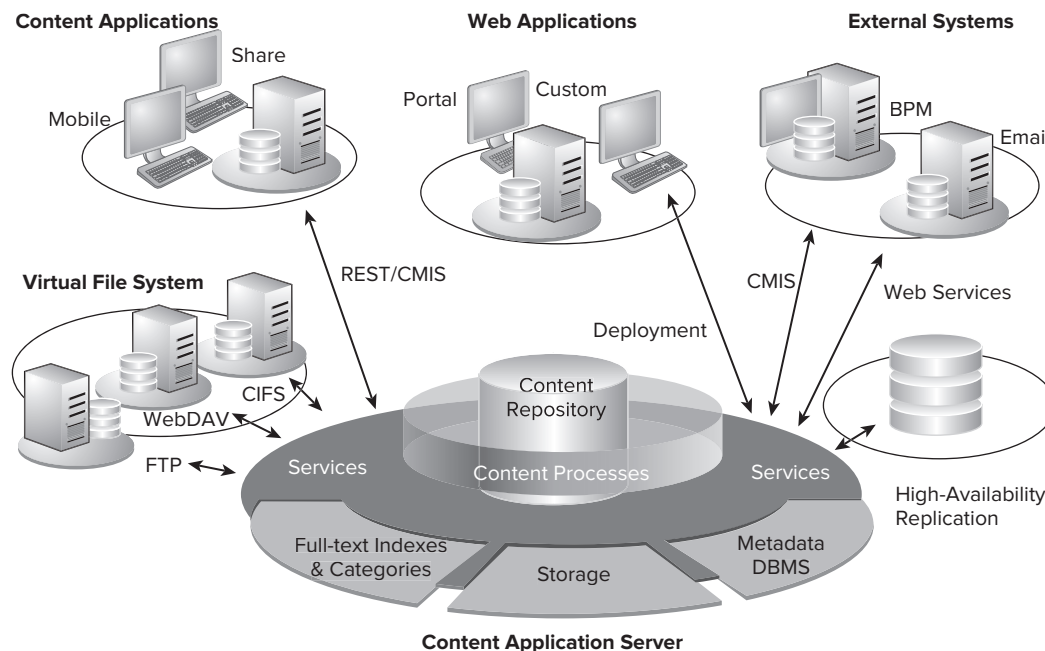


FIGURE 1-1

However, because Alfresco has been created relatively recently compared to other ECM systems, it has been able to take advantage of a more modern architecture. The Alfresco system has grown organically as an entirely Java application, which means that it runs on virtually any system that can run Java Enterprise Edition. At the core is the Spring platform, which provides Alfresco the ability to modularize functionality such as versioning, security, and rules, among other things. Alfresco makes liberal use of scripting to simplify adding new functionality and developing new programming interfaces. This portion of the architecture is known as *Web scripts* and can be used for both data and presentation services. Alfresco has kept the architecture lightweight to make it easy to both download and install and to be able to take advantage of new packaging and deployment options such as in the Cloud.

## The Content Application Server and the Repository

At the heart of the Alfresco system is the Content Application Server, which manages and maintains the Content Repository. The repository is comparable to a database except that it holds more than data. The binary streams of content are stored in the repository and the associated full-text indexes are maintained by the Lucene indexes. The actual binary streams of the content are stored in files managed in the repository, although these files are for internal use only and do not reflect what you might see through the shared drive interfaces. The repository also holds the associations among content items, classifications, and the folder/file structure. The folder/file structure is maintained in the database and is not reflected in the internal file storage structure.



*Lucene is an open source full-text retrieval engine that is part of the Apache project.*

The Content Application Server is responsible for the business logic for the control, access, and update of content in the repository. The Content Application Server allows you to execute applications either as Web scripts or as Java extensions. All the applications of the Alfresco ECM suite are built upon and executed by the Content Application Server. The sample Knowledge Base application used later in this book is built using the Content Application Server and the Alfresco Share application.

## Alfresco Applications

Alfresco applications are built upon the Content Application Server and rely on the Content Application Server to persist, access, query, and manage content. The Alfresco applications exist to provide the basic capabilities that most users need to manage content. The two main applications are Alfresco Share and Alfresco Explorer.

Alfresco Explorer is the original application built with the Alfresco system to manage content. Alfresco Explorer allows you to browse the repository, set up rules and actions, and manage content and its metadata, associations, and classifications. Alfresco Explorer was built using JavaServer Faces and is integrated into the Content Application Server. It is currently being phased out in favor of Alfresco Share. However, many extensions and language packs have been built for Alfresco Explorer. It also has extensive capabilities for managing the repository and should be considered a system administrator tool.

Alfresco Share is the next-generation user interface built entirely with the Alfresco Web script technology and can be used to extend the application. Alfresco Share provides content management capabilities with simple user interfaces, providing users with tools to search and browse the repository; content as thumbnails and associated metadata; previews using Flash renditions of content; and a set of collaboration tools such as Wikis, Discussions, and Blogs. Alfresco Share is also the foundation for Alfresco records management. Alfresco Share is organized as a set of sites that can be used as a meeting place for collaboration.

Most importantly, Alfresco Share is an extensible application foundation that can be extended to create vertical applications for certain vertical industries or specialist domains. Records management is a good example of this. The sample Knowledge Base application created in this book also relies on the Alfresco Share core content management capabilities. There is no need to reinvent the content management interfaces of Alfresco Share in order to provide the general-purpose content controls for your end users.

## The Alfresco Web Tier and Surf

Alfresco provides ECM capabilities as data services, user interfaces, and user applications. The user interface capabilities are provided by applications and application components using Alfresco's Web tier, Surf, originally developed as a faster way to develop content applications using scripting and REST architecture. Surf uses Web scripts heavily to build user interface components and to access information from the Content Application Server. In 2009, Alfresco contributed Surf as a project to the Spring community where it can be used in conjunction with other Spring Web-tier components, such as Spring MVC, Spring Webflow, and Grails.

Web scripts were developed as part of Alfresco version 2.0 to introduce REST-based development in Alfresco. Web scripts are described in Chapters 8 through 11 and rely heavily on the Web 2.0 style of scripting: interpretive execution of small code pieces that can be configured at runtime. This allows the system to be very adaptable and speeds development of new capabilities. Prototyping is very popular in this environment. If performance is an issue, then Java can be used, but the Web script infrastructure can accommodate Java beans as easily as JavaScript. Web scripts add little overhead but provide a great deal of flexibility and development productivity. Web scripts in the Web tier make it possible to quickly build user interface components with Surf or simple HTML, and deploy them as Alfresco Share components, portlets, or other Web platforms such as Google Gadgets.

## Alfresco Programming Models

In order to build an application using the Alfresco Content Application Server, you have the choice of a number of programming models. The simplest model for non-programmers is to use out-of-the-box components of the Alfresco Share application and to use the Rules and Actions model, a set of conditions and actions to take on content, based on those conditions. Rules and actions can be defined using a wizard and can perform such actions as convert content, move content, or execute a simple JavaScript snippet. The sample Knowledge Base application uses rules and actions to do basic processing of content entered into the Knowledge Base.

For more sophisticated processing that doesn't require complex programming, you can use Web scripts. In fact, the Alfresco CMIS implementation and Lotus Quickr integrations were built using Web scripts. By using JavaScript to build these data services, it is easy to create new services in the Alfresco

system. If you need to build new user interfaces or extensions to the Alfresco Share application, then you can also use Web scripts by using a Web templating language like FreeMarker. Most of the Alfresco Share application was built using Web scripts. Chapters 8 through 11 show you how to build Web scripts.

If you wish to or need to use Java to build applications or extend Alfresco Share, you can use the many tools associated with Java that were used to build the Alfresco system. Surf, the Web runtime framework contributed by Alfresco to the Spring project, is covered in Chapters 12 through 16 for both extending the Share application and building your own Web applications. Because Share was built using Surf, it is possible to build your own extensions as a combination of Java programming and Web scripts, or with Java alone. You can also use Java to access or even replace whole pieces of Alfresco, Content Application Server, or Alfresco Share by using the Spring platform. This is not covered in this book, but you can use the source code as an example for rewriting pieces and using Spring beans and configuration to extend or replace functionality in Alfresco.

Finally, if you wish to write applications that use Alfresco but are portable to other ECM systems, then you can use Content Management Interoperability Services (CMIS), the OASIS standard for accessing content repositories (including Alfresco, EMC Documentum, IBM FileNet, and Microsoft SharePoint, among others). The sample Knowledge Base application uses some examples of CMIS to show you how you can build portable applications.

## CONTENT APPLICATIONS

Alfresco can be used to build most ECM applications. These are the applications in which enterprises use content to engage customers, partners, and employees. These applications are content-rich and their business processes tend to focus on the creation, management, delivery, access, and use of content. Aside from the major applications of document, image, records, digital asset, and Web content management, there are a number of specific applications and use cases that add value to the enterprise and reduce both cost and waste. The following applications are typical of ECM and are ones that Alfresco is particularly adept at supporting.

The Alfresco applications can often be used as the foundation for these application use cases. By applying the programming models previously discussed, you can extend these applications or build your own applications using Alfresco.

### Knowledge Management

Knowledge management is the capture of knowledge (from employees in an organization or customers who use their products) and the provision of that knowledge in a form that others can use. Content tends to be the best and most reusable container of knowledge in sharing that knowledge with others. This book uses a knowledge management sample application as the basis for demonstrating how to extend Alfresco and build your own applications.

### Document Management

Document management is one of the oldest ECM applications and represents over a quarter of the ECM industry. Document management is used to manage and share office documents and also to incorporate

the business processes of creation, review, approval, and distribution of documents. It can be industry- or role-specific (such as management of contracts, proposals, specifications, and procedures), while still using a lot of generic capabilities. Therefore, applications like Alfresco Share and Explorer are good foundations for building document management applications.

## Shared Drive Replacement

A more basic form of document management is the replacement of shared drives in the enterprise with a content repository that provides easy access points to content. Shared drives are simple to use because users don't need to be trained and all applications work with them; however, they make a mess of organization, cause a regulatory headache when information is uncontrolled and undiscovered, and are impossible to search in order to find information in terabytes of storage. Because Alfresco supports the protocol used by shared drives, Common Internet File System (CIFS), the repository appears to be a shared drive. With rules, actions, and extensions, you can build complete document management applications that are transparent to the user while getting the content under control and enabling it to be searched.

## Enterprise Portals and Intranets

Corporate portals or intranets are used to communicate with employees and keep them abreast of news and developments in the enterprise. While part of enterprise portals are focused on reporting and analyzing data, a large portion of portals are devoted to content and documents. Although folder hierarchies may be an easy way to organize information for a portal, classifications and metadata are often a better way to target information in the portal to end users. Thus, there are elements of document management and business process for delivering into the portal; however, the presentation of lists of content and navigation through classifications may require programming portlets using Web scripts or Java. Portlets provided as part of the Alfresco platform can supplement this development with standardized navigation, search, and content presentation.

## Web Content Management

Web content management is about managing Web sites, managing the content that goes into Web sites (such as HTML and images), and managing the processes of building, testing, and deploying Web sites and content. While Alfresco can be used for simple Web sites, it is frequently used for creating Web sites that are Web applications, particularly those developed using Java. Some examples of these Web sites publish a lot of information from multiple sources and integrate e-commerce and back-office systems. Surf is a good platform for creating these types of Web applications and Web sites, and the examples used in this book can be helpful for you to develop them.

## Information Publishing

Information publishing is a specialized form of Web content management. It involves the real-time publishing of content from different sources to the Web site and the deployment of that content to the Web farm for Internet access. This can take the form of digital assets such as articles, written internally or syndicated from other sources, or photos. Alfresco is used by many media companies to combine this content and publish it to their Web sites. This straight-through publishing of information requires both strong content control and performance to aggregate and push out the content. Alfresco is well suited to this type of Web content management.

## Records Management

Records management is used to control important information that needs to be retained for extended periods of time. You would use records management over document management in regulated or compliant environments, such as in managing governmental information or personnel records, or where information may be audited. Alfresco has been certified to the U.S. Government 5015.2 records standard and is useful for controlling retention and review periods, providing specialized security, and determining whether the records are archived or destroyed after a specified period of time. However, no two records repositories are the same and although Alfresco records management provides a good foundation, you may want to provide additional functionality through configuration or customization. At the very least, you will need to configure the file plan specific to the department deploying records management.

## Collaborative Content Development

Many knowledge-oriented business processes rely on content as the container of information captured and developed as part of the creative process. Collaborative work such as new product development, sales proposal preparation, or strategic planning more often than not have a document of some sort (a Word document, a PowerPoint presentation, or an Excel spreadsheet) that represents the fruit of the effort. Sometimes the content is the purpose of the collaboration, such as the creation of a report or a book like the one you are reading right now. Alfresco was used to manage the development of this book, and it is a good example of collaborative content management. Alfresco and Alfresco Share are useful tools in collaborative content management that aids knowledge work. Alfresco Share also provides a good platform to build specialized tools and extensions that may help either with collaboration or with handling content in the collaborative and creative processes.

## Case Management

Case Management is the handling of information related to a case, such as an insurance claim, an investigation, or personnel processing. Because of Alfresco's document management capabilities, folder structure, classification schemes, and workflow, it is well suited to managing cases and distributing work in handling cases. Alfresco incorporates the jBPM business process engine from JBoss and can handle sophisticated workflows and queue management. Alfresco has built a content-oriented task model that aggregates all the resources required to perform specific tasks within the case handling process.

## ALFRESCO AND YOUR ENVIRONMENT

Alfresco is applicable for most ECM requirements. It is currently deployed in some of the largest organizations in the world, spanning finance, high technology, manufacturing, and government. It has been used by small, medium, and large businesses at the departmental level, enterprise-wide, and across the Internet. Depending on what type of content application you want to create or deploy, Alfresco can be used and configured to meet the requirements of those applications.

In order to make sure that Alfresco meets your needs and can be an effective content platform and content management system, you should evaluate your environment, your target solution, and which tools are best suited to meet your solution.

## What to Consider

Although Alfresco can scale from small solutions to enterprise-wide infrastructure, it is important to make sure that Alfresco is configured in the right way and that the solution has been programmed with the most appropriate tools and interfaces. Here are some of the areas that you should consider when developing your content application.

### The Audience

The Alfresco system has tools for broad usage and for sophisticated users of content management. By fitting within the user's environment, such as the file system, Microsoft Office, or email, Alfresco can provide services invisibly to a broad range of end users. Alfresco also integrates with enterprise portals and applications where users access content as part of another solution or application. In addition, there are interfaces that allow users to configure powerful rules and workflows without programming. By determining who your audience is and where they need to access content, you can choose the right deployment choices for Alfresco.

Chapter 2 discusses the tools available for presenting content management to the end user. The sample Knowledge Base application uses several different tools to show how you can use interfaces as simple as a shared drive, as well as more sophisticated knowledge-processing tools. Chapter 17 presents you with the patterns of how Alfresco can integrate with other applications.

### The Number of Users

After you've downloaded Alfresco from the Alfresco Web site, the system is capable of easily handling dozens of users. However, if you wish to deploy to an entire enterprise, you may want to consider options of clustering and distribution. Chapter 2 shows you different configurations and the impact on development. Alfresco can run as part of a single application or in a multi-tier environment clustered with many machines.

### The Architecture

Is this application standalone or part of a broader content infrastructure? If content is reused between applications, you should consider a remote repository from your application and application server. If absolute performance is important, perhaps it makes sense to integrate your application, repository, and database as a single application. If you are in a distributed environment or parts of your content architecture are geographically distributed, consider either federation or synchronization options. Understanding your architecture will be key.

### The Level of Configuration or Customization

Analyze how your application will be deployed, how it will be used, and how long you can expect to use it in order to determine whether you want to build a new application, customize one of the existing Alfresco applications, or simply configure the out-of-the-box applications. Configuration can be very powerful, especially in configuring rules and actions or in adding out-of-the-box components. If you choose to customize, you have the option of scripting using Web scripts, thus avoiding heavyweight development. Alternatively, you can develop using a full Java environment using your favorite IDE. Customization can take the form of customizing the Alfresco application or the Content Application Server itself. The sample Knowledge Base application will show you how to do all of these.

## The Enterprise IT Environment

The Alfresco system and applications are written in Java and Java-based technologies to ensure that they are as portable as possible. They are quite possibly the most portable ECM systems available. In addition, Alfresco incorporates Hibernate and iBATIS open source database abstraction layers to allow it to be ported to a number of different database management systems. By using the Spring platform, Alfresco can integrate different caching technologies and security and authentication systems. Alfresco is designed to support different languages as well, allowing you to develop using Java, PHP, .NET, or virtually any language that supports REST-based or Web services-oriented programming.

Thus, practically any IT environment can be supported, but operating system, database, language, and security should be considered when developing and deploying your application. The *Installing and Configuring Alfresco* guide on the Alfresco wiki will give you guidance on how to configure for different databases and authentication mechanisms. Chapter 17 provides advice on how Alfresco can be integrated with other applications.

## The Support Requirements

Alfresco is open source, so it can be used for free. If you are a small organization or small independent software developer, this can be a great option for using content management without the expense of a large ECM system. However, if you wish to deploy your application in a production environment, you may wish to use the Enterprise version of Alfresco, which provides you with support and prompt bug fixing. Also, if you are running with proprietary databases or operating systems, then the Enterprise version provides configurations on certified platforms that include different operating systems and databases such as Oracle or Sybase. You should consider how you build and deploy your application and determine what your support requirements are based upon these activities. The more mission-critical or the more integrated into the enterprise, the more you may need the Enterprise version.

## WHAT IS PROFESSIONAL OPEN SOURCE SOFTWARE?

Alfresco's ECM capabilities are available as open source. If you are a content management developer who has worked with proprietary platforms, you may not be familiar with the implications of open source, nor with the advantages or disadvantages for the products.

Much of the early development of what is now known as open source started with the GNU project by Richard Stallman in the early 1980s to create development tools that emulated Unix but did not require a license from a vendor. Stallman labeled this "Free Software" and felt that the freedom to use this software was paramount. In the mid 1990s Linus Torvalds created a completely open and free Unix-based operating system called Linux under the same license terms as GNU. The Linux movement created a whole genre of software that became known as *open source*. In 1999, IBM backed the Linux operating system and the open source movement in general and thereby gave credence and momentum to open source.

In the early part of the 2000s, entrepreneurs created enterprise software businesses based upon open source and employed a professional open source business model. The database system MySQL, originally developed in the early 1990s, used the GNU GPL (General Public License) to distribute their software. They did this using a dual-license business model that distributed the software under GPL and commercial licenses. Shortly after this, JBoss built and licensed its application server under the

more liberal LGPL (Lesser/Library General Public License). JBoss and others raised venture capital and thrived in a down economy, proving the value of distribution via the open source model and developing business through selling support and services.

## Why Use Open Source?

Open source is a revolution in business models and a means of production rather than anything technological. It is underpinned by three main principles:

- Opening intellectual property and the transparency approach spreads faster than concealing intellectual property and builds a bigger market.
- Sharing intellectual property will grow that intellectual property faster.
- Reusing other people's intellectual property stops wasting time re-inventing the wheel.

This change in business model has led to companies using open source as their preferred distribution model, including Alfresco. These advantages include:

- Broad distribution of product through Internet download
- The ability of users to try the product before buying, eliminating much of the sales and demand generation activities of ordinary enterprise software
- Much broader testing and code review of products in the beta and product phases of development
- Word-of-mouth marketing and open source meet-ups, reducing the cost of marketing
- Lower cost of development through use of other open source components and contributions from the many extensions that users create
- Bug fixes from users who end up fixing problems themselves

The benefits to the user and business community also are clear:

- Users are not locked into a proprietary platform and they are free to use the open source version of the product.
- The vendors providing these products are the support source upon which users and businesses can rely if they need fixes, advice, warranty, or indemnity.
- The open source vendors are held more accountable than the proprietary vendors, since there is always the possibility to switch back to the free version.
- Users could see the source code and make the minor changes they want without waiting for the next release or even change wholesale parts of the product.
- Because of the lower cost and economies of scale of open source distribution, the lower cost of product is passed on to the consumers of services.
- The transparency of open source means that there is greater visibility of changes coming and of what bugs already exist.
- Through the community and transparency aspects of open source, the users play a more intimate role in the development of the products, with a greater influence on product direction.

There are lots of open source programs and projects out there, and some of them are terrible. However, open source works well when software is a commodity and you cannot tell one piece of software from another, such as word processors, Web browsers, and operating systems. When software is a commodity, you are paying only for a brand where the company receives money without improving or innovating the product. Professional open source works well when enterprise software is commoditized. Alfresco was created because of the increasing difficulty in distinguishing one ECM system from another, and open source can allow the community to move Alfresco beyond the others.

Open source development is painted as chaotic and unreliable by proprietary vendors, but in reality it is just as controlled as any other project. There are project leads who make decisions (and have the final decision on what goes in or out) and no more than a handful of people developing the open source project. Sometimes these developers can come from more than one company. With open source this works fairly seamlessly because there are no secrets or intellectual property to hide. There can, however, be much more testing involved than with ordinary software. In most successful open source projects, you can have hundreds of people downloading the software and testing it out and sometimes even providing fixes because they can see and repair the source code.

In order to make this process work and to build a sustainable open source community, somebody must pay for it. Usually an open source company is in markets directly adjacent to the software, such as system integration. By supporting Linux, IBM makes its money selling hardware that runs Linux and selling consulting services to build solutions for customers. IBM has a huge patent portfolio, but they make twice as much money selling hardware and services around open source as they make from licenses from their patents.

Alfresco Software is building a sustainable business and thus a sustainable open source community by selling technical support on the Alfresco system. Not everyone wants or needs support, but it isn't necessary for everyone to want it. With over 2 million downloads, a small percentage of users who need support in a product environment, especially in the Global 2000, is sufficient to support the community, the project, and the company. By participating in the large enterprise content management market, Alfresco has sufficient scale to continue growing and to service the open source community and product.

## Alfresco and the Community

Because Alfresco has built its development processes and its business model on open source, the Alfresco community is extremely important. The process of scoping the product, tracking development, and delivering the product is open and is what distinguishes it from proprietary content management vendors. All the source code is available from the Subversion repository; documentation is available and developed from the wiki; forums provide feedback on plans, bugs, and features; and all the bugs and project plans are available in JIRA through the Internet. You can expect Alfresco to be open about how they build and release product. They won't be perfect, but they believe this transparency holds them accountable to provide a better product for you.

The great thing about open source is you don't re-invent the wheel. If someone has created a piece of code, then others can share it. Ideally, if a community member stumbles across a bug in that code, there is enough information in the code to fix it. If there isn't enough information, then someone else can provide missing documentation. Pitching in and sharing helps save time and effort.

The community has already contributed to Alfresco in many ways. Some code and functionality contributions have helped distinguish Alfresco as a product. Contributed bug fixes allow Alfresco engineers to concentrate on new functionality. Community translations of the user interface enable access to Alfresco around the world. Alfresco has already been translated into more languages than some major products will ever see. Integrations and solutions have been a particularly rich vein of sharing. More experienced users and developers have been extremely helpful in assisting new users in the forums.

If you do develop a solution or can contribute, please take the time to share, as it not only helps Alfresco, but also assists users like you. If you have a solution or translation based on Alfresco, contribute it to the Forge. If you have a bug fix, contribute it through JIRA. If you have time to help others or you see a question that you can answer in the forums, please do, and others will return the favor.

