Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The alfresco and share WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within ~tomcat/webapps/alfresco or ~tomcat/webapps/share. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let's explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the amps directory.

The second file I place in the amps_share directory.

I stop the Alfresco server.

Next I use the apply_amps command, which is found in the bin directory. This command applies the AMP files that are located in the amps and amps_share directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The apply_amps command has copied the content into the alfresco.war file. When the server starts the alfresco.war file is expanded and deployed into the ~tomcat/webapps directory, maintaining the same folder structure.

The same is true for the Share components from the alfresco-rm-share AMP file, now found in the share.war file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

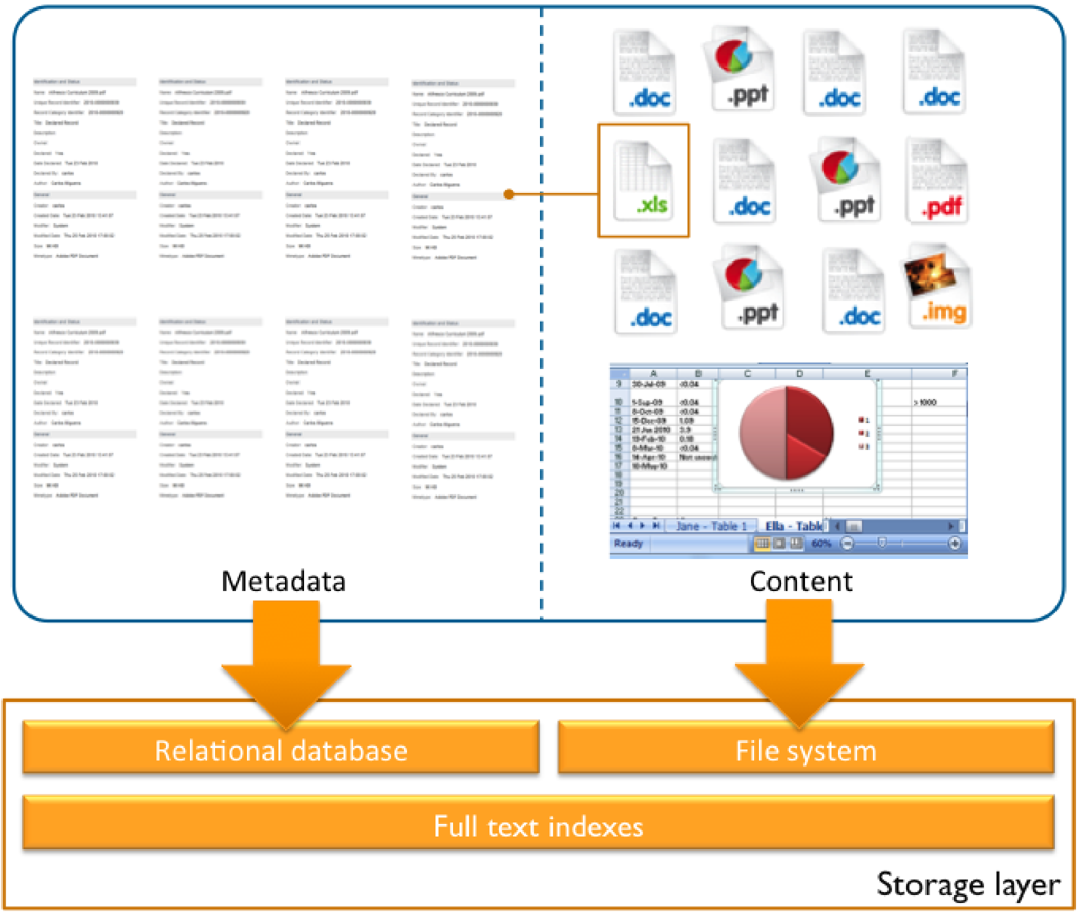
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

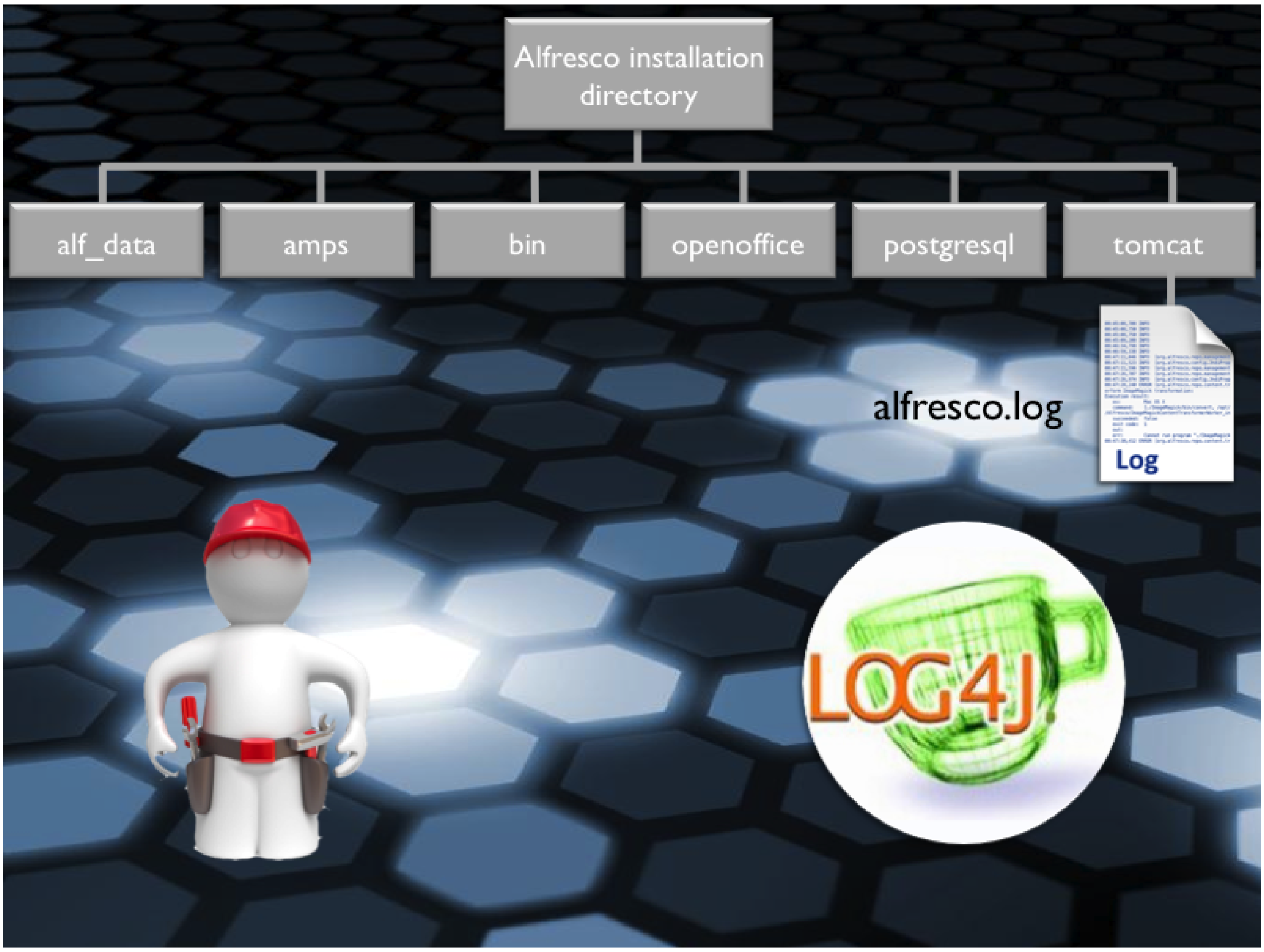
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

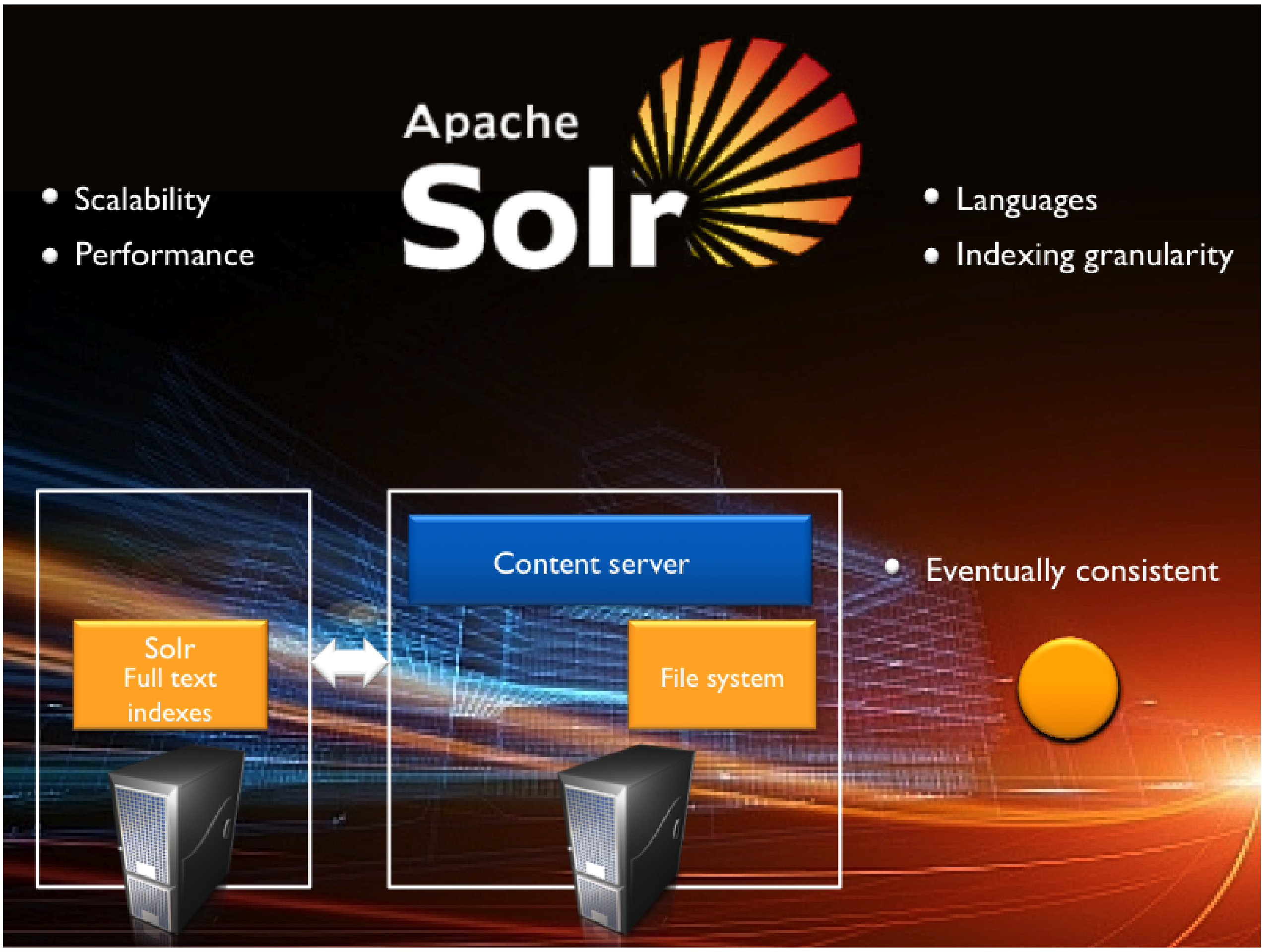
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

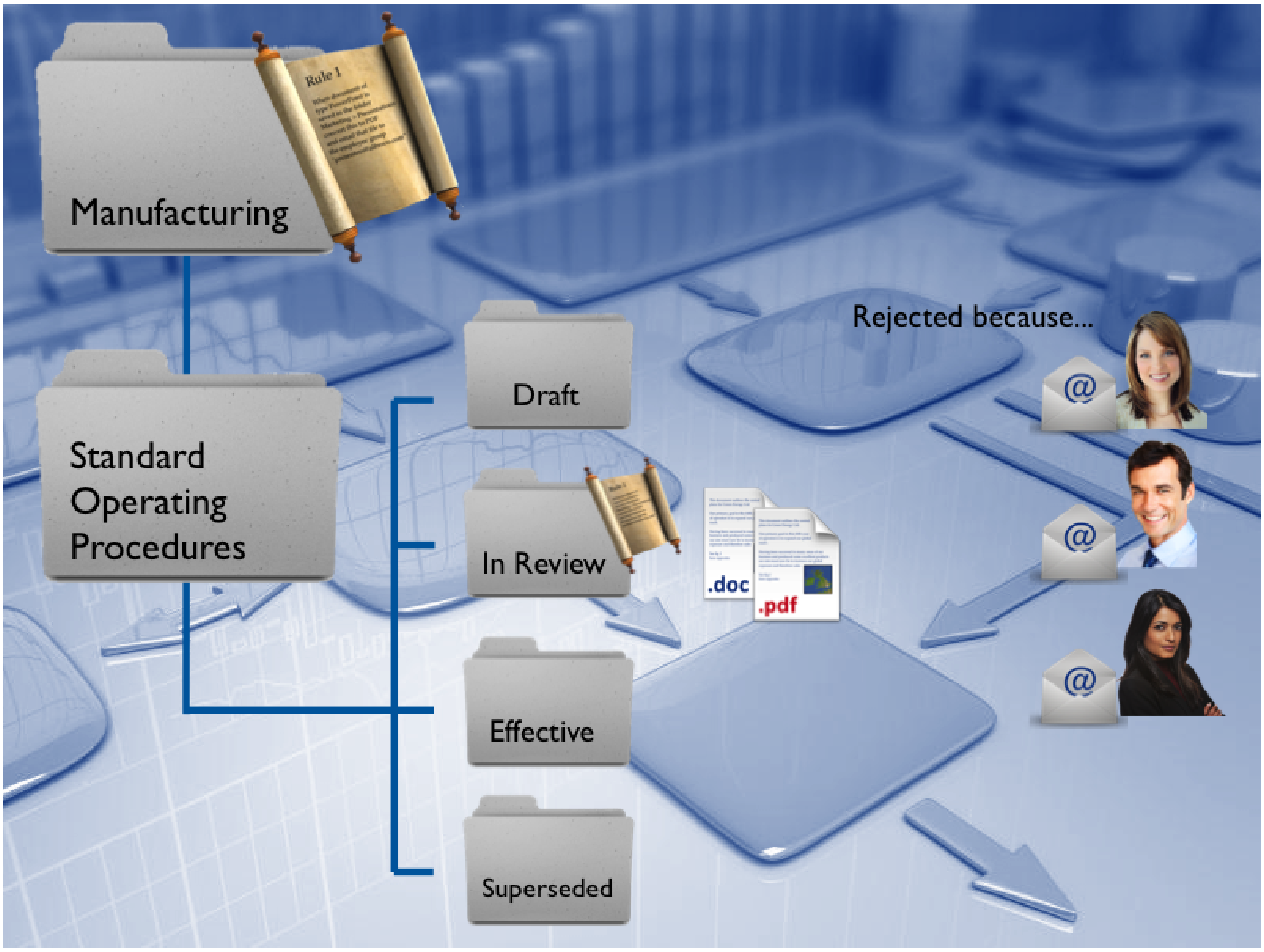
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

   1. Navigate to the folder `Geo-Thermal Division` in the repository.
   2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

      `Geo-Thermal Division > Marketing > Presentations`

      - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
      - Make the rule inheritable.
      - Set the rule to run in the background.
   3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

      `Geo-Thermal Division > Marketing > Branding Project`

      The Building a Brand Identify file is found in the folder:

      `Desktop > Assets > Managing the repository`
   4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

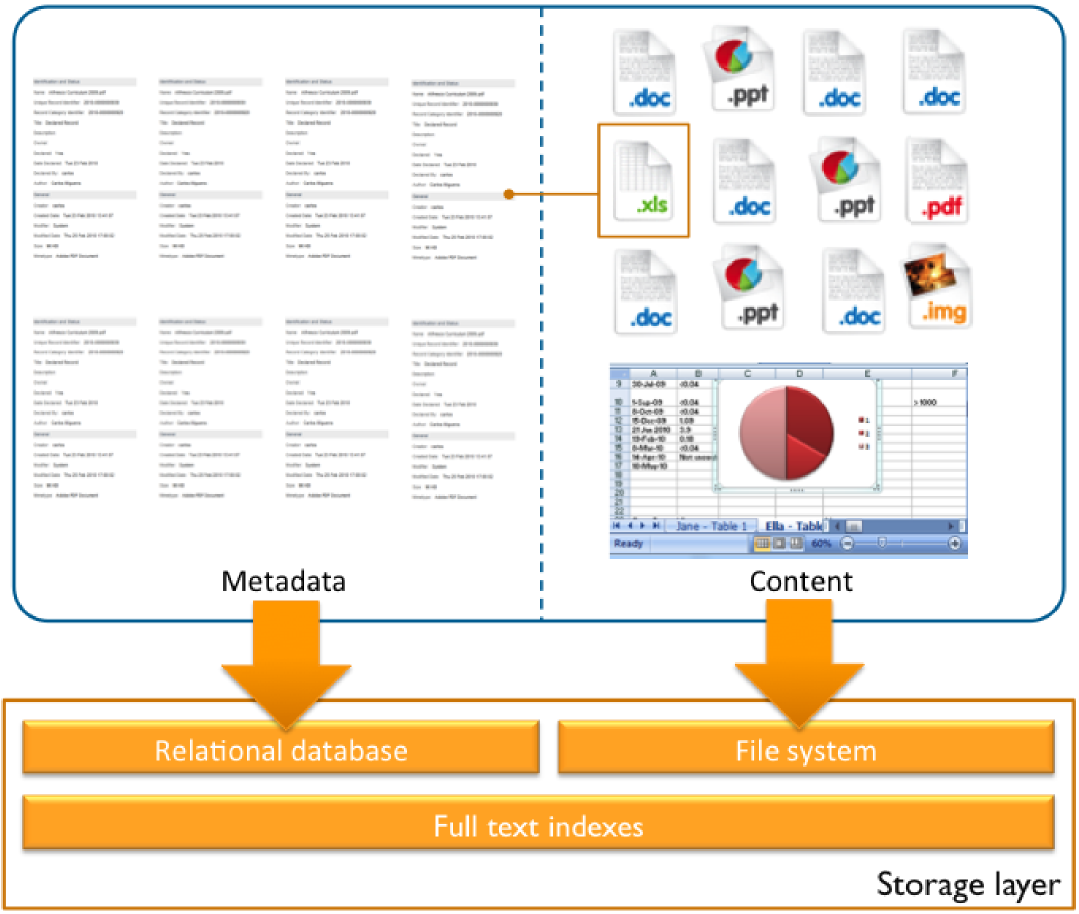
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

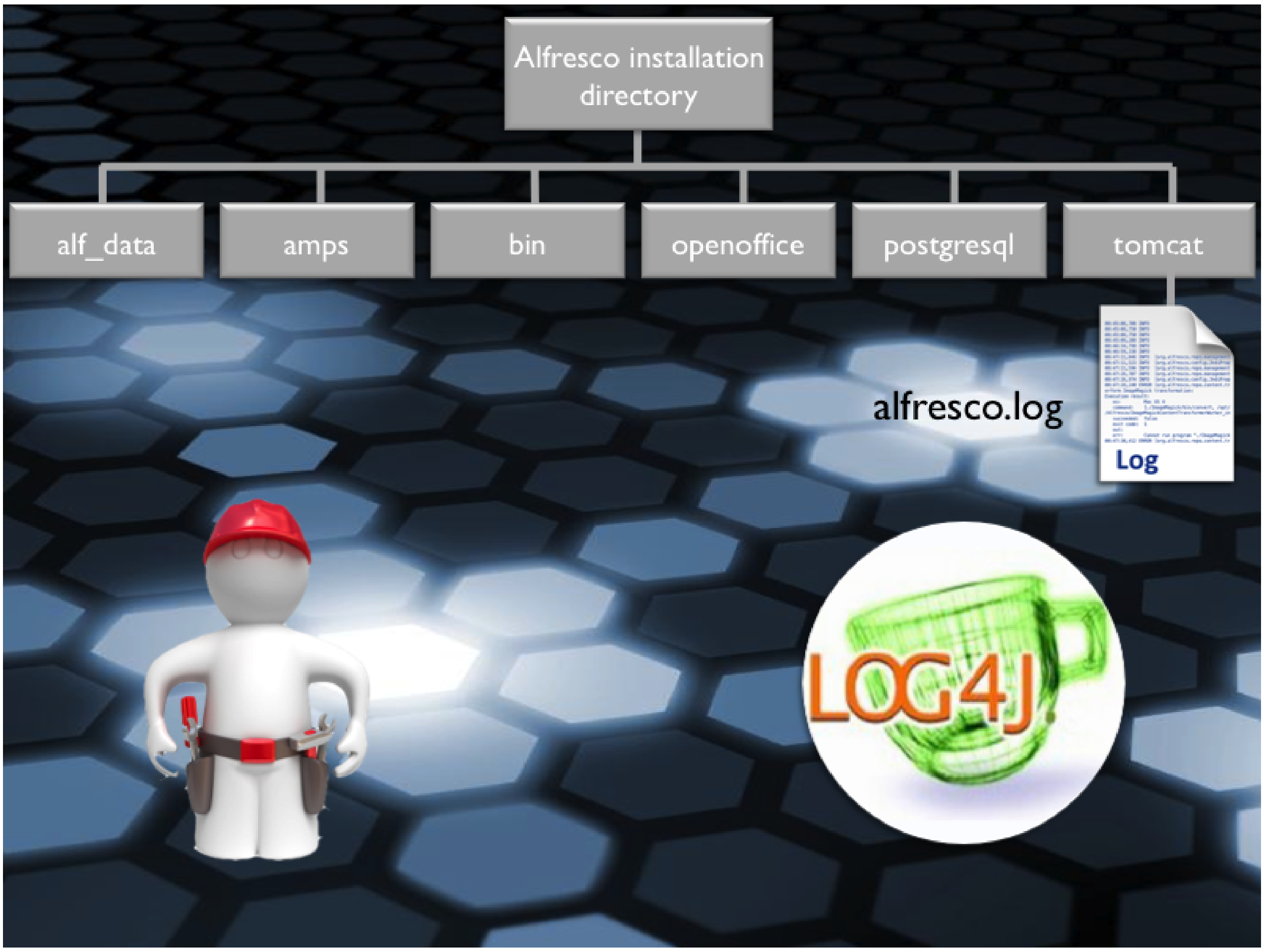
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

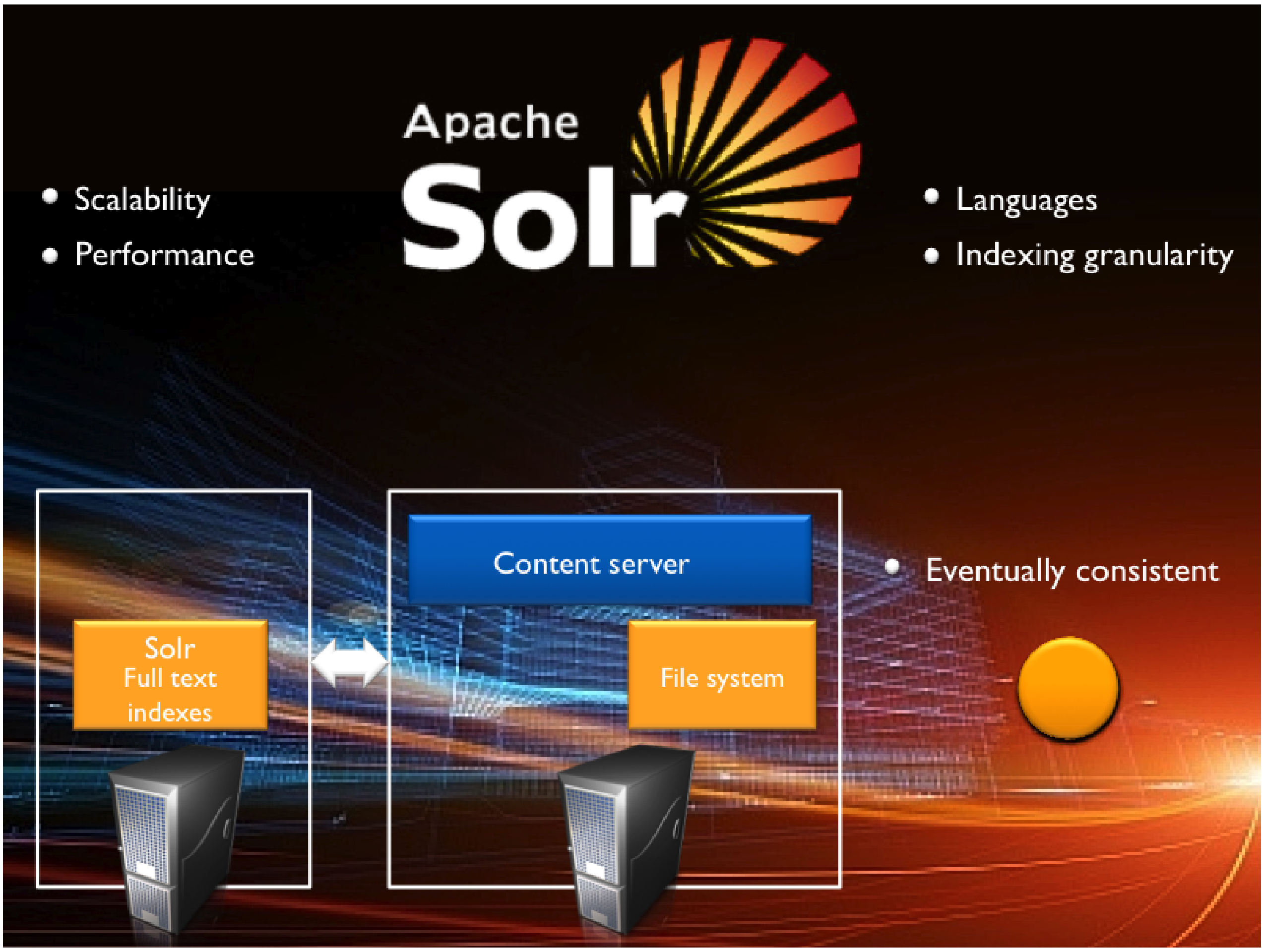
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

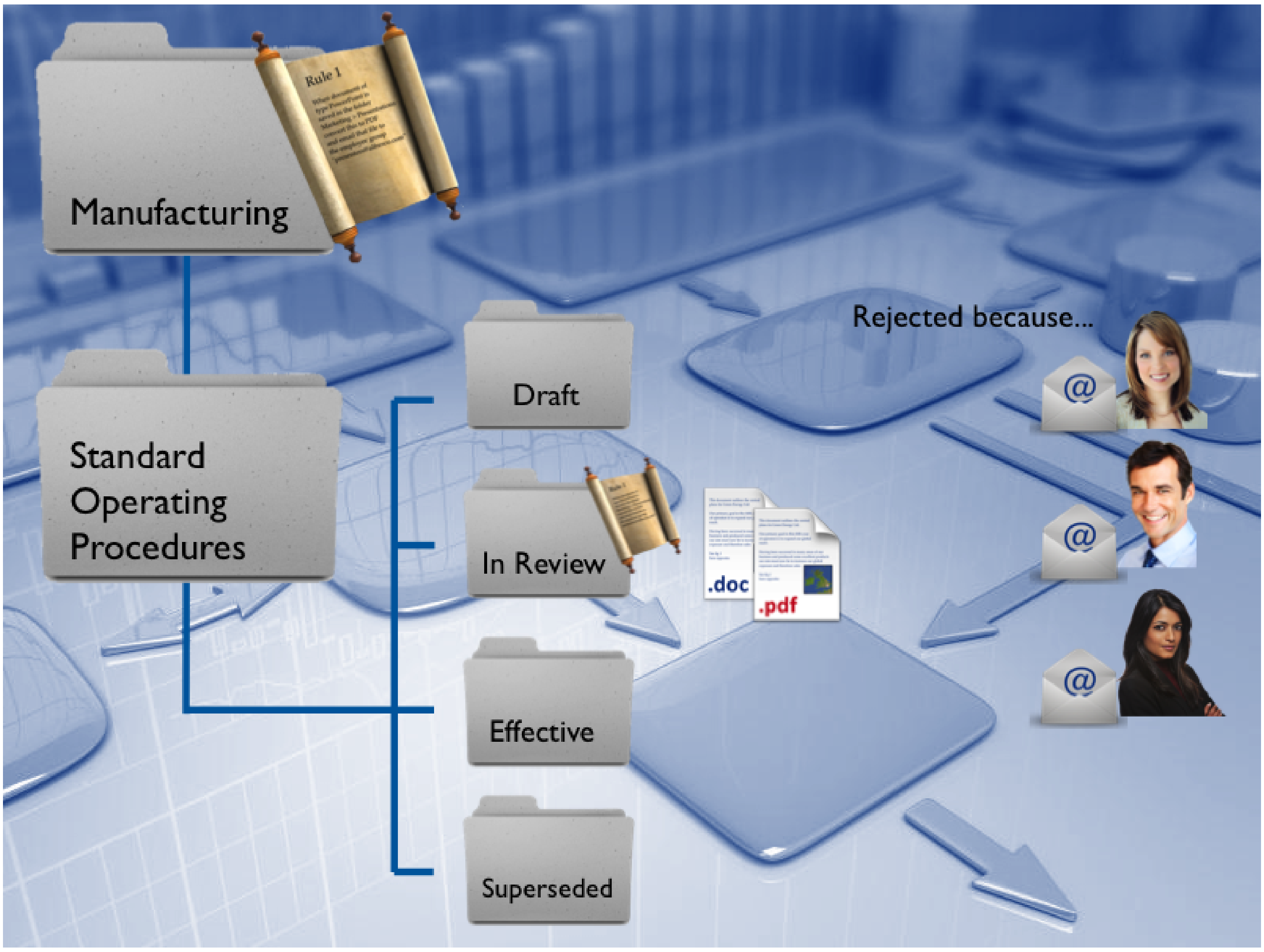
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

.amp
Alfresco module package
XML
.img
.css

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

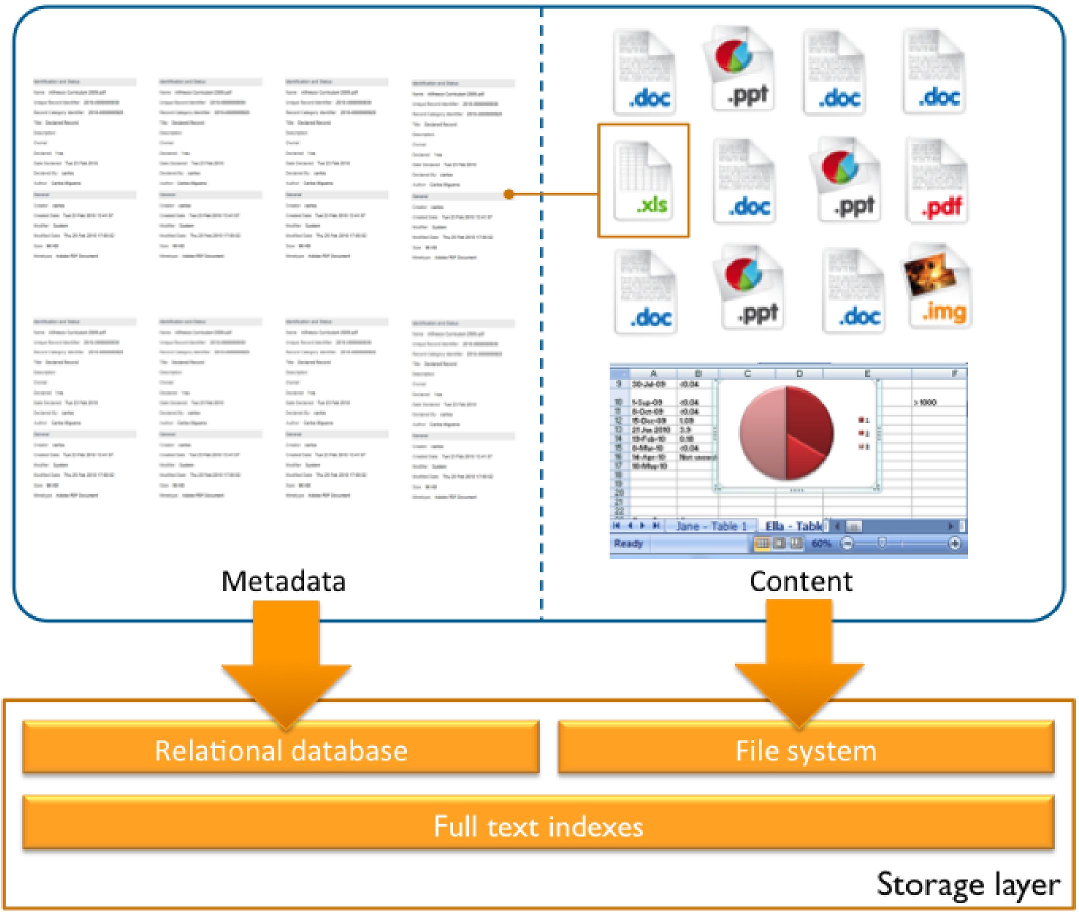
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

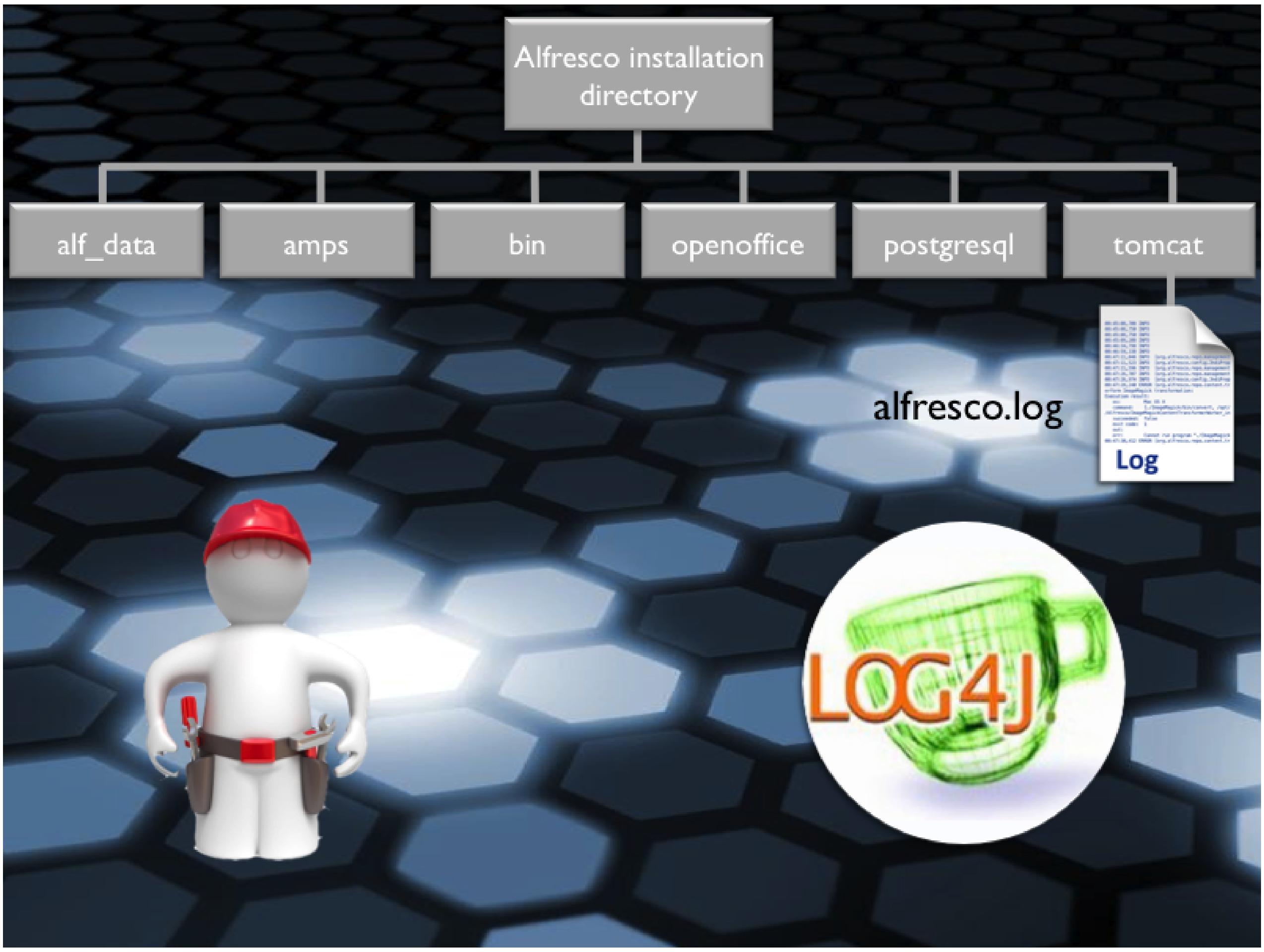
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

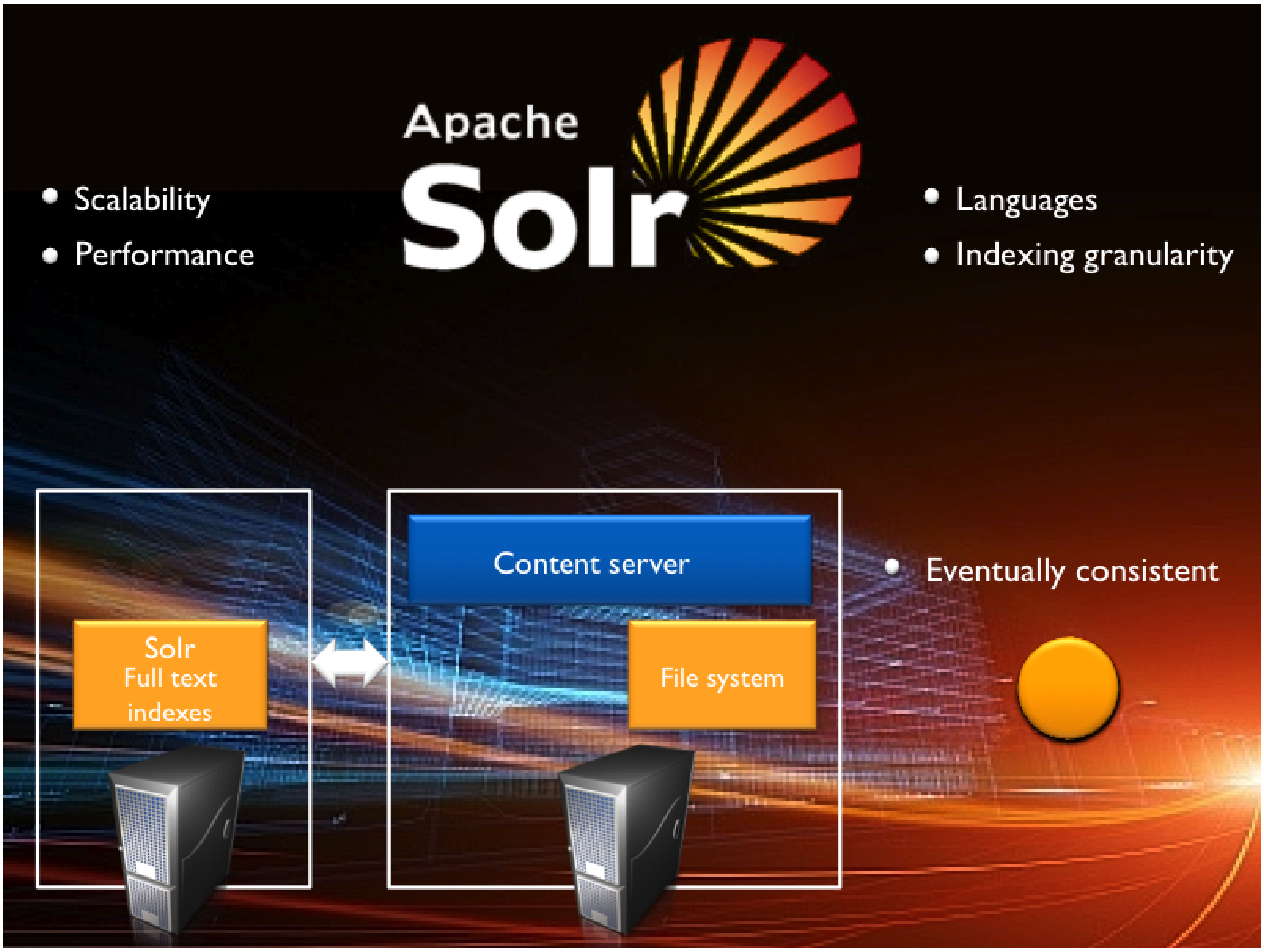
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

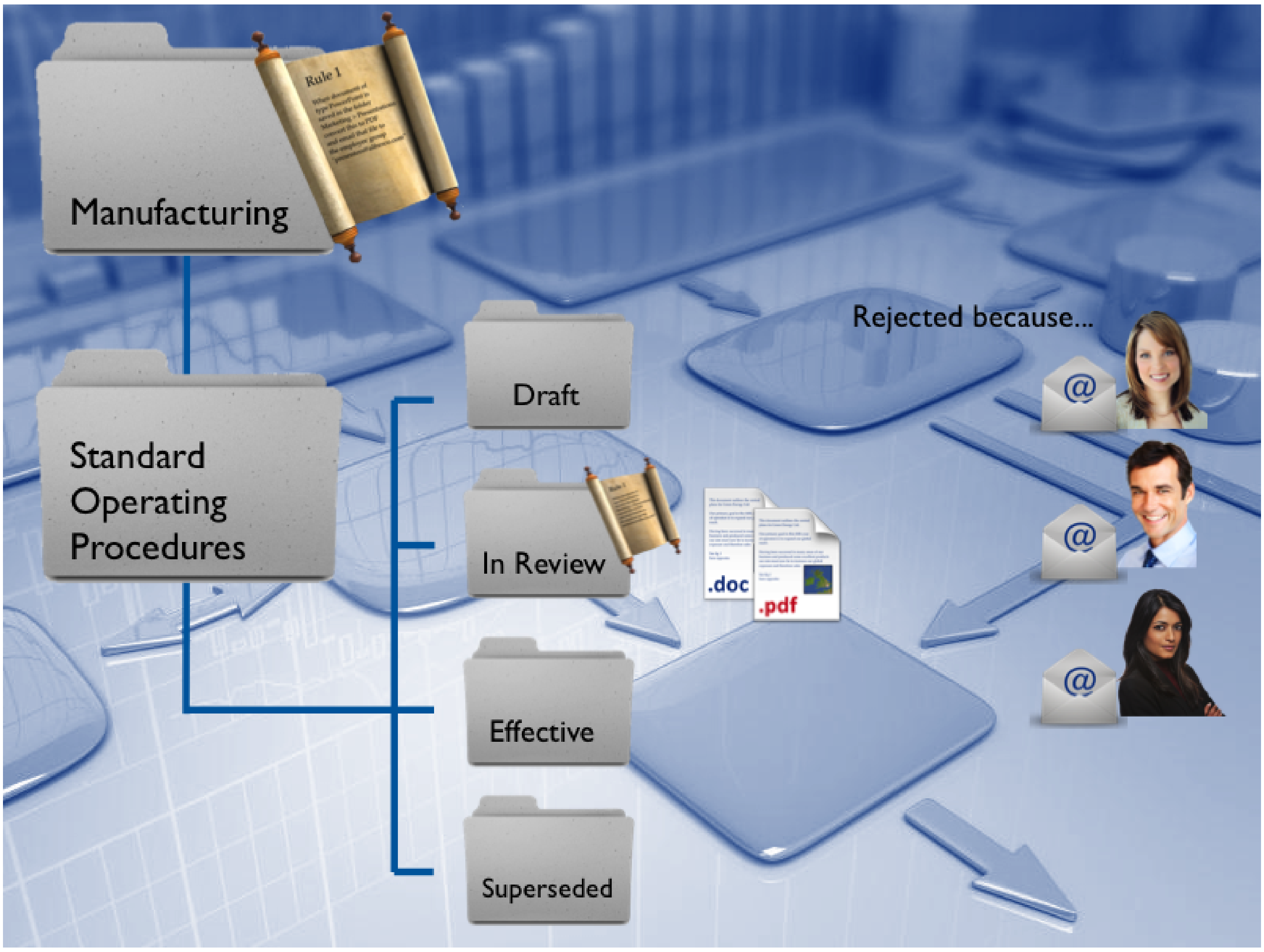
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.
    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco Elements

Managing the Repository

Contents

Document information

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The alfresco and share WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within ~tomcat/webapps/alfresco or ~tomcat/webapps/share. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let's explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the amps directory.

The second file I place in the amps_share directory.

I stop the Alfresco server.

Next I use the apply_amps command, which is found in the bin directory. This command applies the AMP files that are located in the amps and amps_share directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The apply_amps command has copied the content into the alfresco.war file. When the server starts the alfresco.war file is expanded and deployed into the ~tomcat/webapps directory, maintaining the same folder structure.

The same is true for the Share components from the alfresco-rm-share AMP file, now found in the share.war file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn’t have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

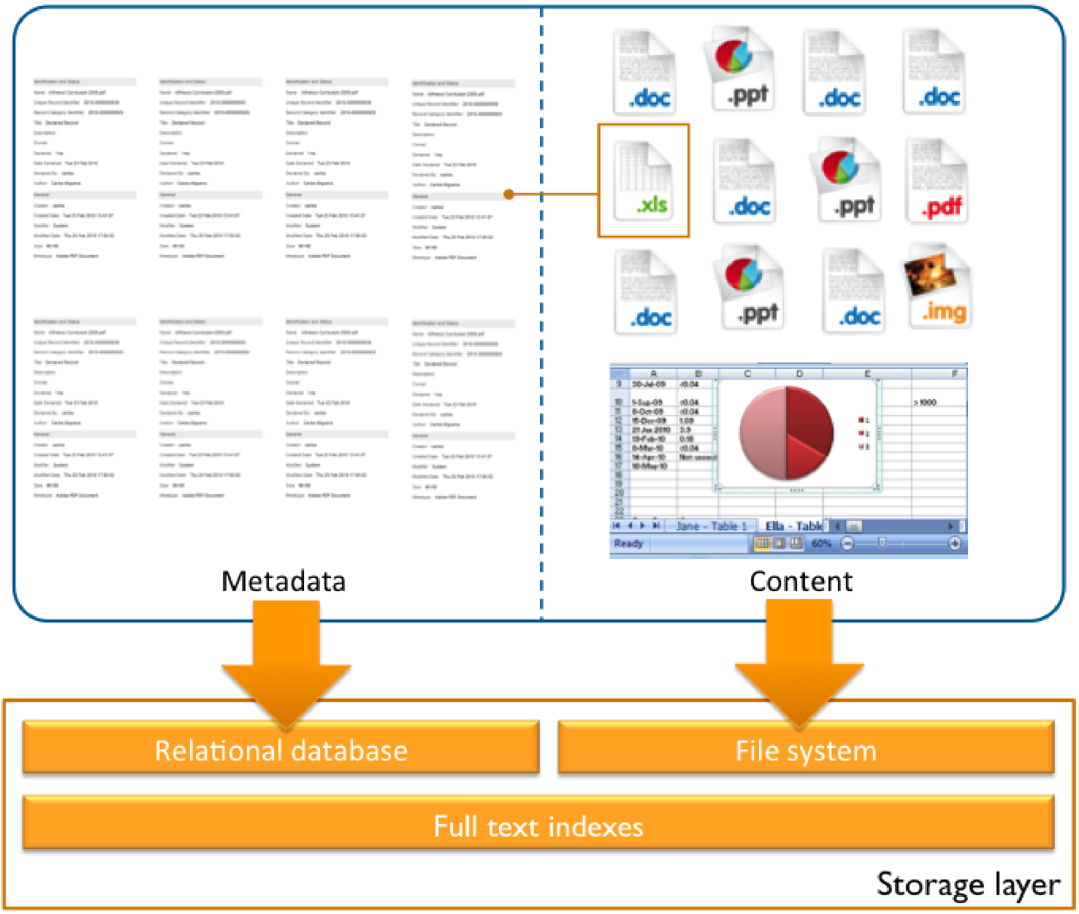
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don’t have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

        (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

        You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

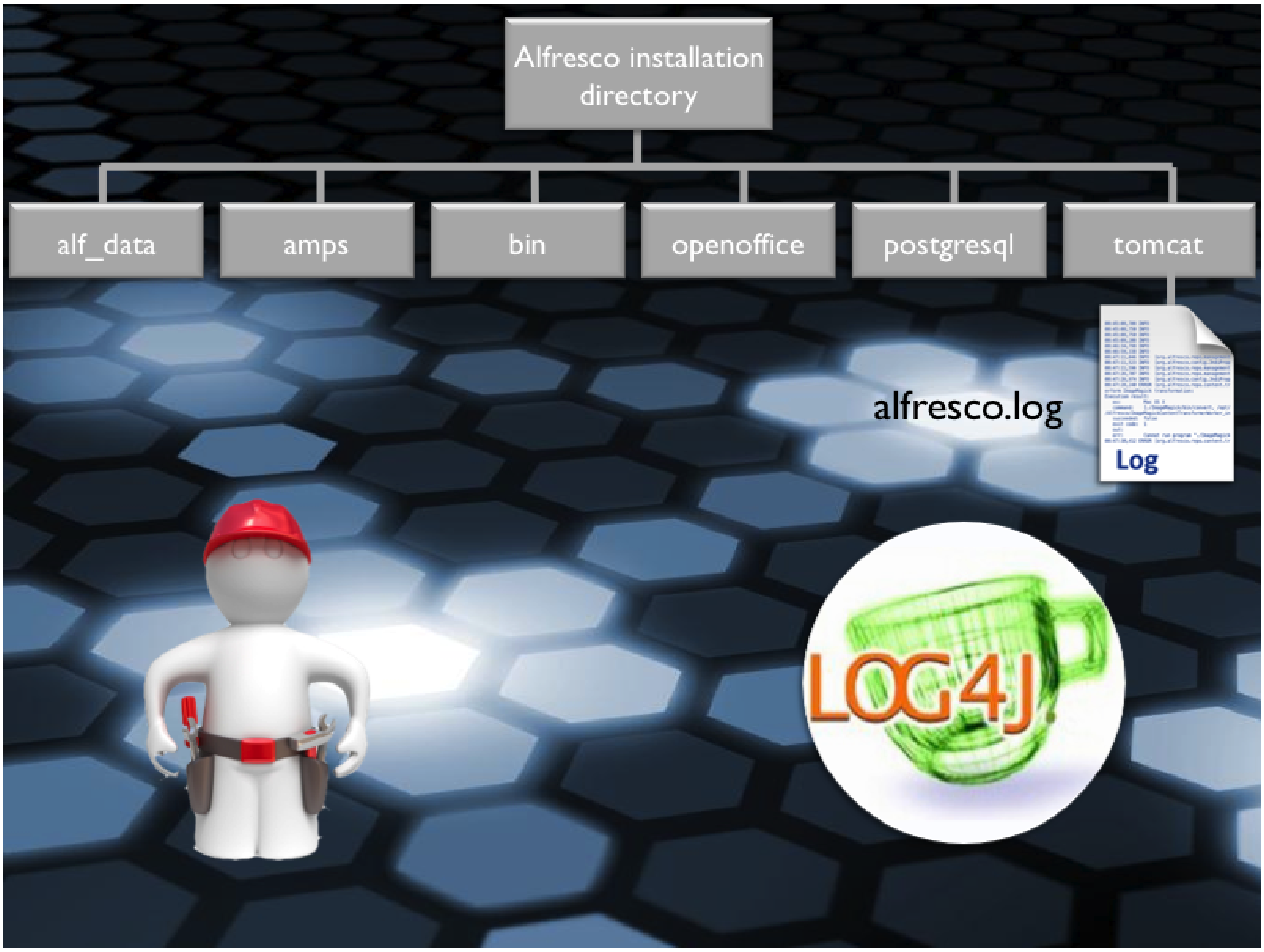
        Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

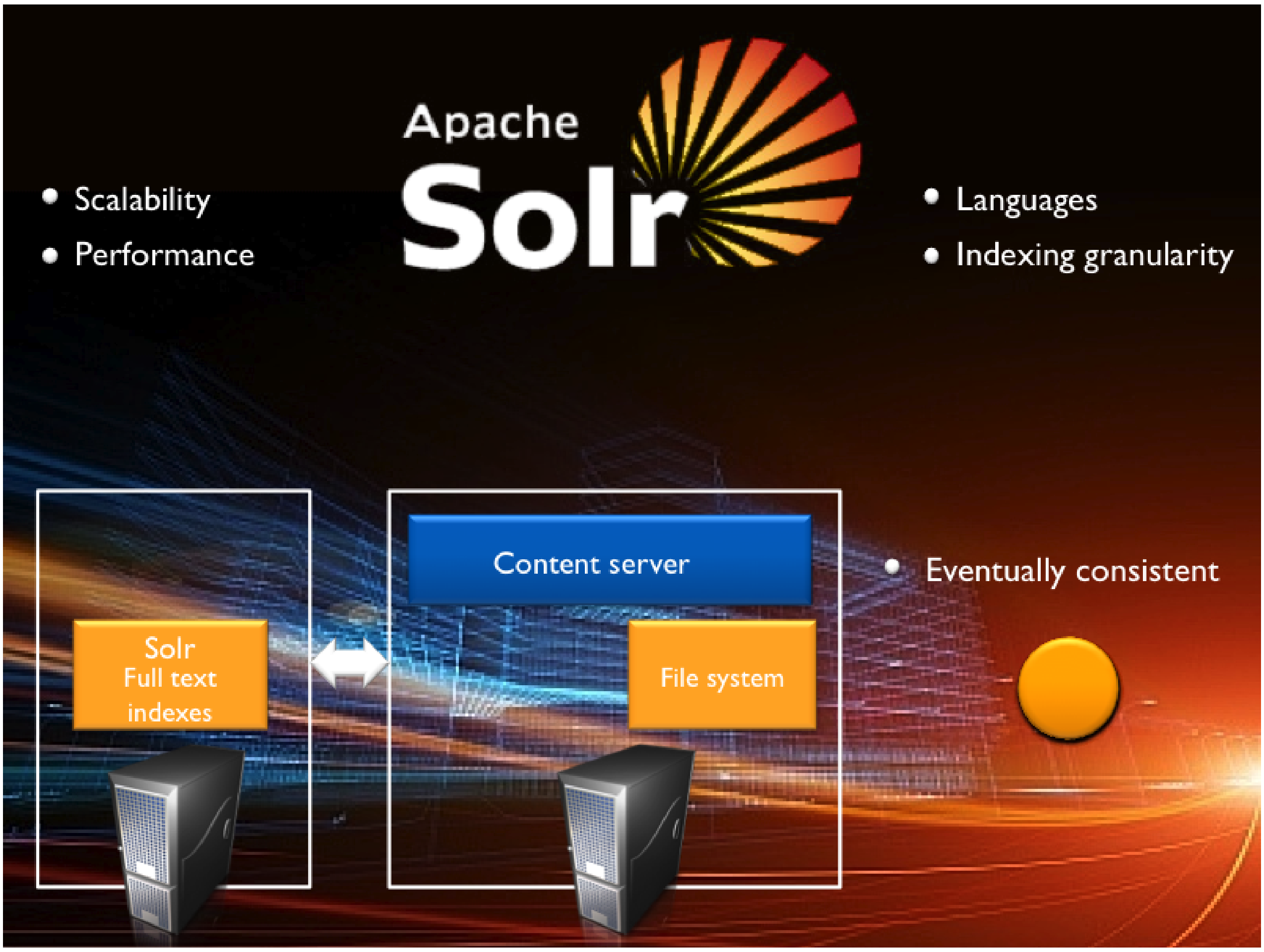
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

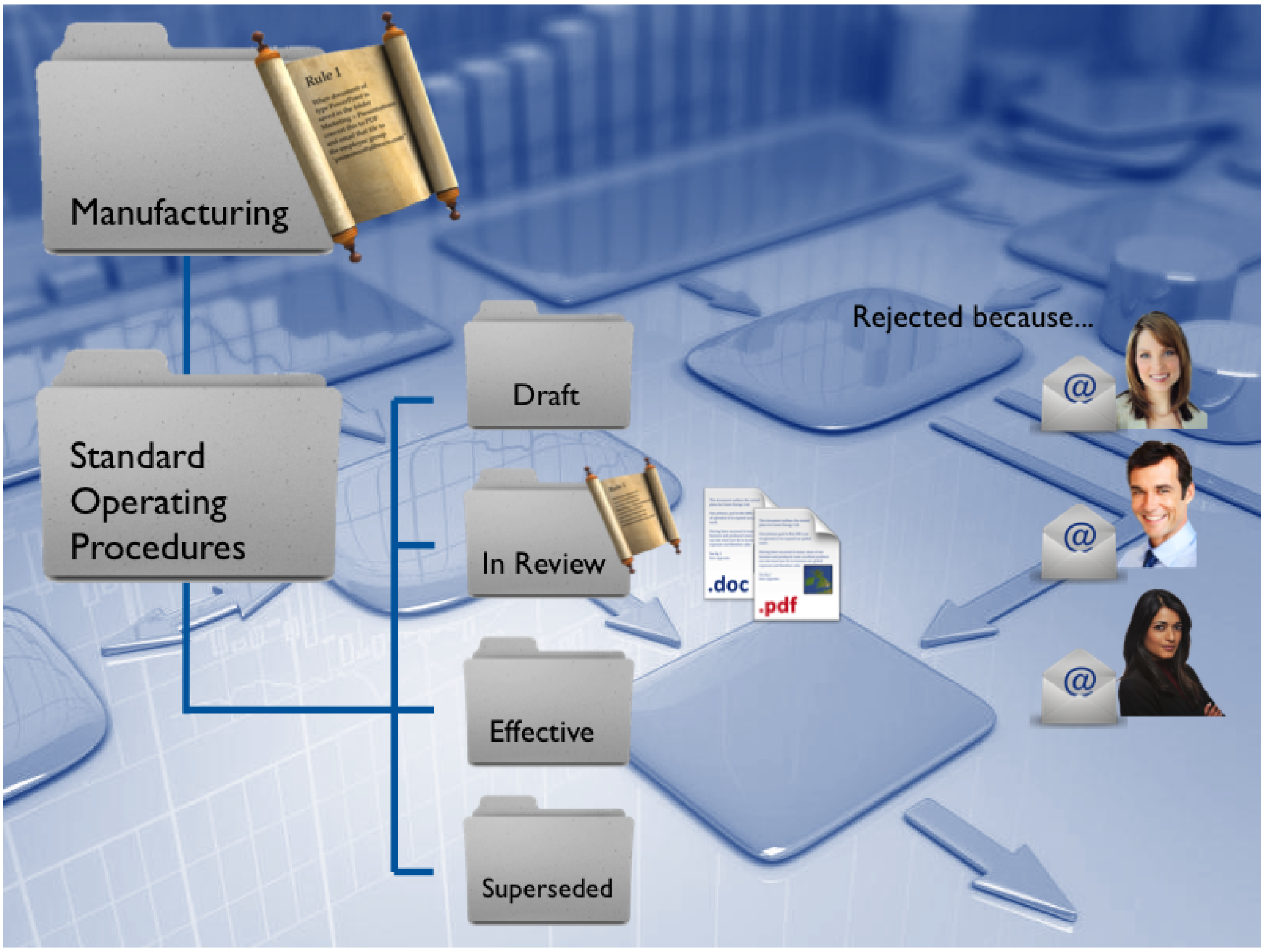
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.
    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let's explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

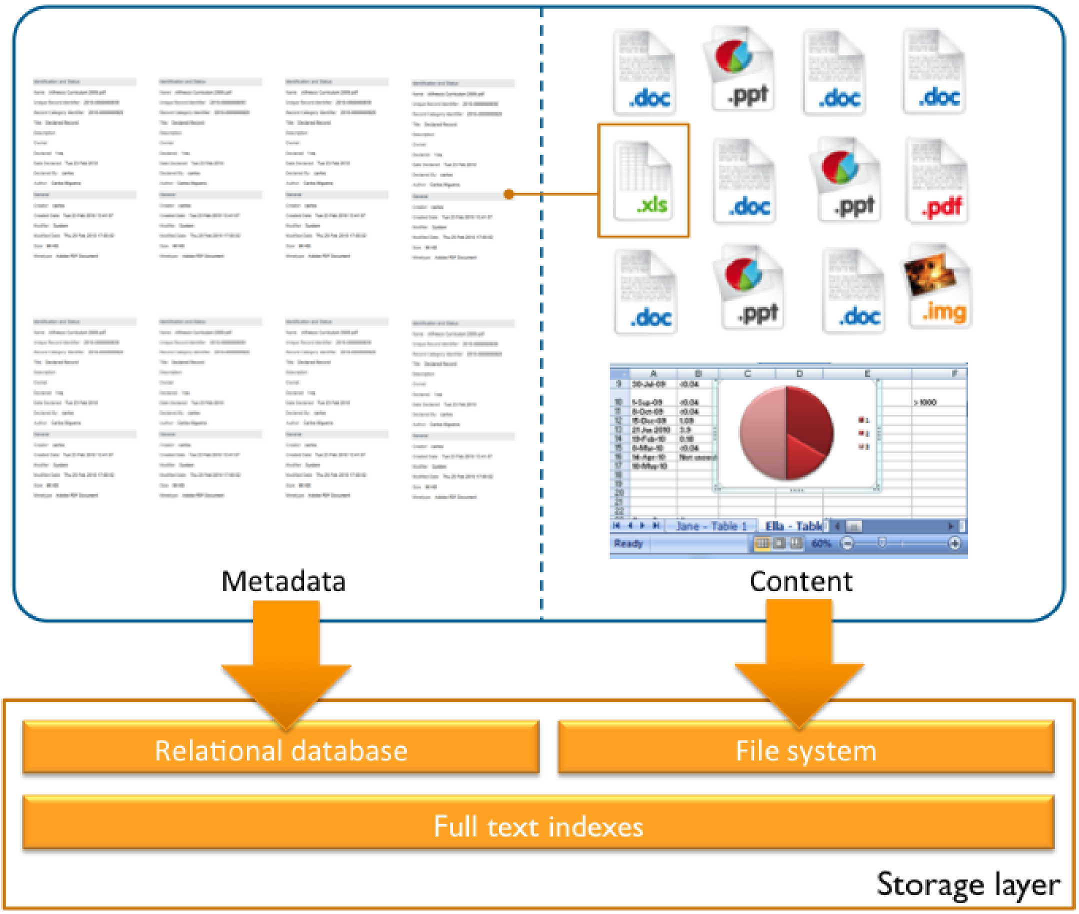
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator’s panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

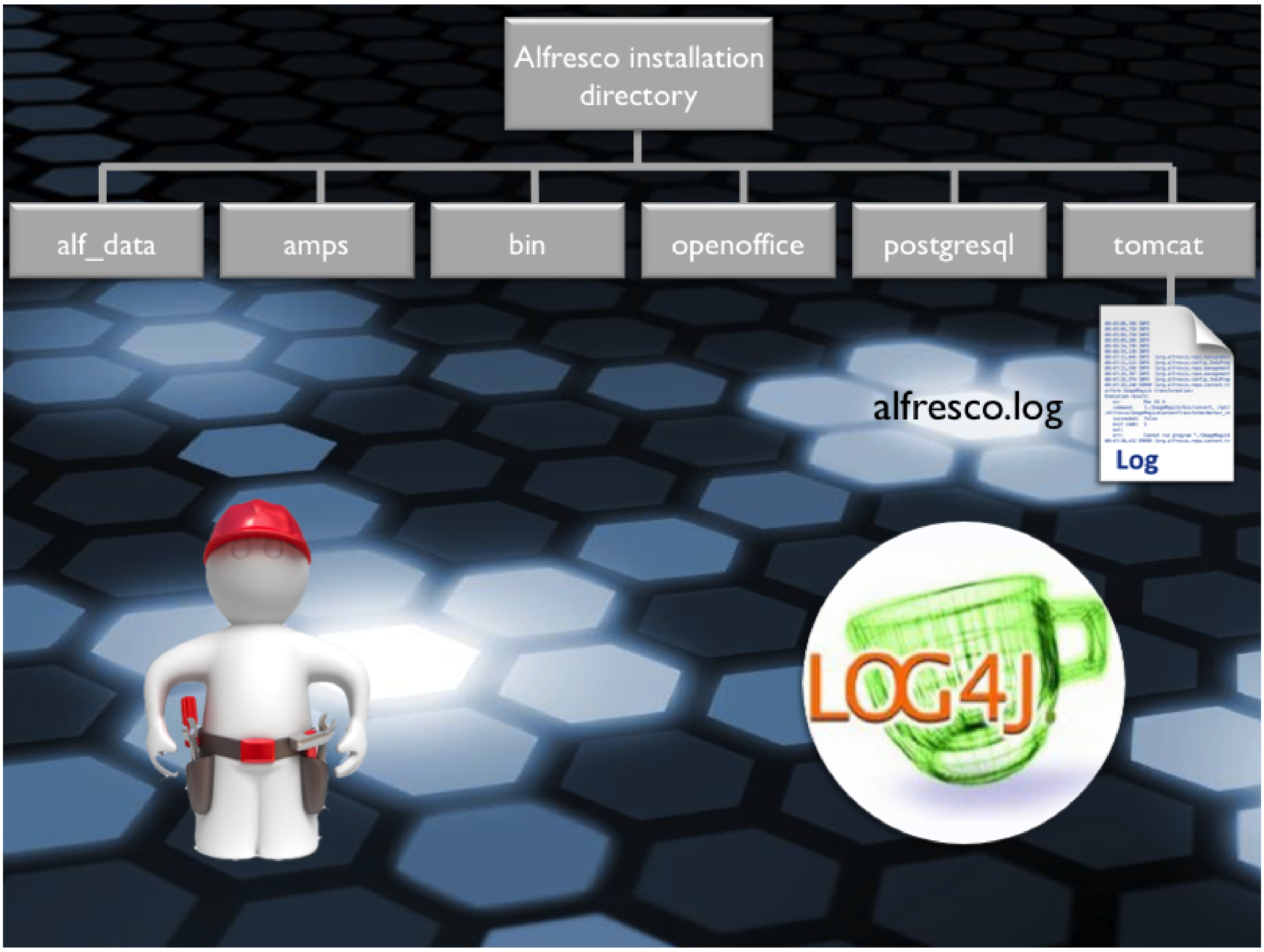
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

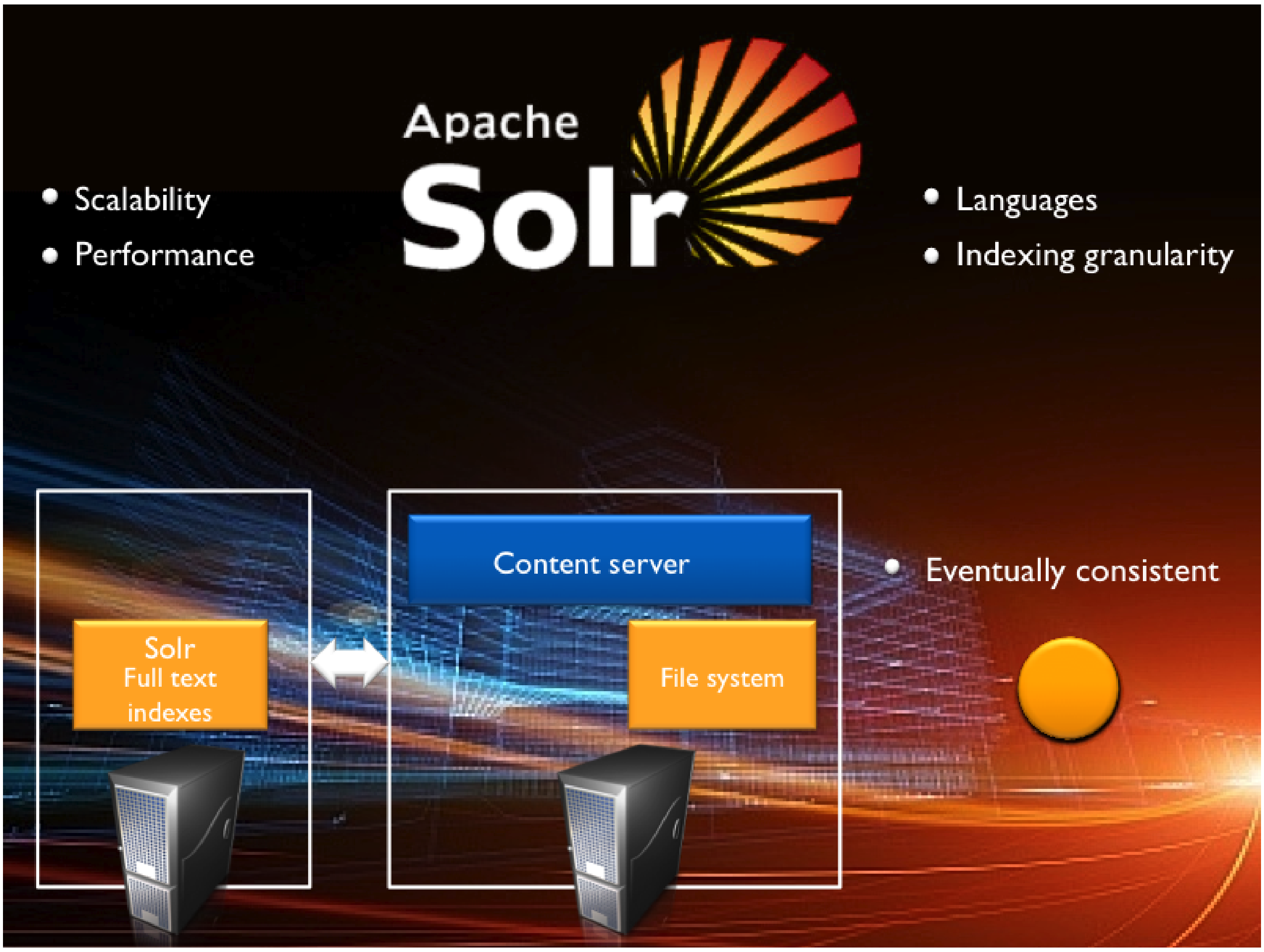
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

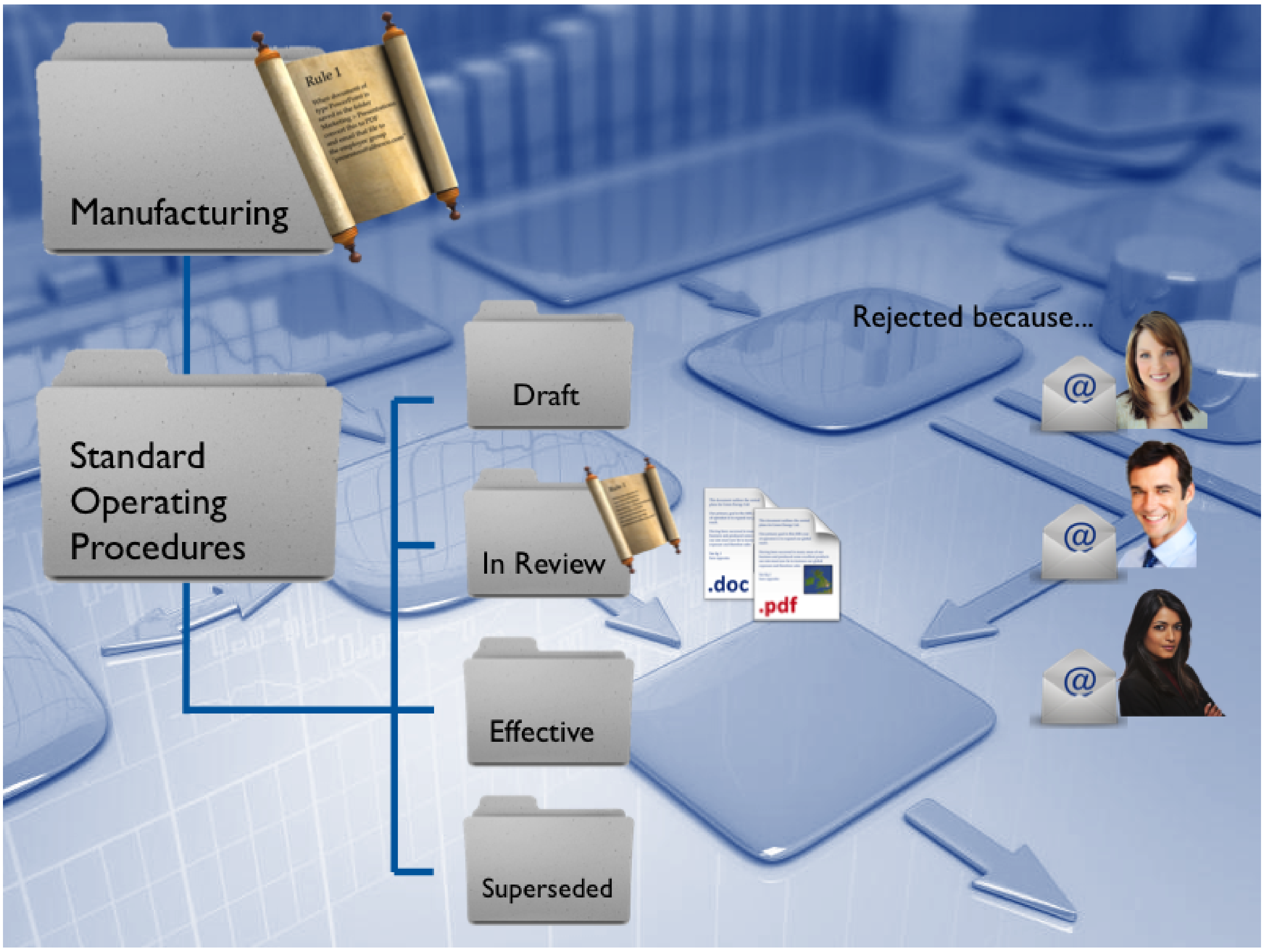
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let's explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn’t have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

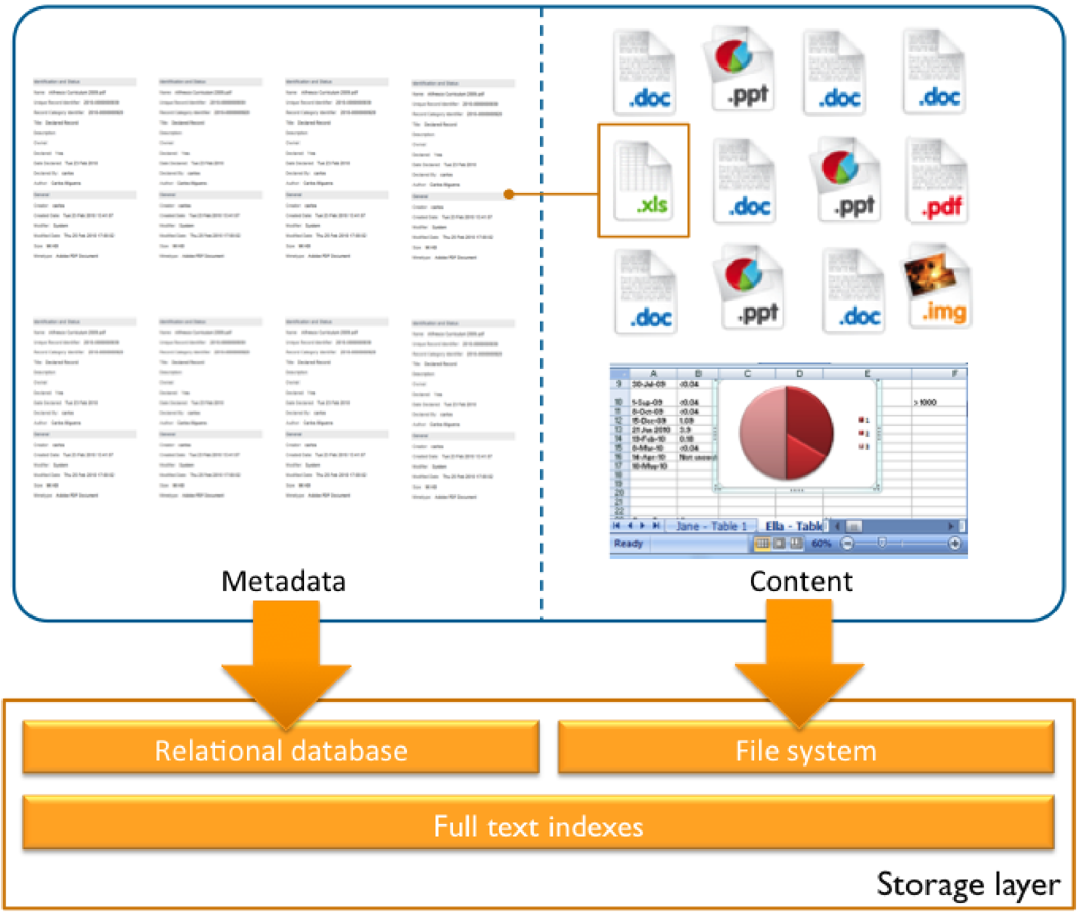
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don’t have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let's explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

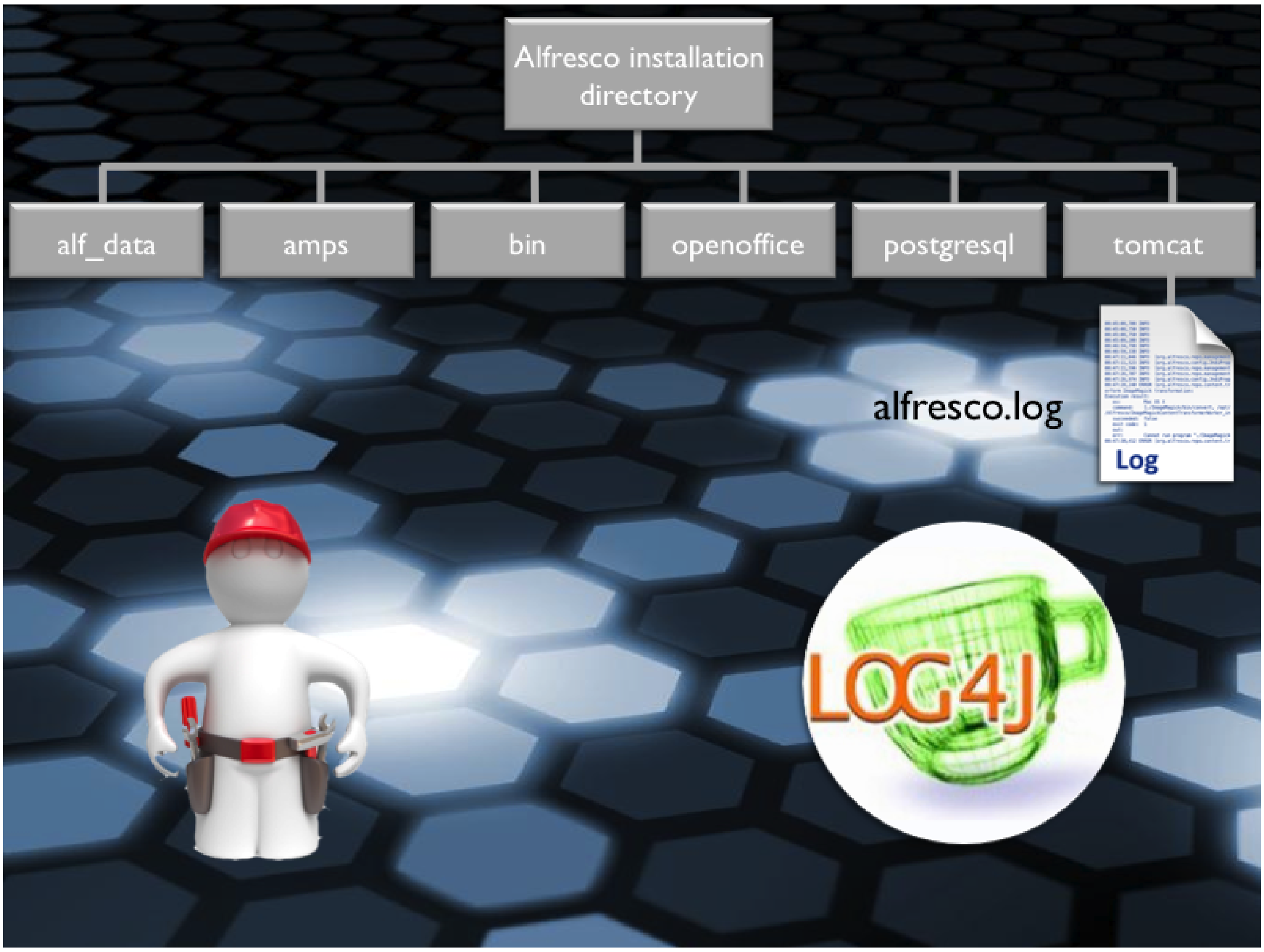
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

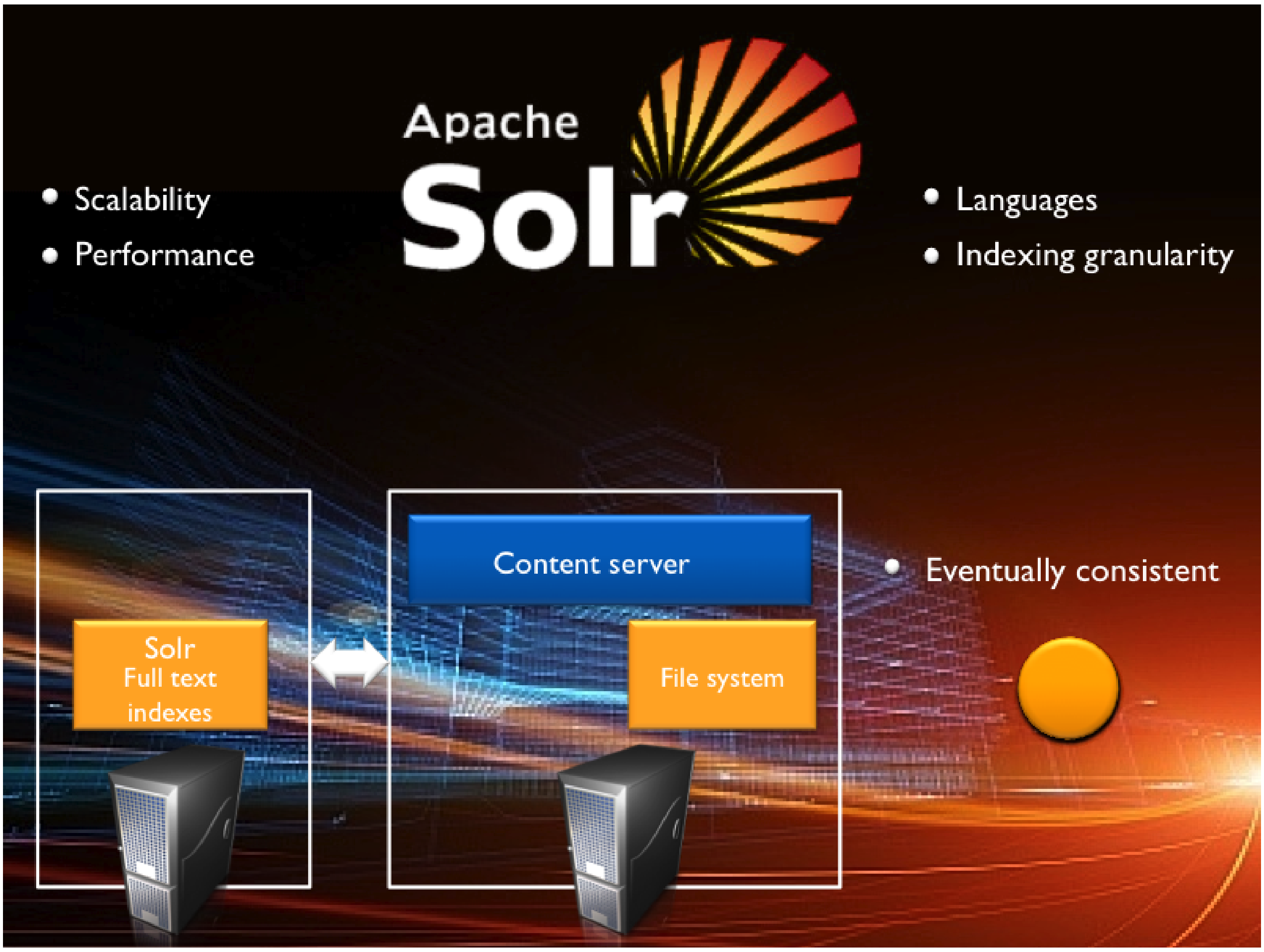
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

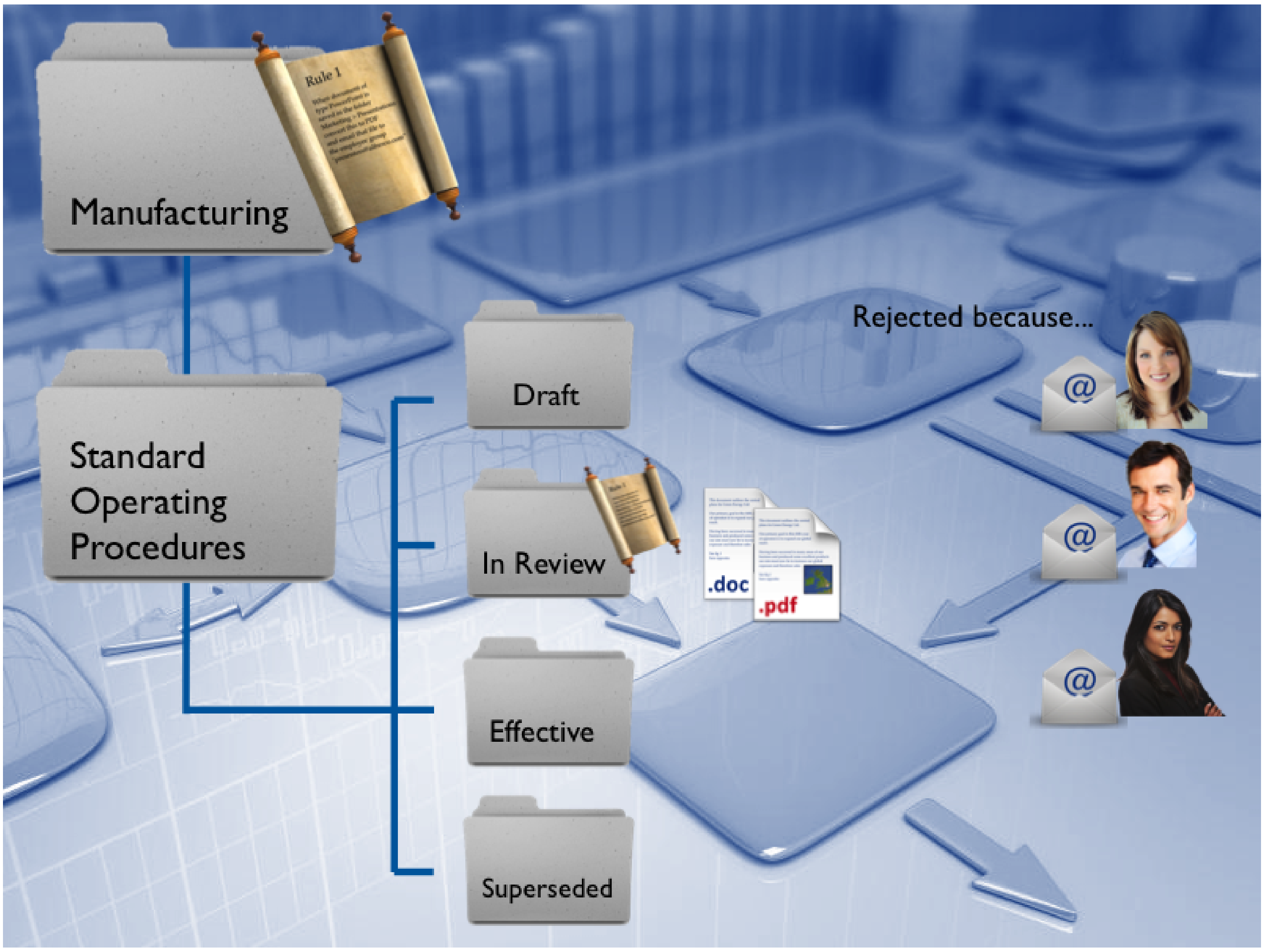
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.
    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn’t have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

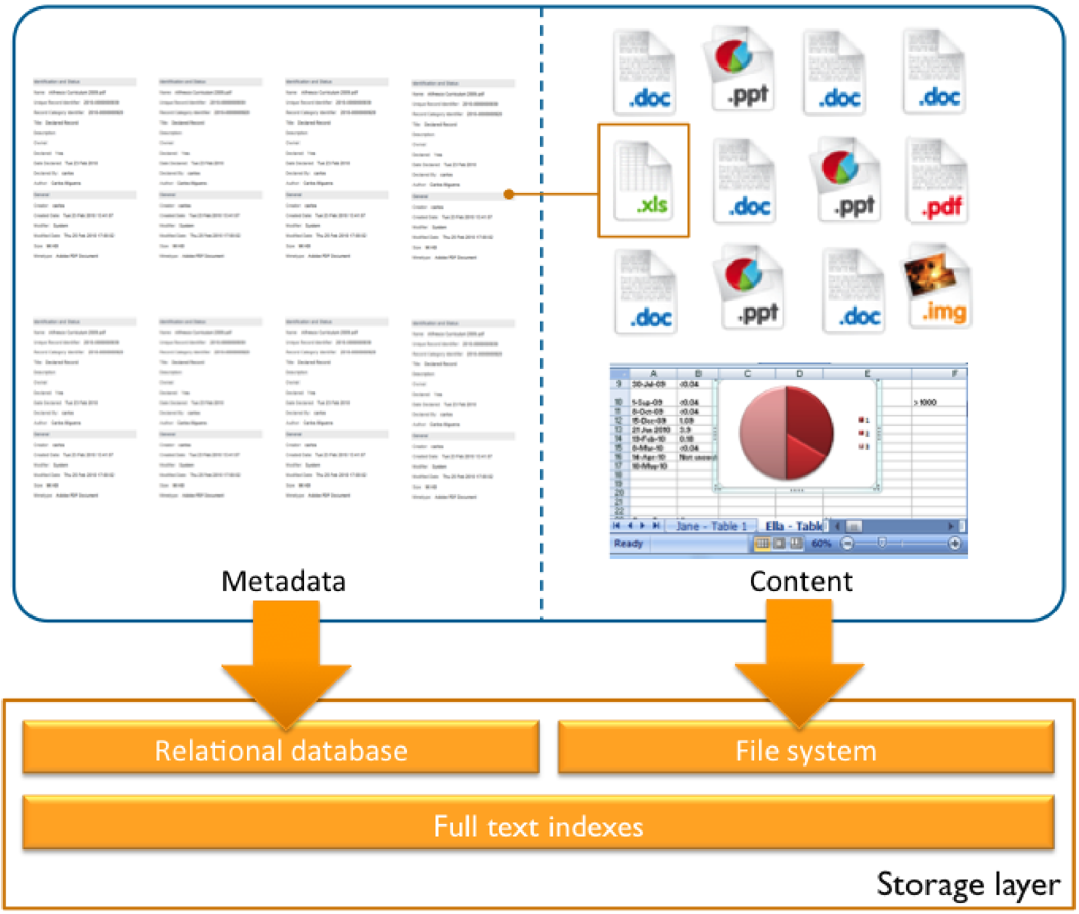
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don’t have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

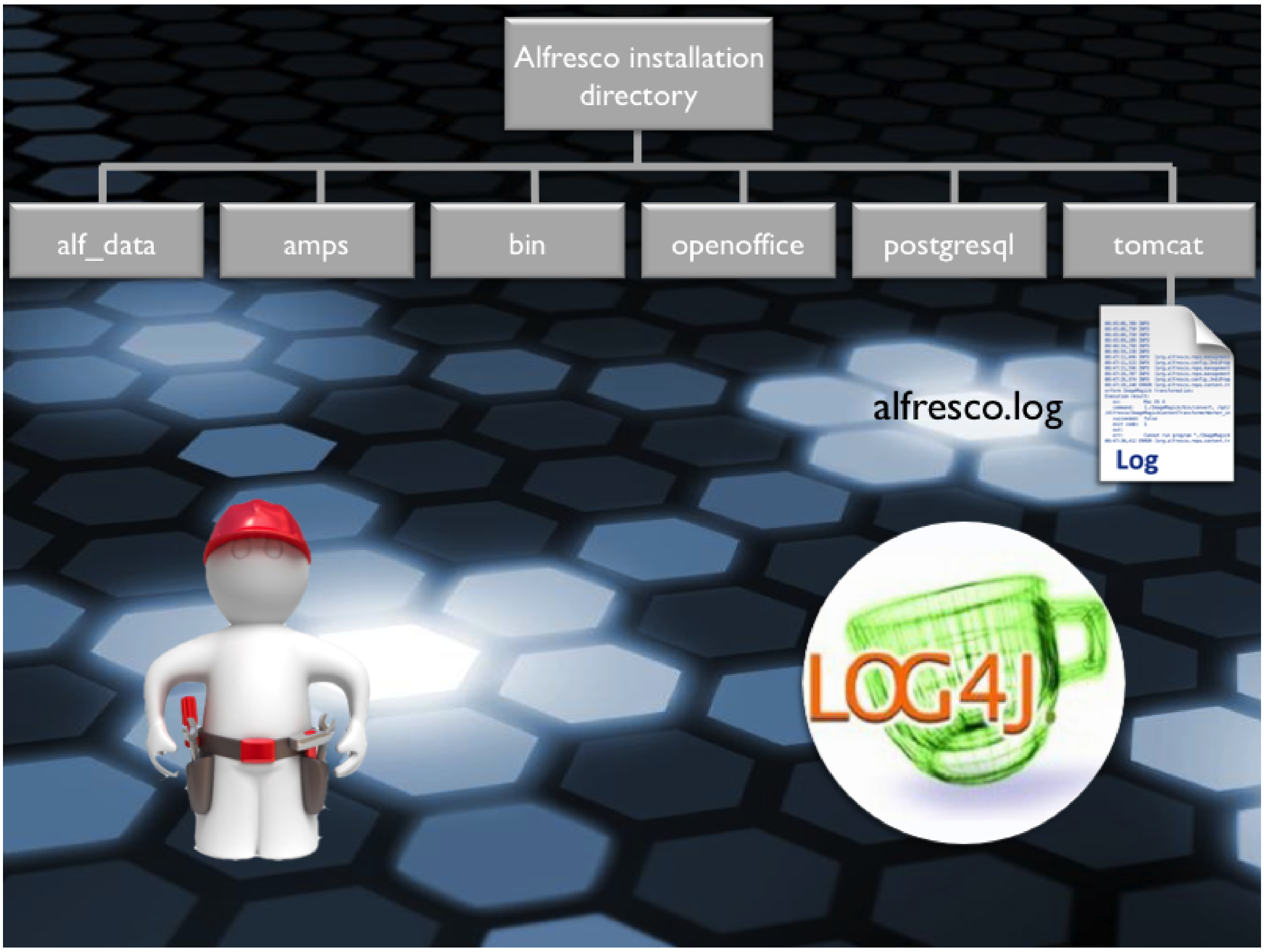
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

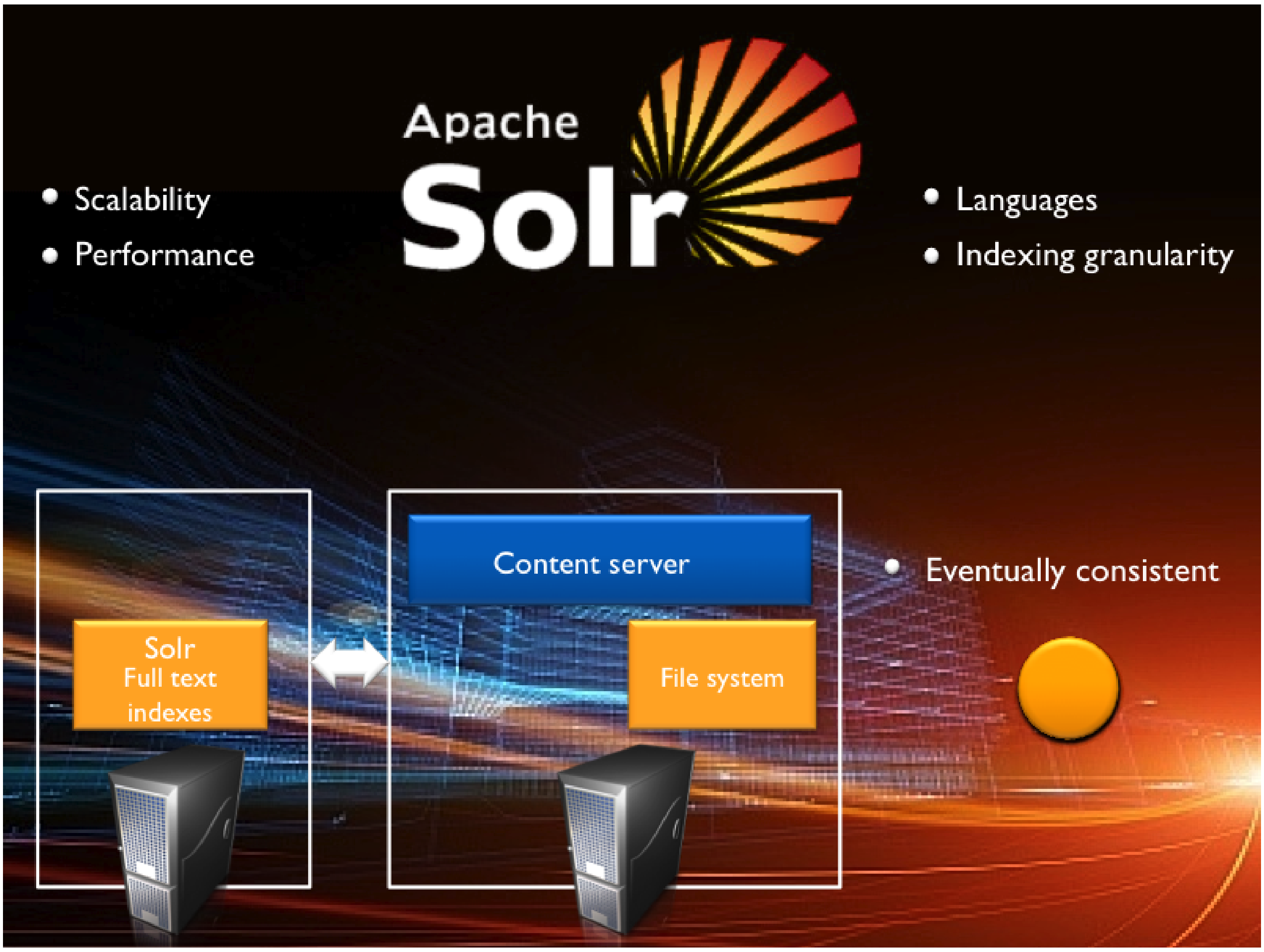
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

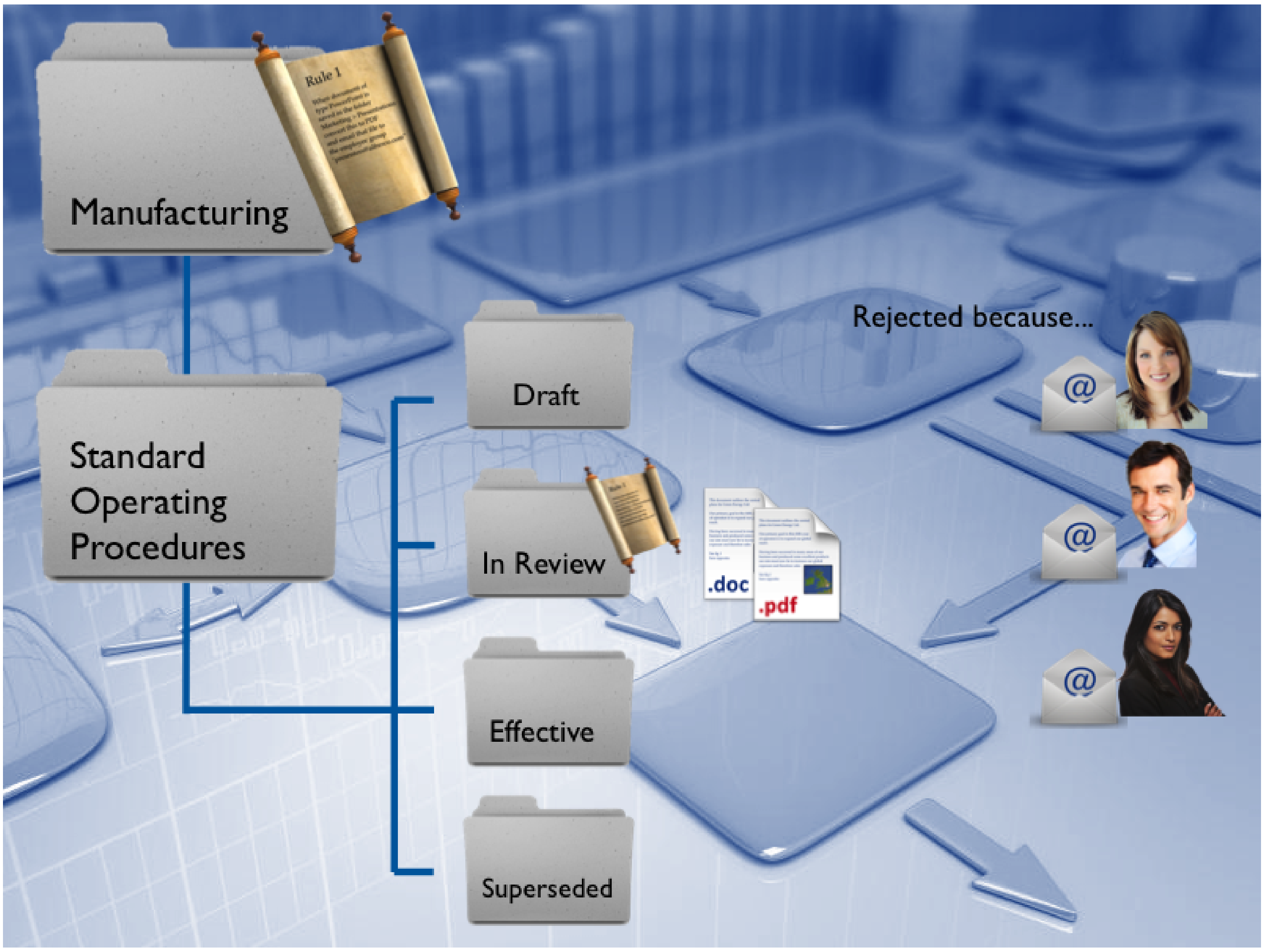
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.
    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

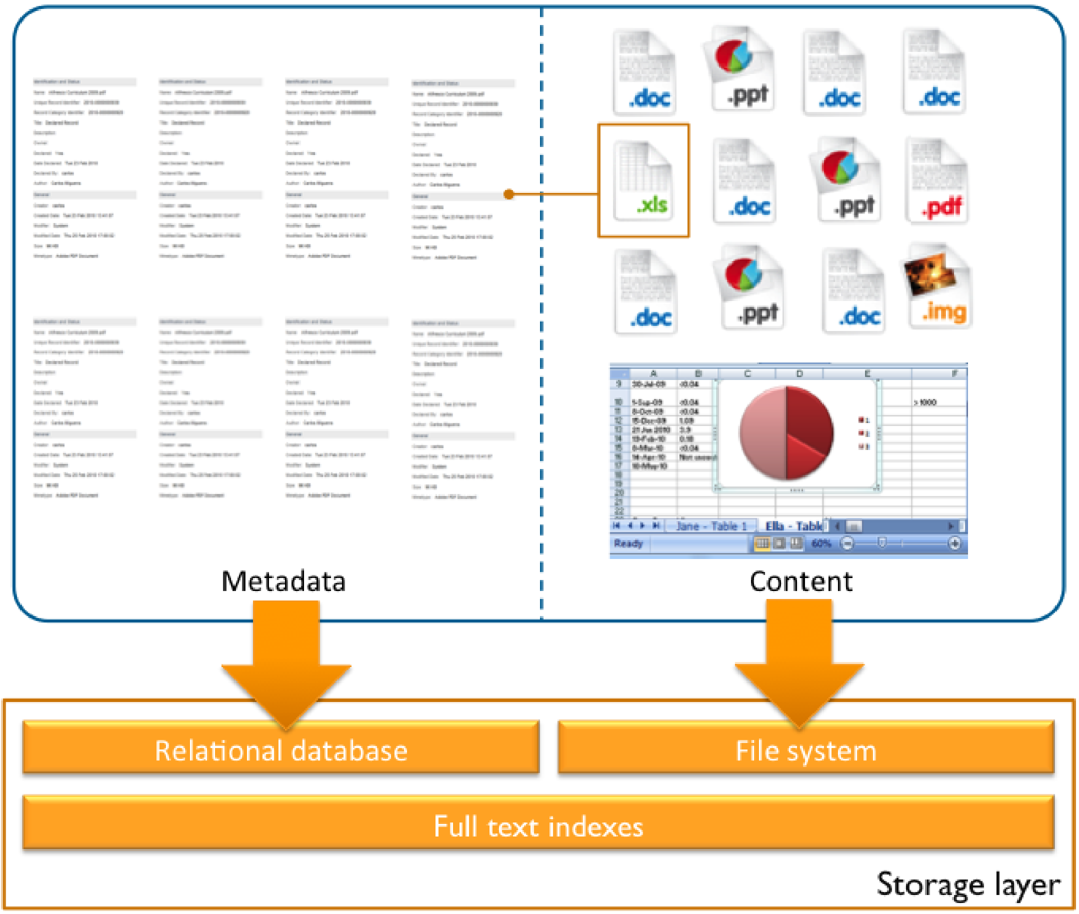
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

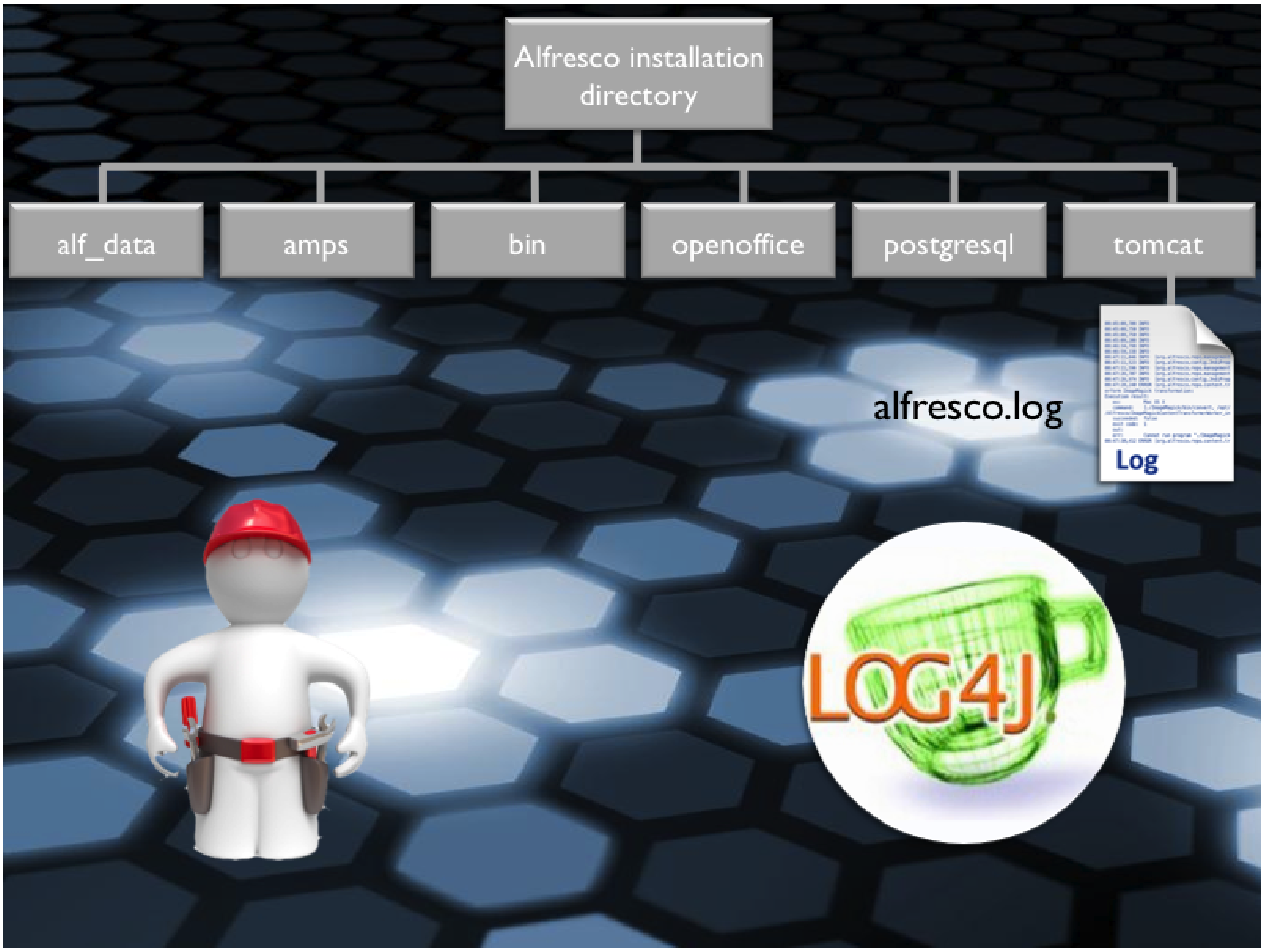
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

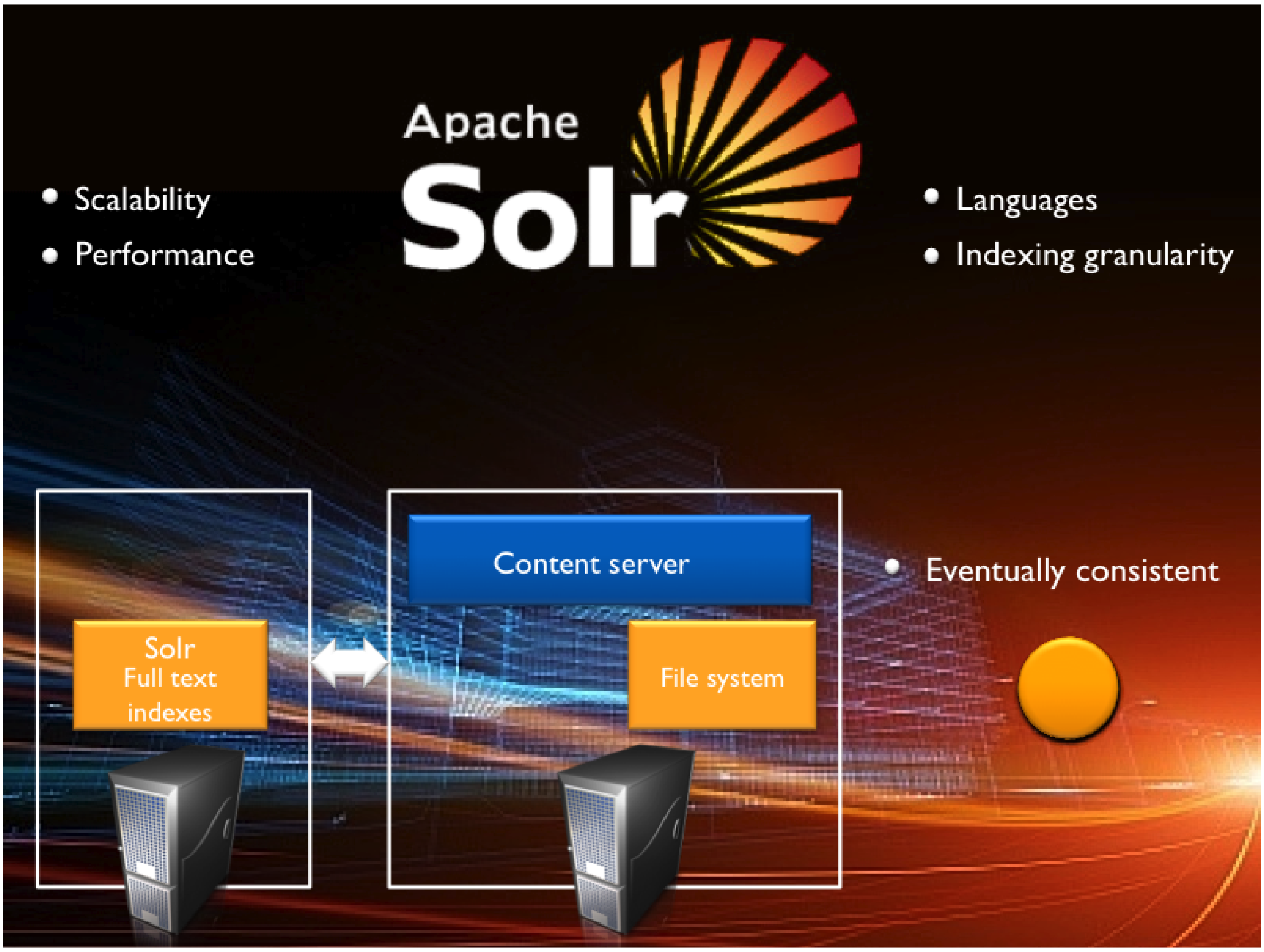
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

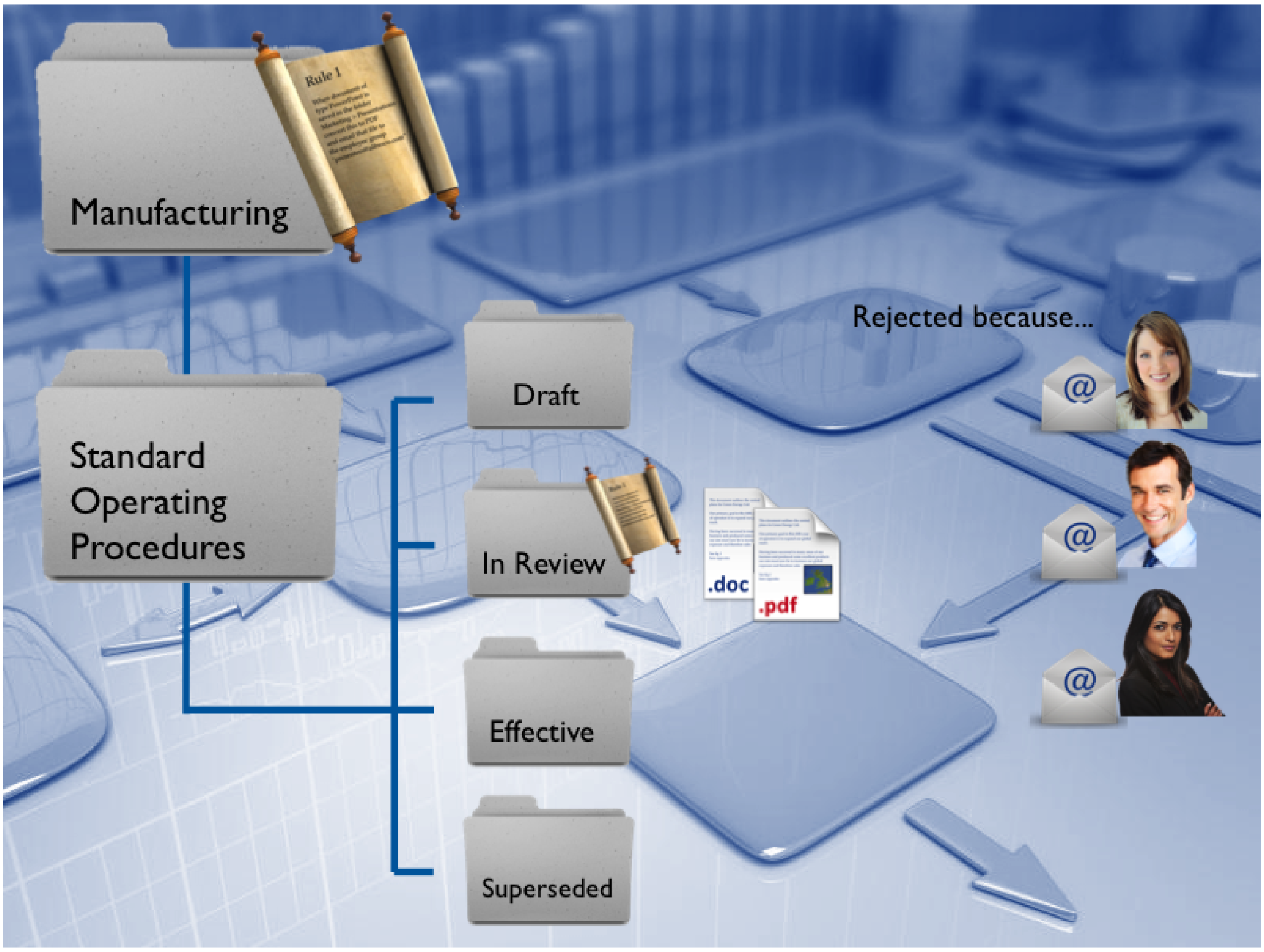
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn’t have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

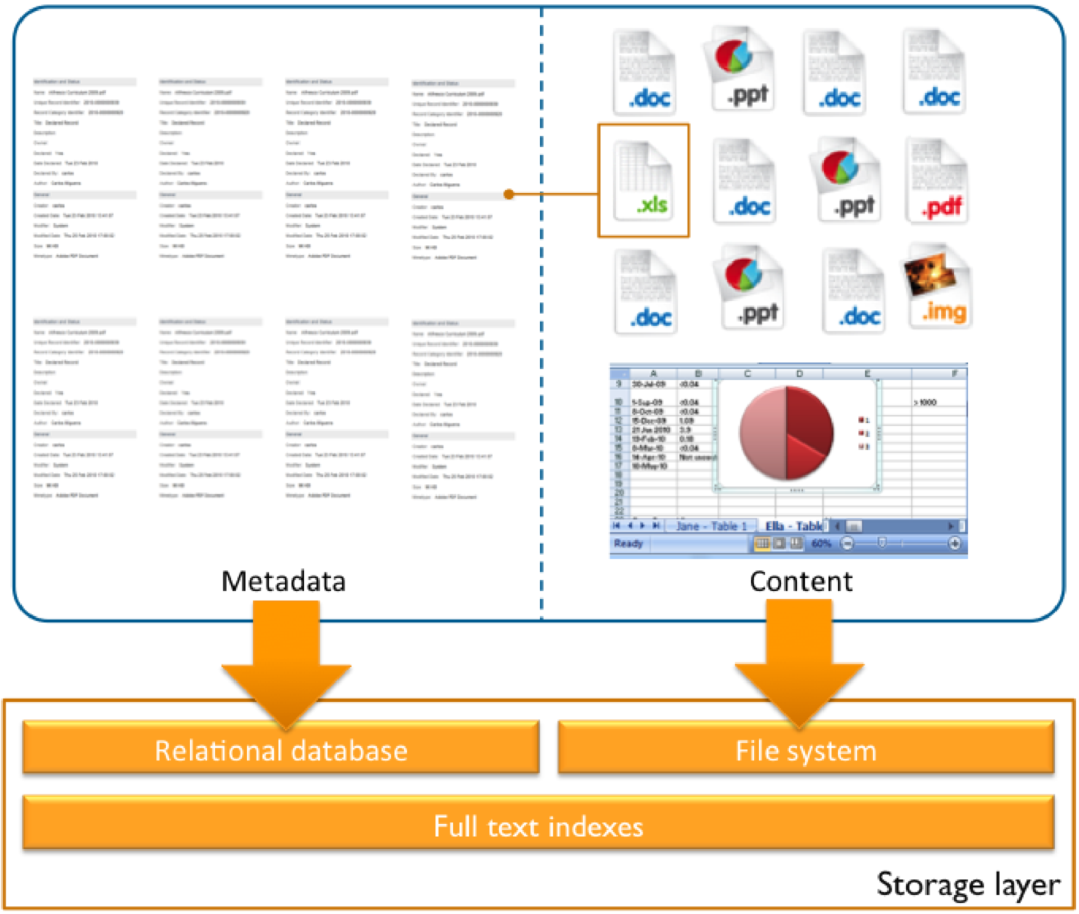
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don’t have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

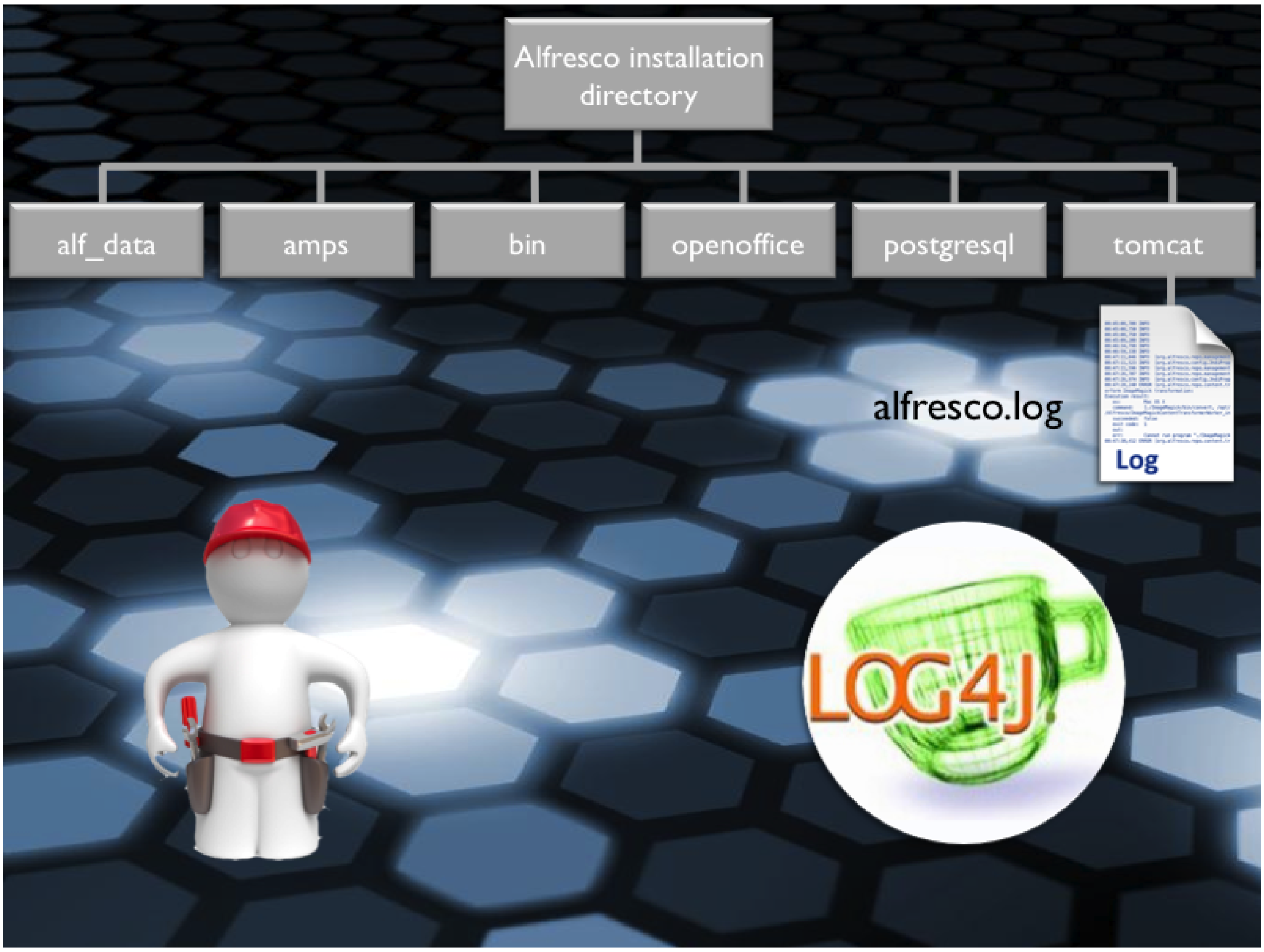
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

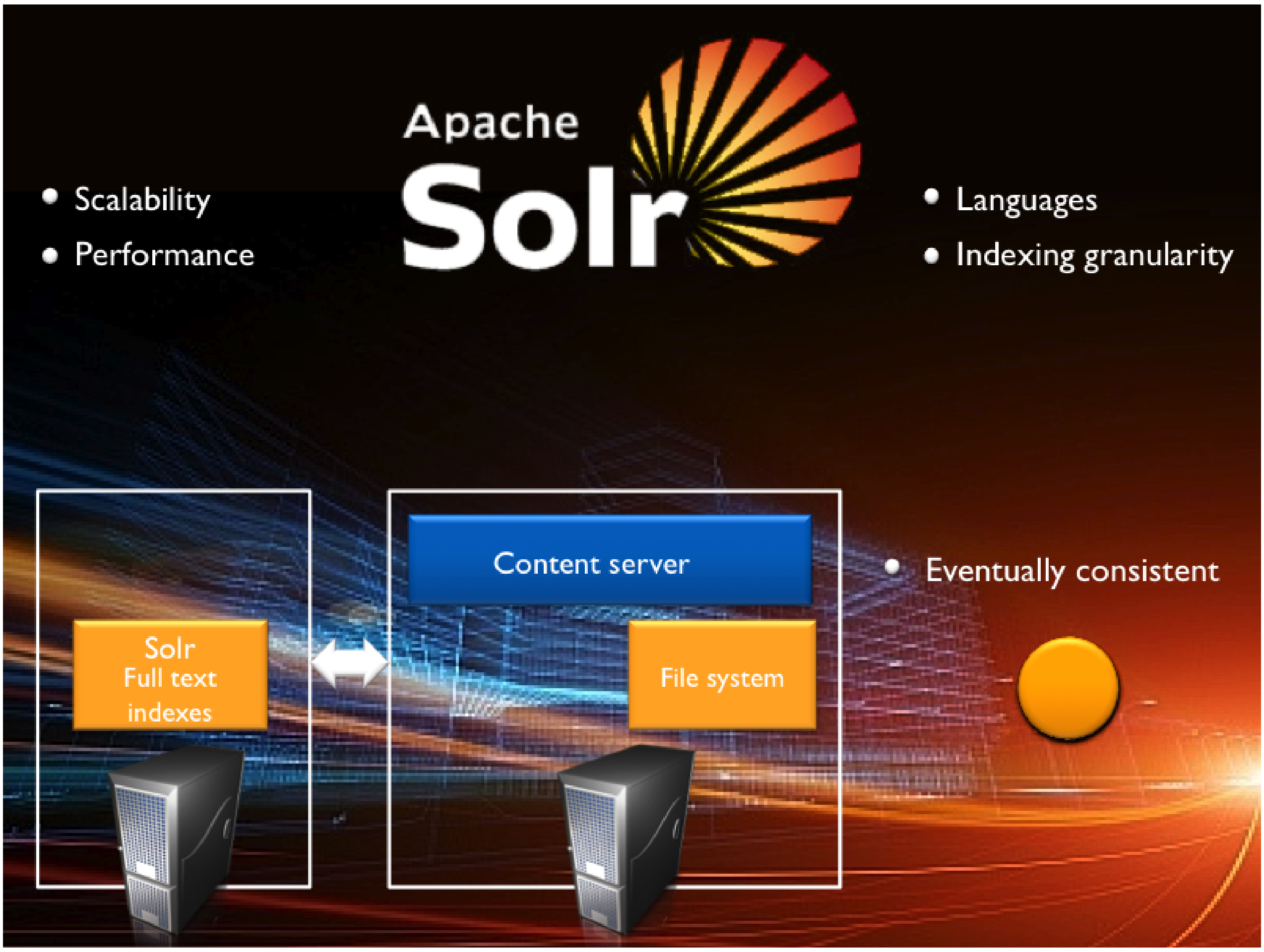
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

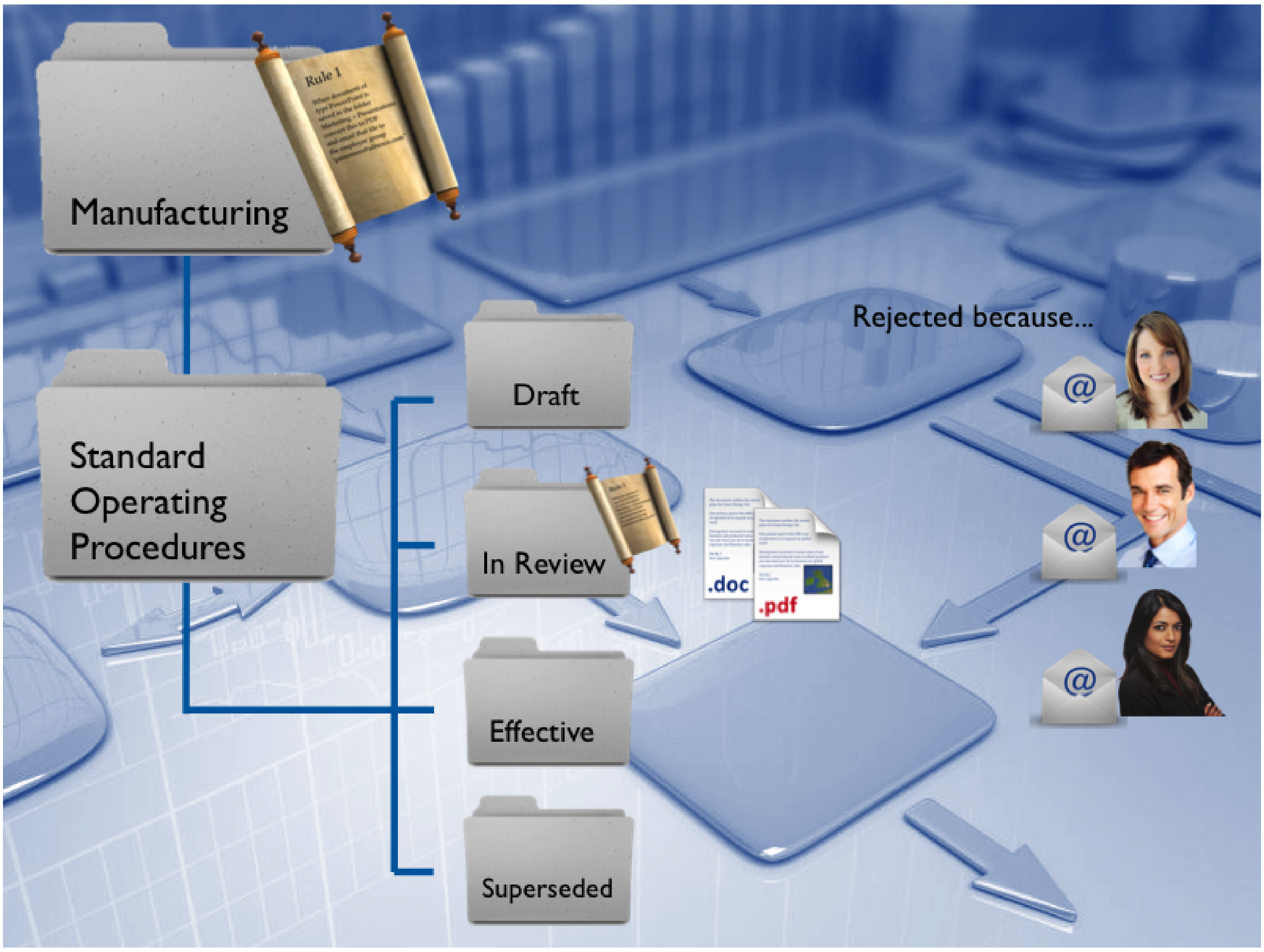
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The alfresco and share WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within ~tomcat/webapps/alfresco or ~tomcat/webapps/share. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the amps directory.

The second file I place in the amps_share directory.

I stop the Alfresco server.

Next I use the apply_amps command, which is found in the bin directory. This command applies the AMP files that are located in the amps and amps_share directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The apply_amps command has copied the content into the alfresco.war file. When the server starts the alfresco.war file is expanded and deployed into the ~tomcat/webapps directory, maintaining the same folder structure.

The same is true for the Share components from the alfresco-rm-share AMP file, now found in the share.war file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

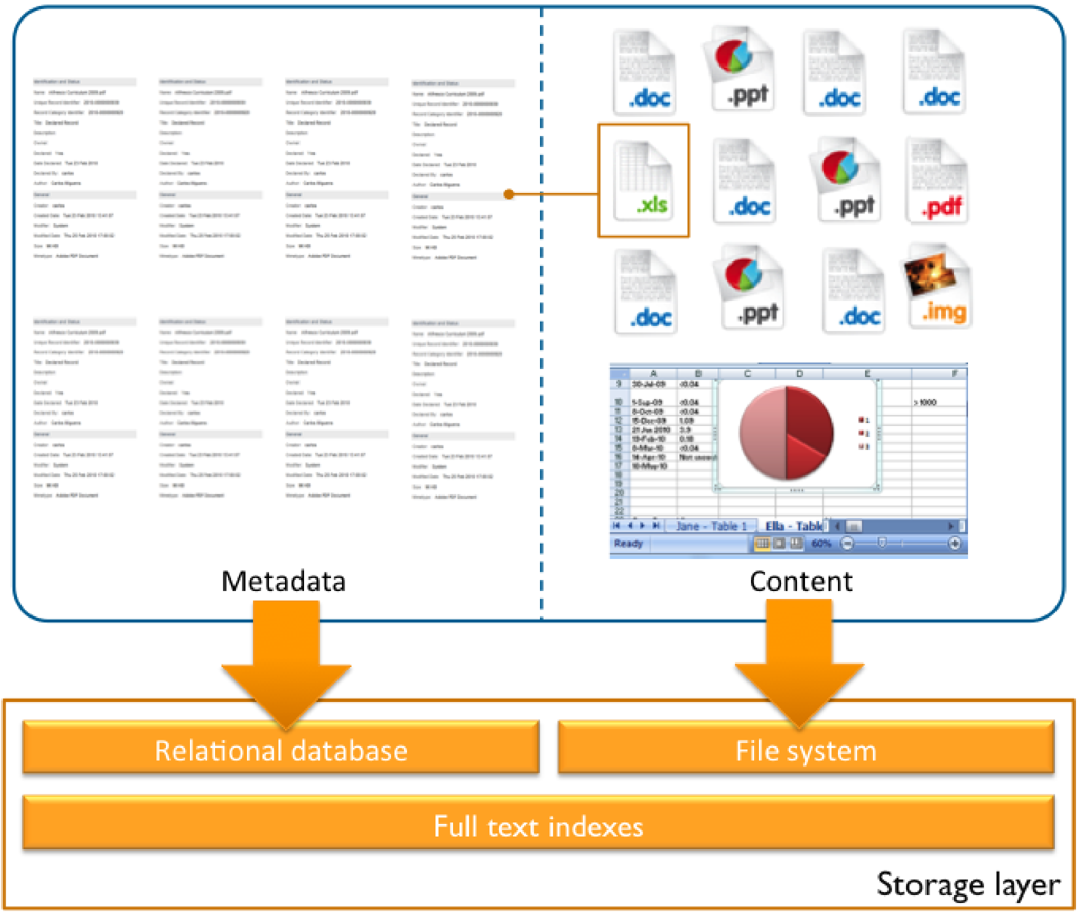
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

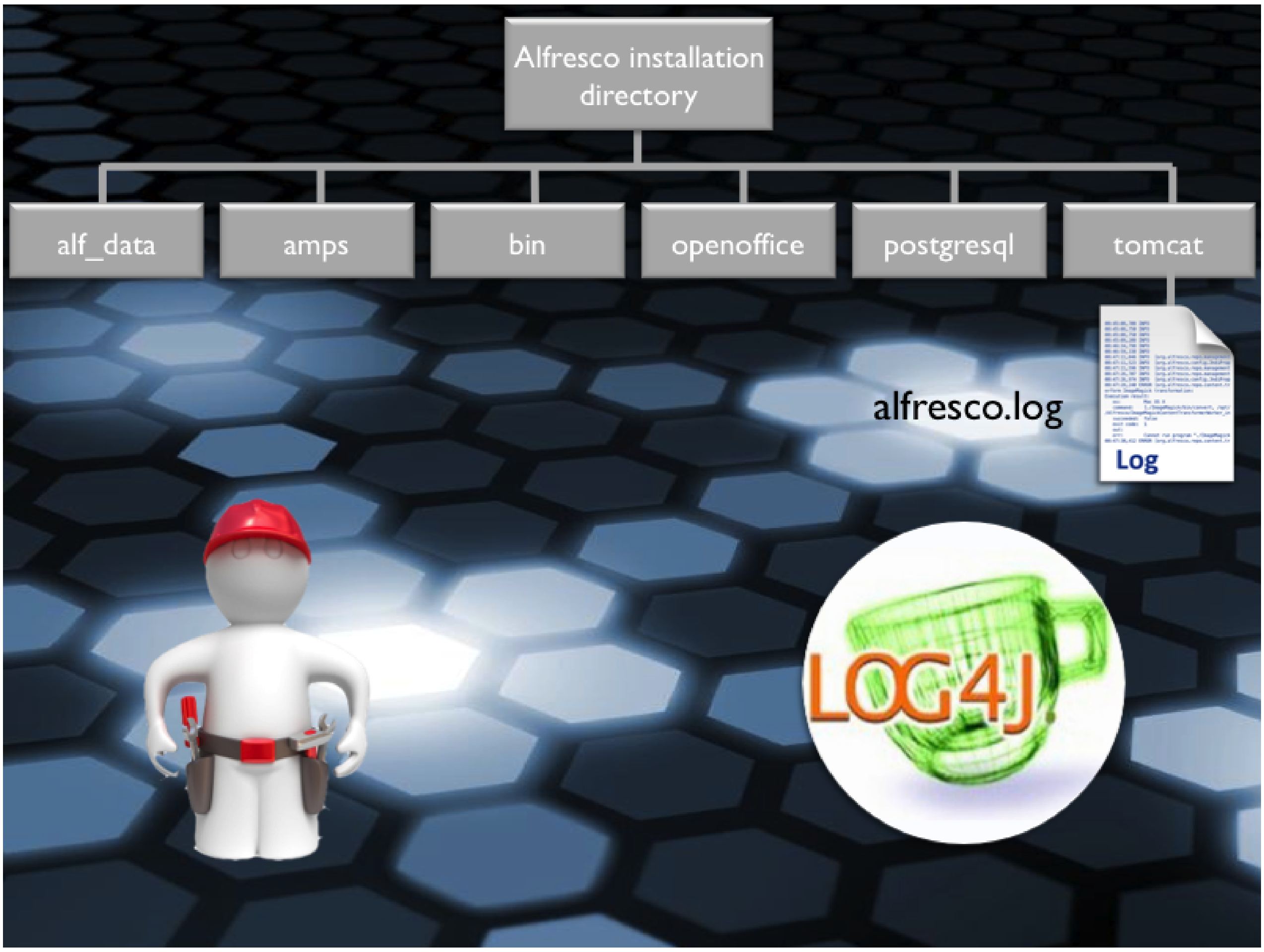
       Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

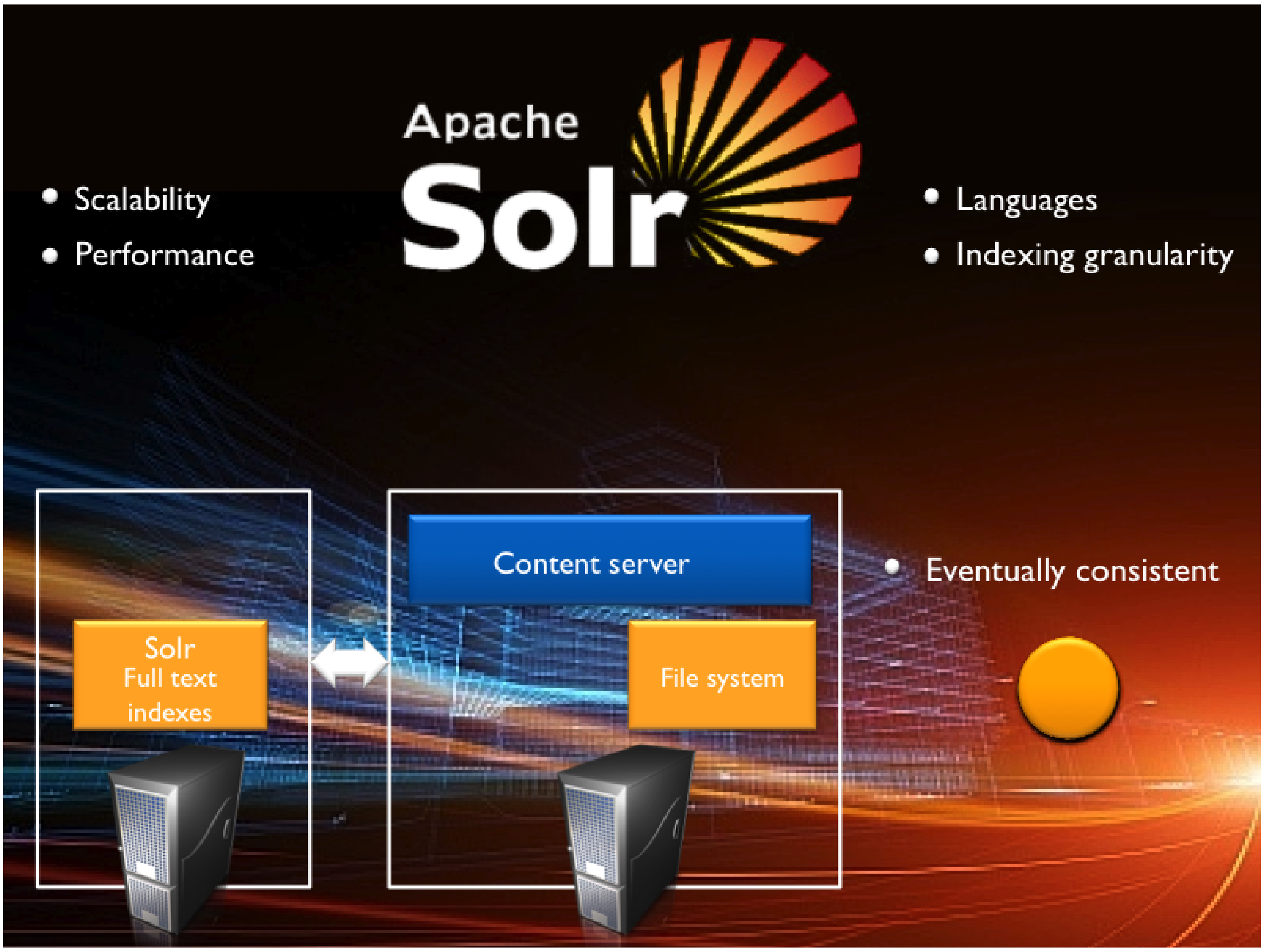
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

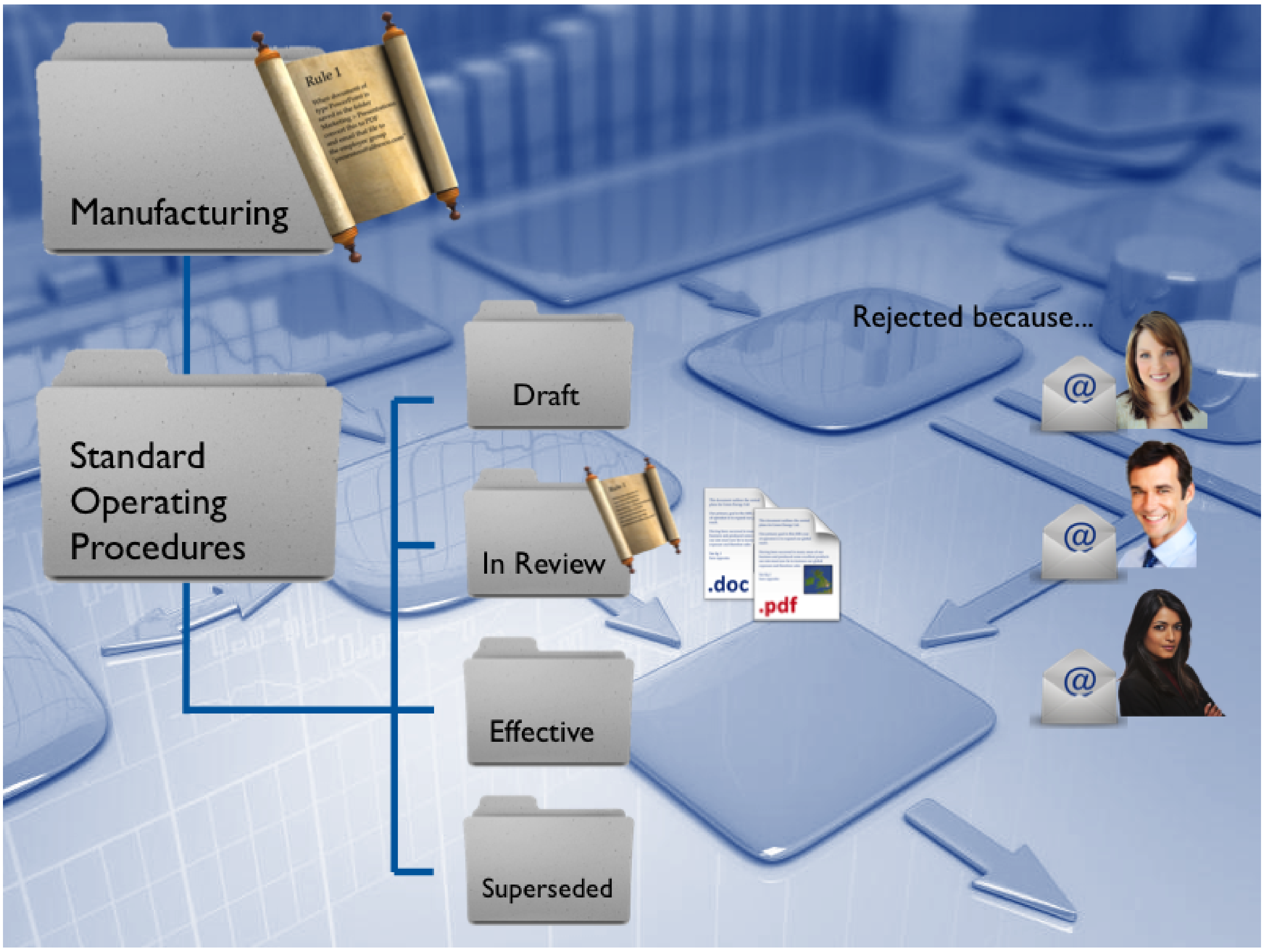
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.
    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco™ Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

.amp
Alfresco module package
XML
.img
.css

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn't have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

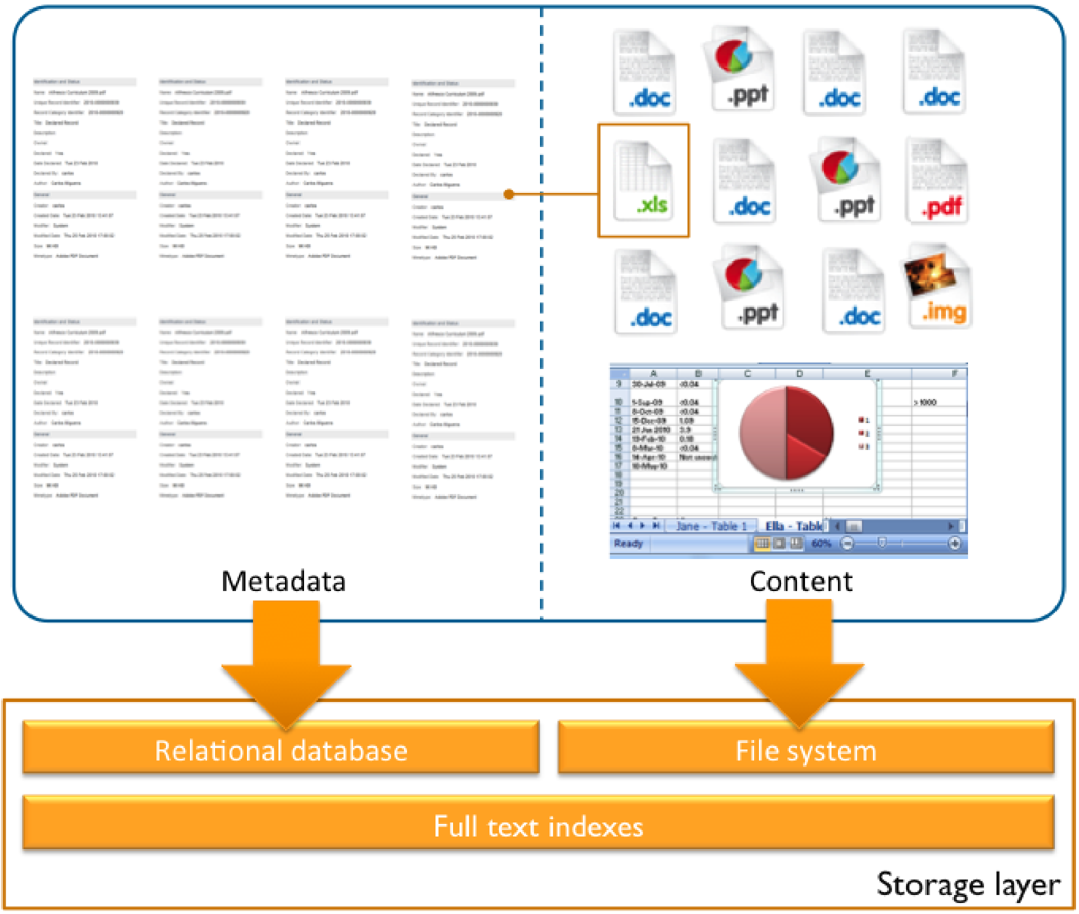
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don't have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:
    1. Start JConsole and connect via the remote process.

        (The connection string and account details are attached to this presentation.)
    2. Disable inbound email.

        You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

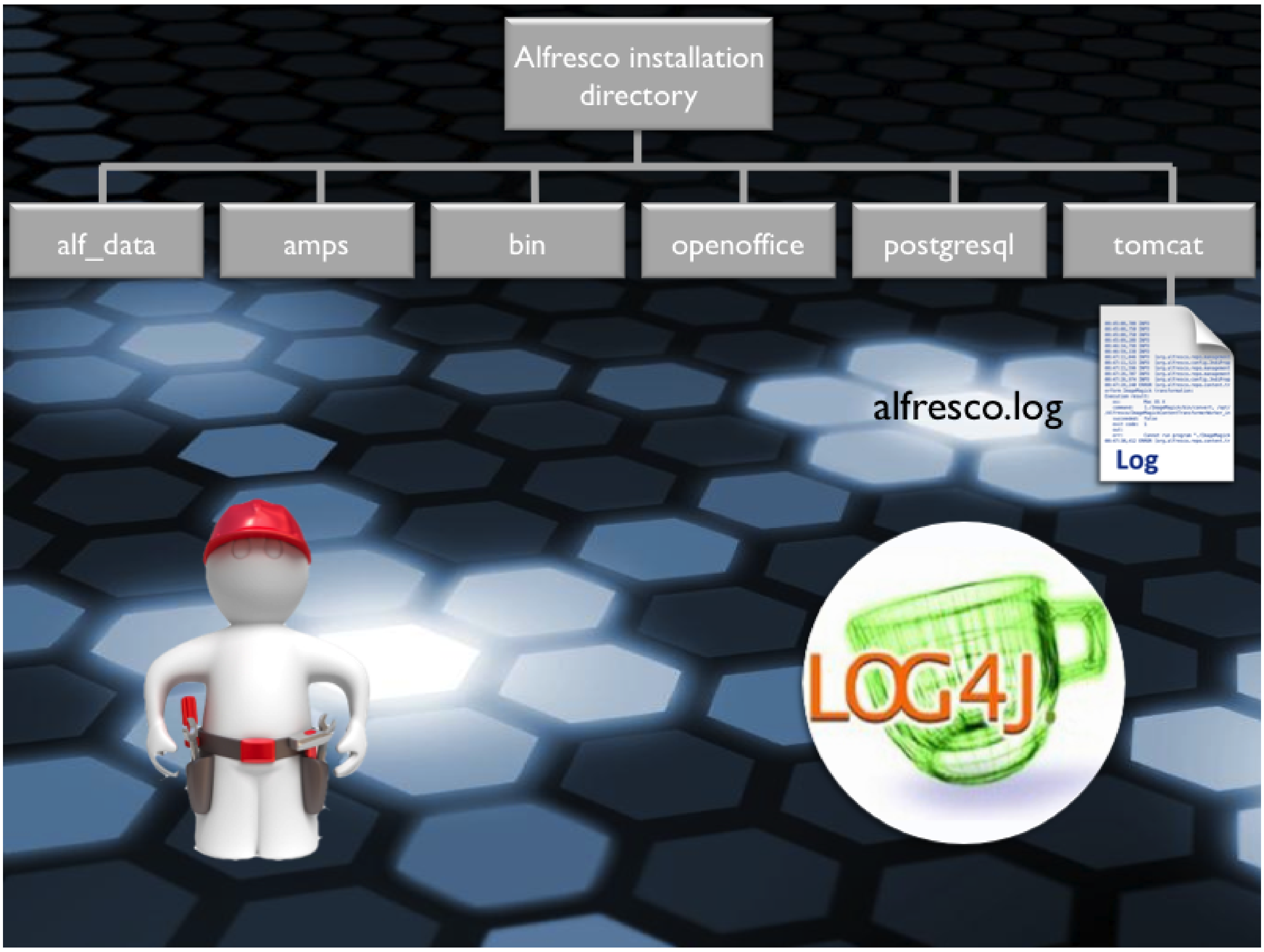
        Remember to restart this subsystem using the Operations.
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

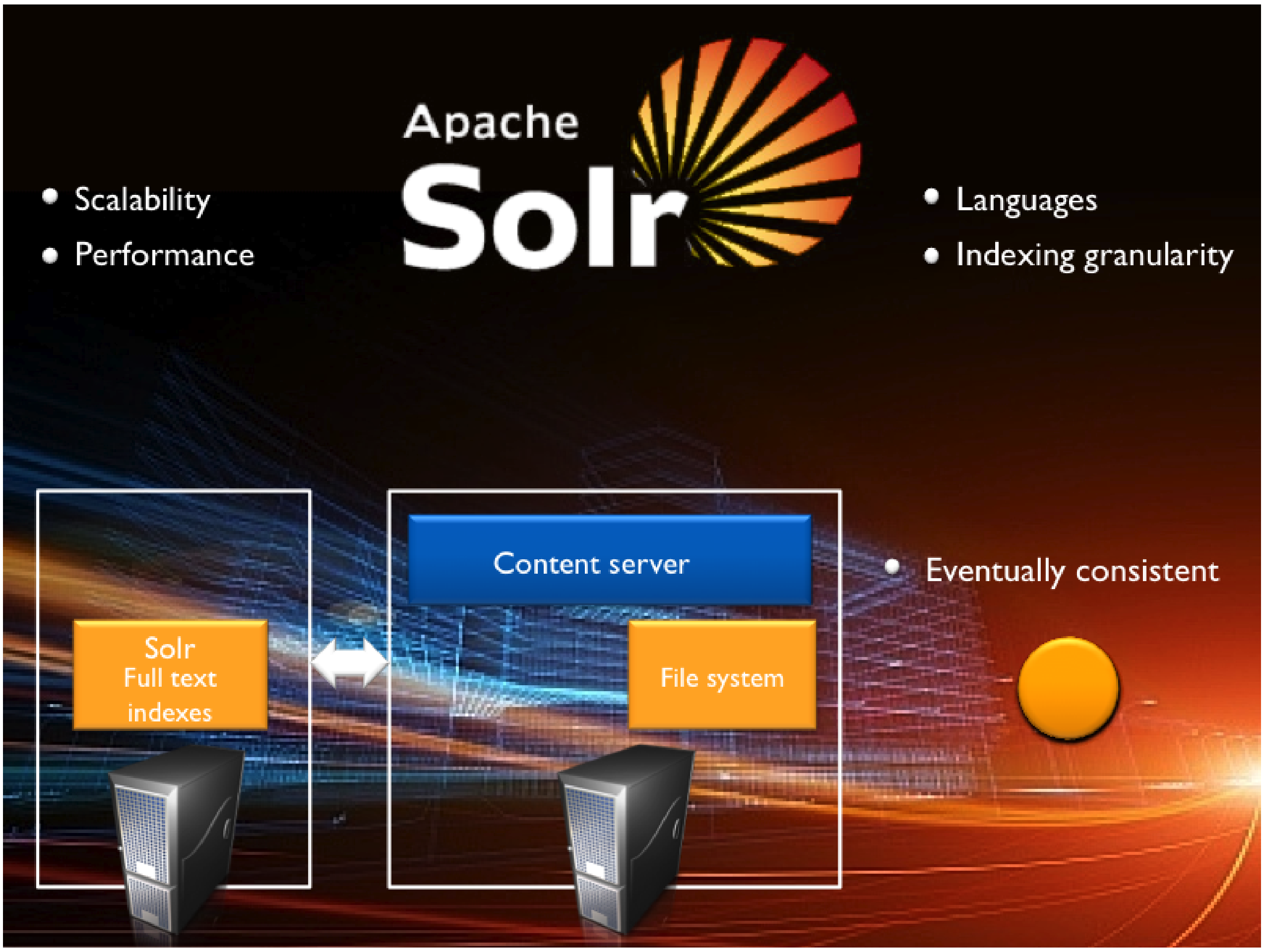
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

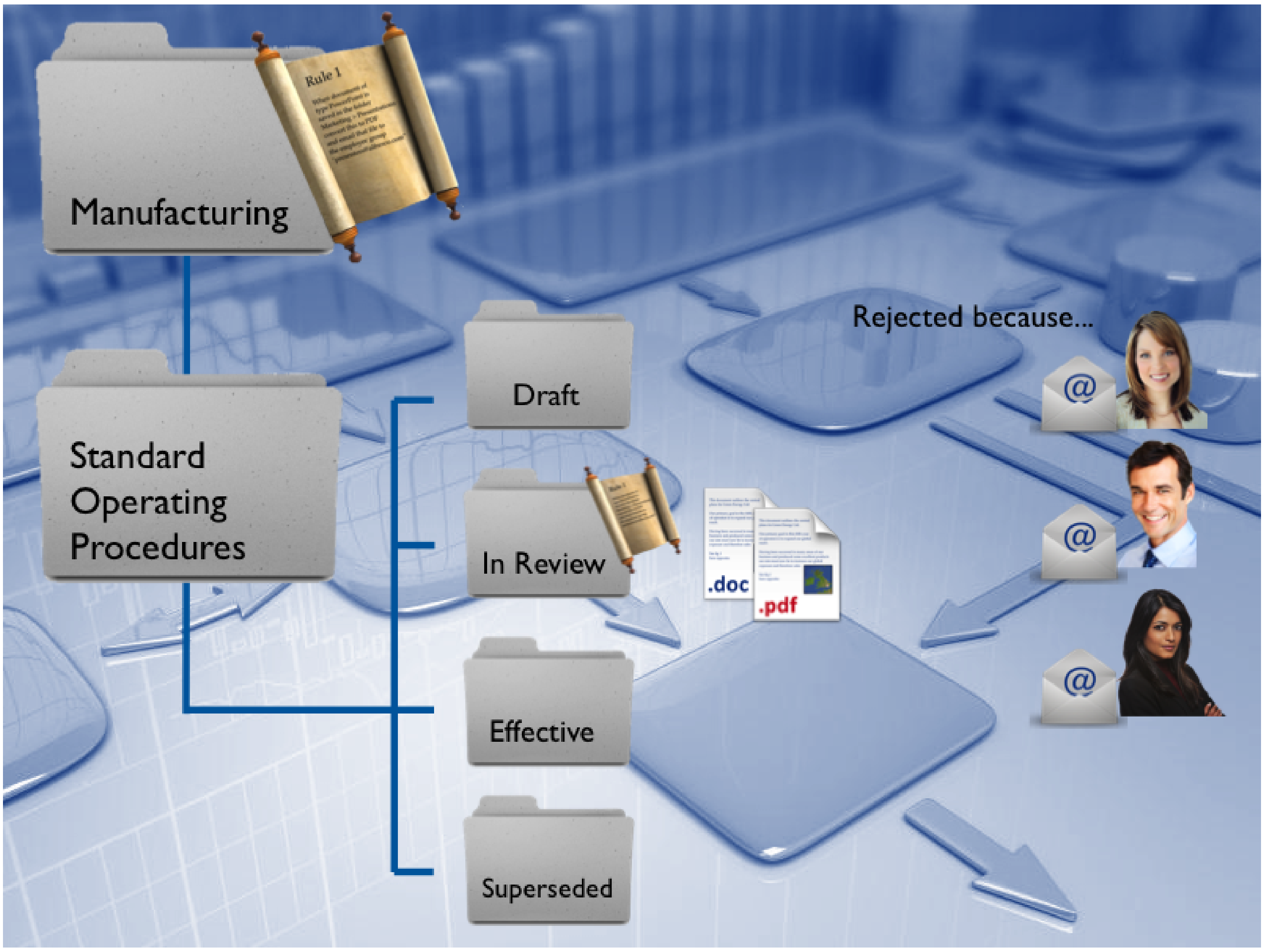
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.

Alfresco Training

Alfresco Elements

Managing the Repository

Contents

Document information

The United States export control laws and regulations, including the Export Administration Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

You may not use this documentation if you are a competitor of Alfresco, except with Alfresco's prior written consent. In addition, you may not use the documentation for purposes of evaluating its functionality or for any competitive purposes.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at training@alfresco.com.

Managing the repository

Introduction

Managing the repository consists of undertaking tasks to ensure you have a reliable and well performing repository, as well as managing ongoing tasks which may change the repository or apply adjustments.

The regular ongoing tasks include monitoring health and usage of the repository and preventative maintenance. At other times you may need to install applications and change the times and frequency of scheduled jobs.

Alfresco provides a very powerful way to add rules to a folder to provide creative solutions for automating and managing your content, we will look at what you might use such rules for and how you can set these rules up as administrator.

AMP files

For production applications the recommended way of deploying applications is to use the Alfresco module package (AMP). An AMP file is a collection of code, XML, images and CSS that collectively extend the functionality or data provided by the repository.

It may be fine to deploy a single small extension manually, however once you have more than one extension or a complex extension then installation will be challenging.

An AMP file is essentially a ZIP file with a pre-determined internal folder structure and, at its minimum, a single required configuration file. An AMP file is applied to a WAR file, which is then redeployed to the application server.

The `alfresco` and `share` WAR files contain the core Alfresco product and the Alfresco Share user interface. These are held in WAR files which are expanded and deployed during the server startup.

AMP files which are applied to these WAR files are naturally distributed at this time.

The extension will therefore appear as part of the Alfresco web application; for example, all the files are within `~tomcat/webapps/alfresco` or `~tomcat/webapps/share`. The repository also maintains a registry of the installed modules and their version enabling the modules to be upgraded independently of the WAR file.

There are some disadvantages though, since this means the files are all loaded by the application server's main classloader, you lose the ability to reload configuration files dynamically. In addition, since the AMP construction, WAR integration, and deployment are not conducive to a streamlined development cycle, you should only use AMP files for completed extensions.

Demo: AMP deployment

The Alfresco Records Management software is distributed as AMP files, let’s explore the installation to demonstrate the process of installing an AMP.

The software comes in two components;

```
alfresco-rm-2.0.1-147.amp
```

```
alfresco-rm-share-2.0.1-147.amp
```

The first file contains the additional functionality that is applied to the alfresco installation. The second file contains the software to enhance the Share user interface.

If you view the contents of these files with a zip utility you can see the folder structure.

I move the first file to the `amps` directory.

The second file I place in the `amps_share` directory.

I stop the Alfresco server.

Next I use the `apply_amps` command, which is found in the `bin` directory. This command applies the AMP files that are located in the `amps` and `amps_share` directories.

I start the Alfresco server, then login as the administrator.

I can now add the Records Managements functionality to the dashboard.

Returning to the original AMP file I again explore the folder structure. The `apply_amps` command has copied the content into the `alfresco.war` file. When the server starts the `alfresco.war` file is expanded and deployed into the `~tomcat/webapps` directory, maintaining the same folder structure.

The same is true for the Share components from the `alfresco-rm-share` AMP file, now found in the `share.war` file and similarly deployed.

This deployment method, commands and paths are the same on alternate platforms.

Scheduled jobs

Alfresco runs a number of inbuilt jobs which perform background processing, the open source Quartz Scheduler is used to perform the scheduling function. The scheduler runs a number of jobs from various content cleanup jobs, through full-text indexing and feed automation jobs. Some of the additional modules such as Web Content Services and Records Management also implement their own periodic jobs to perform specialized functions.

The frequency of jobs is optimized for performance in most situations but can be altered. This requires some thought and understanding to ensure that the system behaves as expected and is therefore considered an advanced system administration topic.

A healthy repository

Your regular preventative maintenance routine should start with repository monitoring. A healthy repository is characterized by a number of different factors; no errors in the log files, sufficient free disk space, a range of high quality backups, little or no fragmentation on disk volumes and a fast performing database. If all of these factors are correct then you should enjoy a stress-free experience which provides good performance for users. Performance is obviously subjective, but in a healthy repository response times should be good and you shouldn’t have complaints from users.

This list provides you with guidance of the areas to look at on a regular basis.

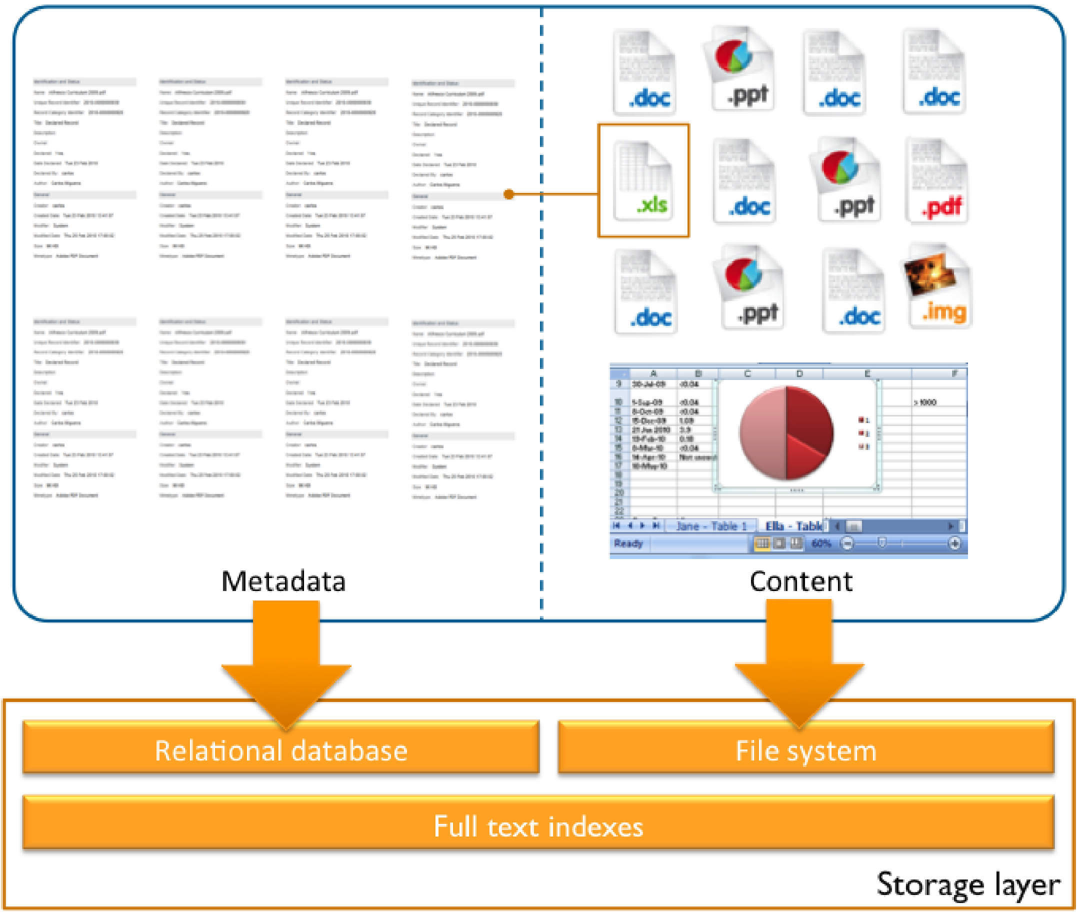
- Check log files
- Check disk space
- Check backups
- The state of the full text indexes
- Remove orphaned content
- Database optimization

How regularly you look at these depends on a number of factors such as the usage patterns of your repository, the service level objective you have for your system and the transaction throughput. If your production system is heavily utilized and you work in a well structured IT department you may use various tools to make this job easier, for instance you may integrate the Alfresco log file into a complete log monitoring service for the organization so that you are notified of any log files errors which occur through a notification service, like email, that means you don’t have to physically inspect the log file.

File storage method

Within the Alfresco storage layer content is stored on the file system and content metadata is stored within the database. This is an efficient architecture which delivers optimal performance.

When a document is deleted from the system only its metadata is removed, the content file is left as orphaned content. To deal with this Alfresco implements a scheduled job to clean-up these files, part of you routine maintenance should ensure that this job is running at the right intervals and executing successfully.



Since Alfresco relies so heavily on the underlying database it is of utmost importance that this is performing at peak efficiency. All of the database systems which Alfresco works with have ways of tuning and optimizing performance for different scenarios. It is important that you involve a database administrator early to setup a regime of database maintenance and optimization which suits your usage profile.

Monitoring usage

Monitoring usage is best achieved through a Java Management Extension (JMX) console, (this must support JMX remoting). There are a number of JMX consoles available, for example Jmanage, MC4J and JConsole. We use JConsole in our examples as this ships automatically with Java standard edition 5 onwards. The JMX interface is only available with the Alfresco Enterprise edition.

Demo: JMX console

Let’s explore the use of the JConsole Java Management Extension with Alfresco.

JConsole is invoked through the terminal, on the Windows platform it is usually found in the `bin` directory of the Java install.

As indicated the Java Management Extension must support remoting. I connect to the Alfresco repository using the remote process connection, using the following string.

service:jmx:rmi://ignored/jndi/rmi://127.0.0.1:50500/alfresco/jmxrmi

This string is attached to this presentation, so you can download and use it within an upcoming lab.

The default username is controlRole and password change_asap (for productions system you obviously need to change this !)

There is a large array of information that can be seen using a JConsole, for example the number of users and groups in the system, the number of connections defined and used, the total size of content currently stored, the size of the indexes, the transformers available and the patches applied to the system.

This information exceeds well over 100 discrete items of information. Should you ever be in contact with Alfresco Support they will often ask for a JMX dump as this provides a snapshot of the current system parameters and its status (this is achieved through the administrator's panel of Alfresco Share).

Not only is it possible to view system information, it is possible to alter and adjust the live system by editing parameters through JConsole.

For example you can enable or disable file servers such as CIFS.

Whenever you make a change to a subsystem value the subsystem will be stopped and you must restart it if you want the service to be enacted. Such a change is synchronously actioned and does not require you to restart the server.

Subsystems (such as authentication, replication, email, file servers) are a good use case for a JMX console, where you can build complex configurations without endless restarts.

The revert method will revert the property value to that held in the `alfresco-global.properties` file or in its absence the default value.

Most property edits are persisted in the database and will be remembered on server restarts. Some property changes, such as the logging level, take their value from their respective configuration file upon startup and are not persisted, these are principally the systems outside of the core alfresco system.

There may be cases in a production situation where you would wish to disable JMX entirely, you can do this by editing the core-service-context.xml file and commenting out the `RMIRegistryFactoryBean` section. This change will take effect at the next server restart.

JMX edits vs global properties

As discussed in the Repository configuration Alfresco Element the `alfresco-global.properties` file is loaded last. Any property value held in that file overrides the value from a previously loaded configuration file.

In the case where you edit system values with a JMX console these will persist through server restarts and settings in the `alfresco-global.properties` file will be overridden.

It is therefore good practise to copy any parameters edited through a JMX console to the `alfresco-global.properties` file.

JMX and clustering

Using a JMX console when you have a clustered environment means that property changes are made once and across the whole environment. If you were to access a specific machine and edit a property file directly then this could lead to consistency issues depending on your configuration.

JMX

Whilst the JMX console is exceptionally useful it only provides a real-time view of the system. If you are interested in historical trends and patterns, such as document growth, you will need to do some additional work such as configuring the audit trail to capture the information you are interested in.

You can learn more about the JMX interface from the Alfresco documentation online.

`http://docs.alfresco.com`

Lab - JMX console

1. In this lab you will use JConsole to access the Alfresco server and make property value changes. Your tasks are:

    1. Start JConsole and connect via the remote process.

       (The connection string and account details are attached to this presentation.)

    2. Disable inbound email.

       You will find this property in `Alfresco > Configuration > email > inbound > Attributes > email.inbound.enabled`

       Remember to restart this subsystem using the Operations.

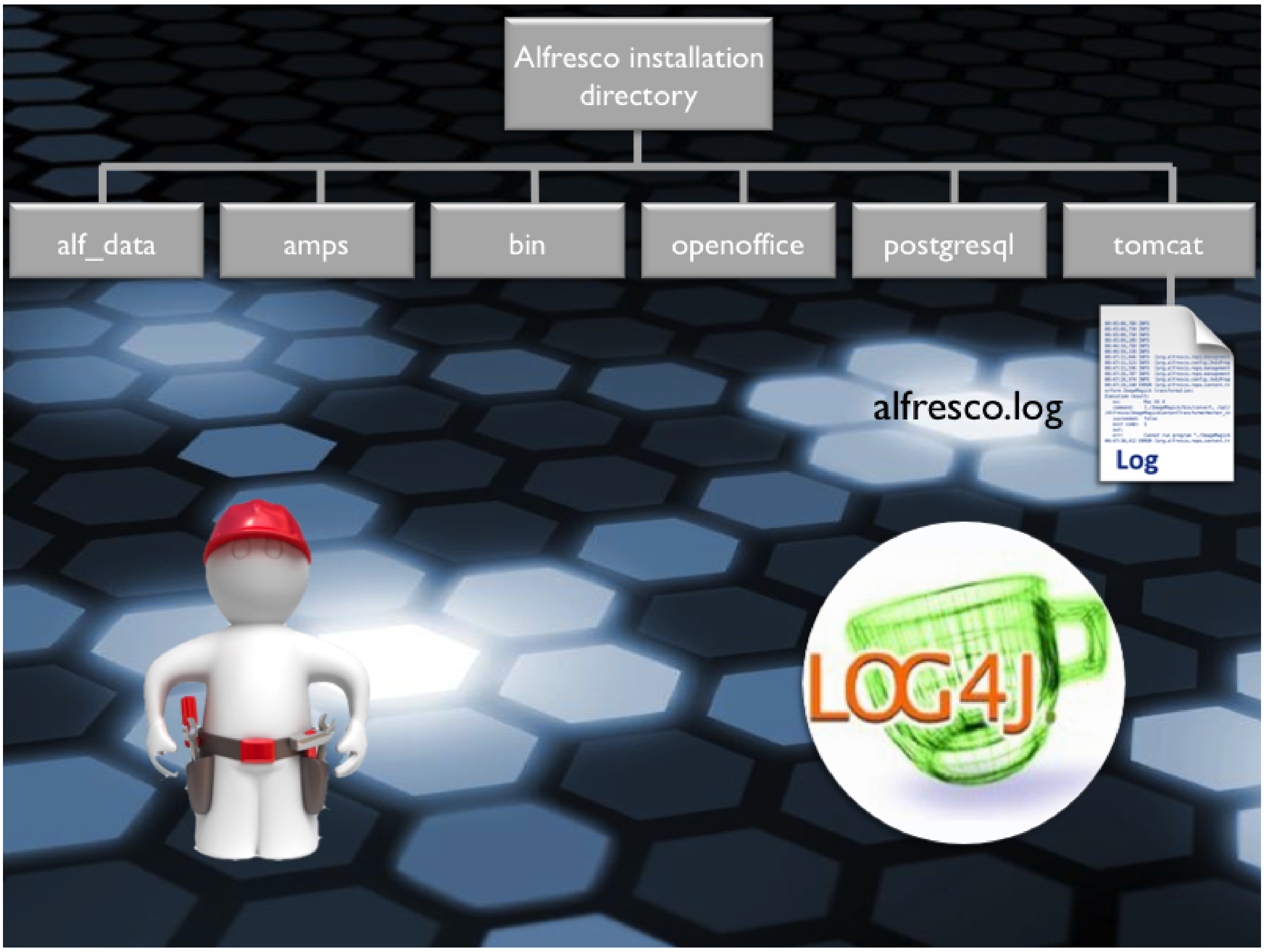
    3. Restart the Alfresco server.
    4. Check that the change has persisted.
    5. Revert this change.

Managing the repository

Logging

Alfresco records server activity and errors within the `alfresco.log` file.

The logging system used is Log4j which is highly configurable and provides varying levels of error reporting allowing you to also use the log file for debugging and troubleshooting.



Although the alfresco.log file is your first port of call, Alfresco does not operate in isolation and other parts of the system may have an impact on Alfresco, hence there are other log files which you should also monitor for errors. The primary ones are the log file for your database and the log file for your application server. You will also want to visit the operating system logs as these may provide evidence of low level problems which affect higher level systems such as Alfresco.

The logs for the operating system, application server and database will vary from system to system, so you need to check the documentation for your particular stack. On some systems there may be an interactive monitoring tool for example the one shown here for MySQL. For Alfresco however the items you find in the `alfresco.log` file will come in a consistent pattern.

A good tool for sending you an email to periodically highlight errors that have occurred is `tail_n_mail: http://bucardo.org/wiki/Tail_n_mail`

This is a Perl script and is therefore supported on multiple platforms.

Storage space

Most of us use our hard disks like closets, stuffing in files and then forgetting about them. But no matter how big a disk you have, it's going to run out of free space one day, and running out of

disk space when the CEO is trying to save his or her imminent presentation could hurt you badly. Keeping an eye on disk usage doesn't take much time or effort, here are some tips and tools. Alfresco does not provide a tool for monitoring disk space, there are many excellent tools already available to perform this task, you need to choose one that is appropriate to your operating system and infrastructure. The main rule when monitoring disk space is to ensure that the content location has at least 20% free. You may want to enable user quotas inside Alfresco, but this very much depends on your organization's policy and this can only ever be a blunt instrument in an enterprise content management system like Alfresco. Whilst you store any type of content inside Alfresco there are some content types which you should consider not allowing inside your repository, mainly these are executable files and databases, such as Access databases.

Demo: User quotas

Alfresco has the ability to track the cumulative size of content added by each user of the system. This is typically displayed in kilobytes, megabytes or gigabytes.

This feature is disabled by default in version 4 of Alfresco and can be enabled by setting the following value in the `alfresco-global.properties` file.

```
system.usages.enabled=true
```

This will come into effect on the next server restart.

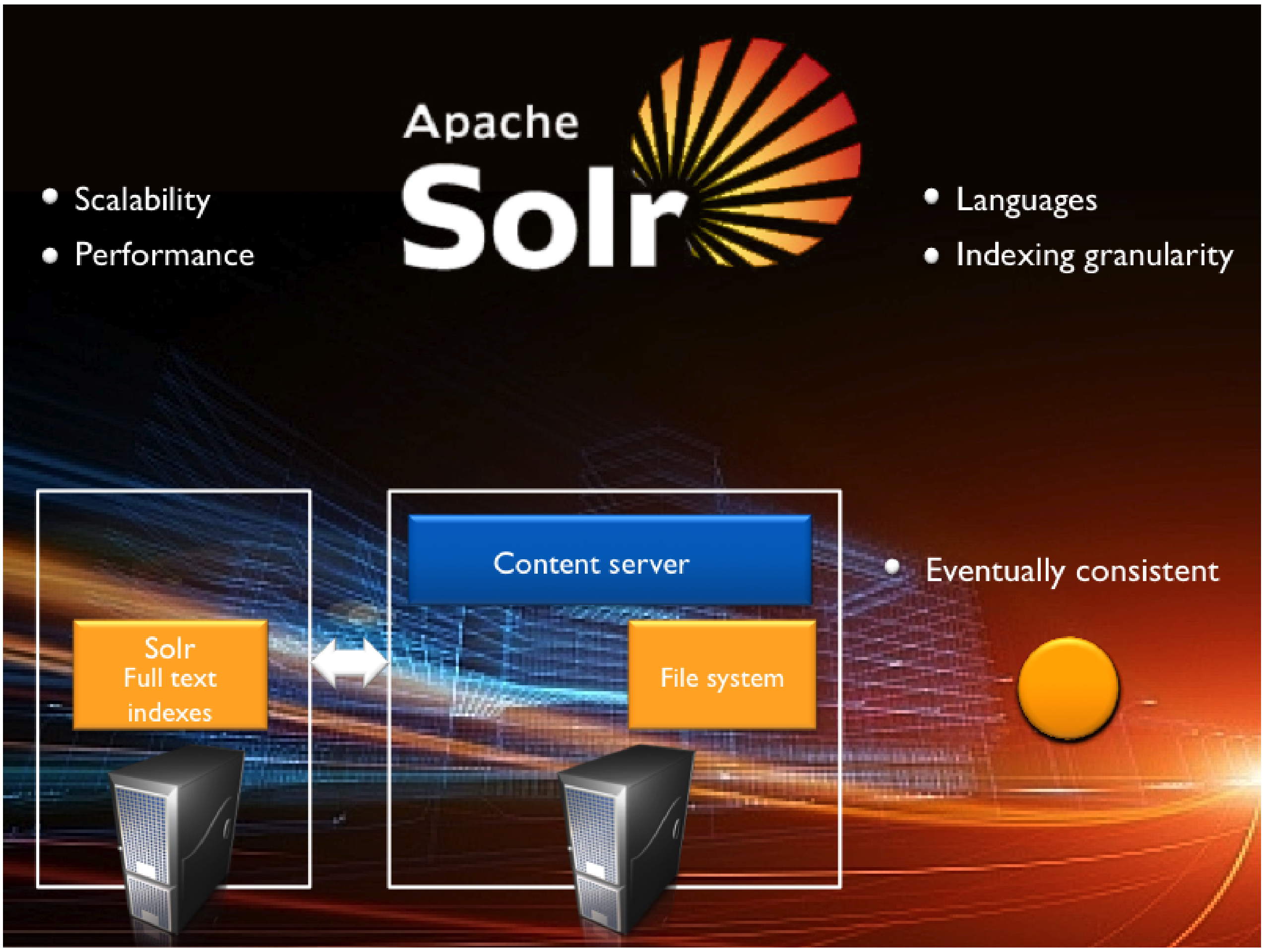
The current stored content size is visible to the administrator when a user is examined in the User administration tool.

The administrator can optionally set a quota for a given user. This can be set when the user is created or imposed at a later time.

If a user tries to add or edit content, via any repository interface, so that usage will exceed their current quota then the user will see the error message "Quota Exceeded".

Solr

Alfresco version 4 introduced a change in architecture for indexing and user search; Apache Solr is now the default full text indexing technology. Lucene was used prior to this and can be enabled in favor of Solr if required.



Solr embeds and leverages Lucene, however Solr provides a functionally abstracted layer and can be installed on a dedicated server.

Solr delivers a number of advantages over Lucene including scalability, improved performance, enhanced language support and fine control over what gets indexed.

A principle issue for Lucene is transactional dependence; the indexes are updated as content is loaded or edited. This delivers fully accurate and synchronised indexes, but means the content server is locked during the transaction. This is a particular problem if the indexes ever have to be rebuilt from scratch, an operation that can take many hours with large content repositories.

The drawback and consideration when using Solr, is that it works asynchronously and is known as "eventually consistent". Content is indexed at small regular intervals and until the indexes are up to date.

Database factors

It is anticipated you will have an experienced, certified database administrator on staff to support your Alfresco installation, however you should understand why certain performance problems may arise with databases and here we take a high level overview of the factors which affect database performance.

One of the primary factors affecting database performance relates to having the correct indexes defined, however you should never need to do this as Alfresco has already created the optimal indexes for best performance. If your DBA finds that an index appears to be missing or adding an index improves performance this should be reported to Alfresco support.

All databases can be tuned for performance in different ways for differing usage patterns, for example high throughput, long running queries, decision support or mixed usage. This is typically

done through a number of parameters which affect cache size, threads and connections, pooling of resources and so on. There are courses, books and documentation available which deal with database tuning and your resident DBA should be familiar with these.

All the databases which Alfresco run have query optimizers which rely on database statistics to be up-to-date for best performance, all the databases offer tools for gathering and updating these statistics and this is often an overlooked task. Note that in some database types index maintenance and optimization is an expensive, and in some cases blocking, operation that can severely impact Alfresco performance while in progress. Your certified DBA will have best practices for scheduling these operations in your database.

All databases use files on the file system to manage their tables and indexes. Over time the files and tables can become fragmented and this too can have a degrading effect on performance. You should ensure that your DBA has set in place procedures for dealing with this fragmentation in the Alfresco schema.

Of course all of these depend on the database which you are running and you need to look to the database specific documentation for more information than can be provided here.

State of the database

If you have access to the database there are a number of visual tools which can be run that will provide you at least an overview of the health of your database. Some databases come with their own very useful graphical tools, for example MySQL, equally there are generic tools out there which can monitor a variety of databases and provide useful information, for example Hyperic and Splunk.

Backup

Ultimately as with any system you have to plan for unexpected disasters and problems. Any system is prone to failures of varying kinds ranging from hardware failure through to simple human error. You must ensure that you have a good set of backups and that backups have been tested to work and your recovery procedures support the business.

This process is explored in the Alfresco Element, Backup & recovery.

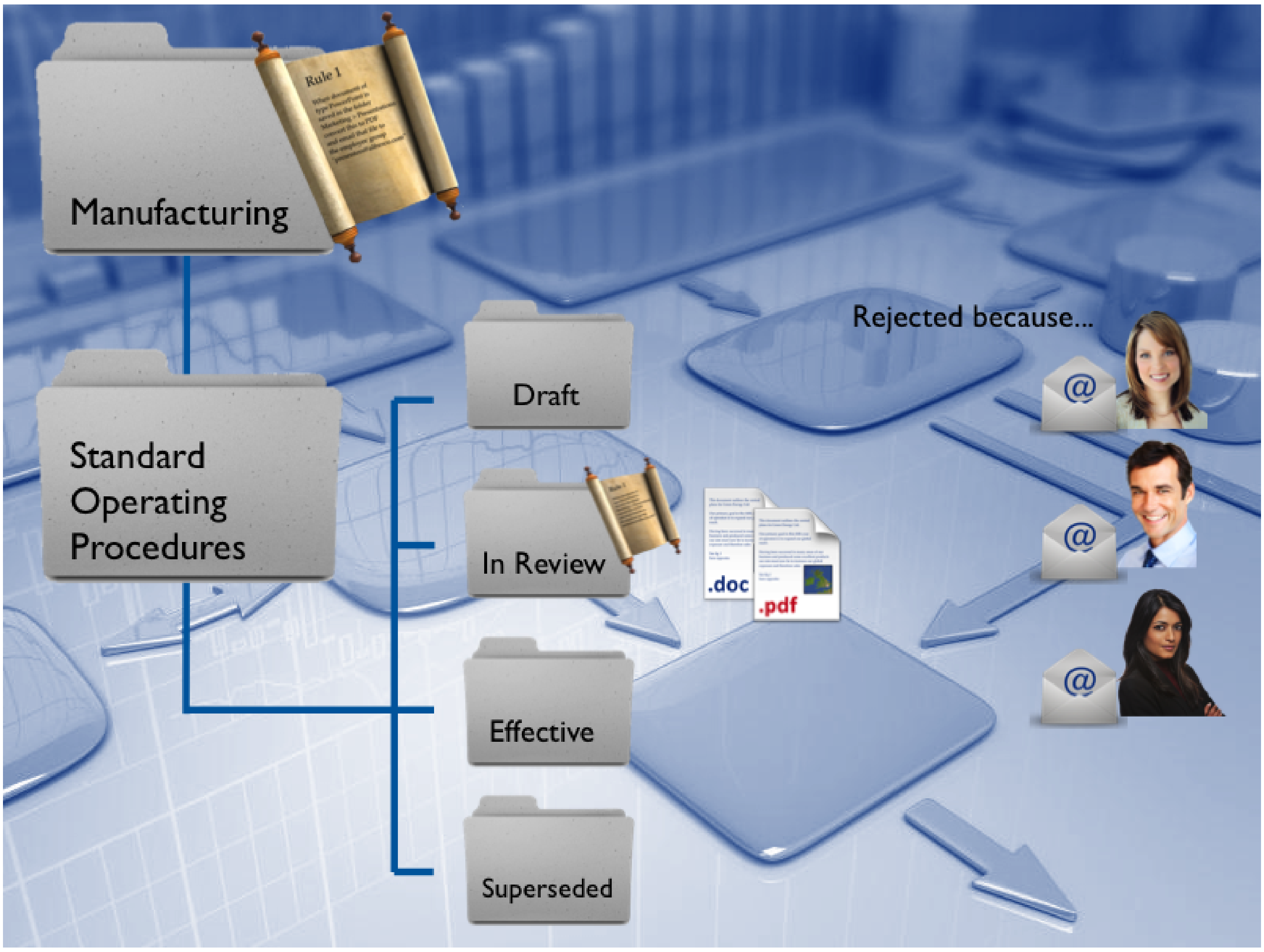
Holistic monitoring

We have talked about various ways of monitoring the health of your installation, you can of course use a number of widely available tools, which your organization may already use, for a holistic monitoring of all the components which make up an Alfresco installation, from the database to the application server.

Content rules

Content rules are a very powerful capability within Alfresco and let you accomplish a wide variety of content management tasks in your repository through a simple user interface. Typically these tasks would have previously required a developer to customize the system. Rules are limited by imagination rather than functionality, however some of the more typical uses of rules are for example: to perform some action on the document content; to perform some action on the document's properties; to change a document's status, notify somebody of a change in a document or put it into a workflow. Using rules you may also move, or copy, a document between folders.

Rules are defined on a folder and have a name and can be inherited, by default they just apply to the folder on which they are defined. Rules are designed to be run automatically when the rule condition is met and are by default run in the foreground synchronously.



Although each rule is simple and easy to define, by their nature they typically perform a single action. For more complex requirements you may need to combine two or more rules together to achieve your goals. When you have multiple rules they are triggered in a particular sequence which can be defined by you as an administrator.

Whilst a lot can be achieved with the rule actions which are standard, it is possible that the operation you want to achieve on your content is not there out of the box. In this case rules provide a catch-all action, execute script, action which allows a developer to write their own behavior, thus extending the range and power of content rules.

Demo: Content rules

In the following demonstration and lab we are going to use a folder structure which has already be created in the Green Energy repository. This defines the standard operating procedures for manufacturing.

In this example a rule has been added to the standard operating procedures folder which adds an aspect to any document added to this folder. The rule is such that it is inherited down to any child folders.

Another of the folders, In Review has its own rule defined Transform to PDF.

This is typically how you establish rules across folders.

In this demonstration I will establish a review and approval process using rules.

As the administrator I navigate within the repository to:

```
Geo-Thermal Division > Manufacturing > Standard Operating Procedures
```

And create a rule 'Transform to PDF for review'.

This operates on Word documents only and transforms content newly created or moved into the folder to a PDF file. The rule is created as a background process rather than the default foreground and is not applied to subfolders.

I will now add a second rule to this same folder where an approval action moves the PDF file to the Effective folder and a rejected action moves the file to Draft. The process is a foreground one and not applied to subfolders.

The ability to create and manage rules is defined by the authority held by the user. Within a Share site a user must hold the role of Manager in order to create and manage rules. Outside of a Share site and for folders within the repository the Coordinator permission must be held. The creator and owner of a folder will automatically have this ability. Security and permissions are examined in the Permissions Alfresco Element.

This is clearly a very simple rule and in the practice you would probably add further rules to add functionality such as notify an individual or group when a document had been placed in the Review folder. Add an aspect such as an effective date when the document was approved. You might wish to add an aspect to capture comments if a document were rejected. All of this and more is possible through the rules capability of Alfresco.

Now the rule is established let me demonstrate its functionality.

I move the Standard Operating Procedures document (which is in Microsoft Word format) from the Draft folder to the In Review folder. Immediately the rule creates a PDF version of this file.

Next I approve the PDF document and witness this being moved to the Effective folder.

The alternate action would be to reject the document and in this case it is moved to the Draft folder.

In practise you would have users undertake these move, approve or reject actions, however security and permissions have yet to be established for these folders so I undertook this demonstration as the administrator. security and permissions are examined Permissions Alfresco Element.

Lab - Content rules

1. In this lab you are going to create a rule to transform any PowerPoint file created or placed in a folder structure to PDF, and move the newly generated PDF file to a specific folder. You can log in as the administrator for this lab.

    1. Navigate to the folder `Geo-Thermal Division` in the repository.
    2. Add a rule to the Marketing folder to transform PowerPoint files to PDF and move the created PDF file to the folder:

        `Geo-Thermal Division > Marketing > Presentations`

        - The Mime type you should choose is "`Microsoft PowerPoint 2007 Presentation`".
        - Make the rule inheritable.
        - Set the rule to run in the background.
    3. Test the rule by uploading the PowerPoint file: `Building a Brand Identity.pptx` to the folder:

        `Geo-Thermal Division > Marketing > Branding Project`

        The Building a Brand Identify file is found in the folder:

        `Desktop > Assets > Managing the repository`
    4. Check the Presentations folder for the newly created PDF file.

Managing the repository

Best practise

Rules by default are set to run in the foreground. This makes the user wait while the rules complete. Running a rule in the background however can raise additional difficulties and working in handling error conditions. Run all rules in the background unless you know that a rule is likely to generate errors often.

Alfresco lets you re-use rules from another folder. Consider using rules already defined which do what you need rather than redefining the same rule many times. Rules are simple so break-up complex business processes into multiple rules.

Another technique you can use is the concept of a drop zone or drop folder. This involves creating a folder which is used simply for uploading documents and has all the rules on it. The rules filter the incoming content and move the documents to the final destination folder.

Finally if you need to have a marker or status on a document the easiest way to do that is to create an aspect and then attached the aspect through a rule.

Summary

This section has given you a good grounding into the tasks which need to be done on a regular basis to keep an Alfresco repository healthy. We have also looked at the best way of installing applications and introduced the concept of scheduled jobs.