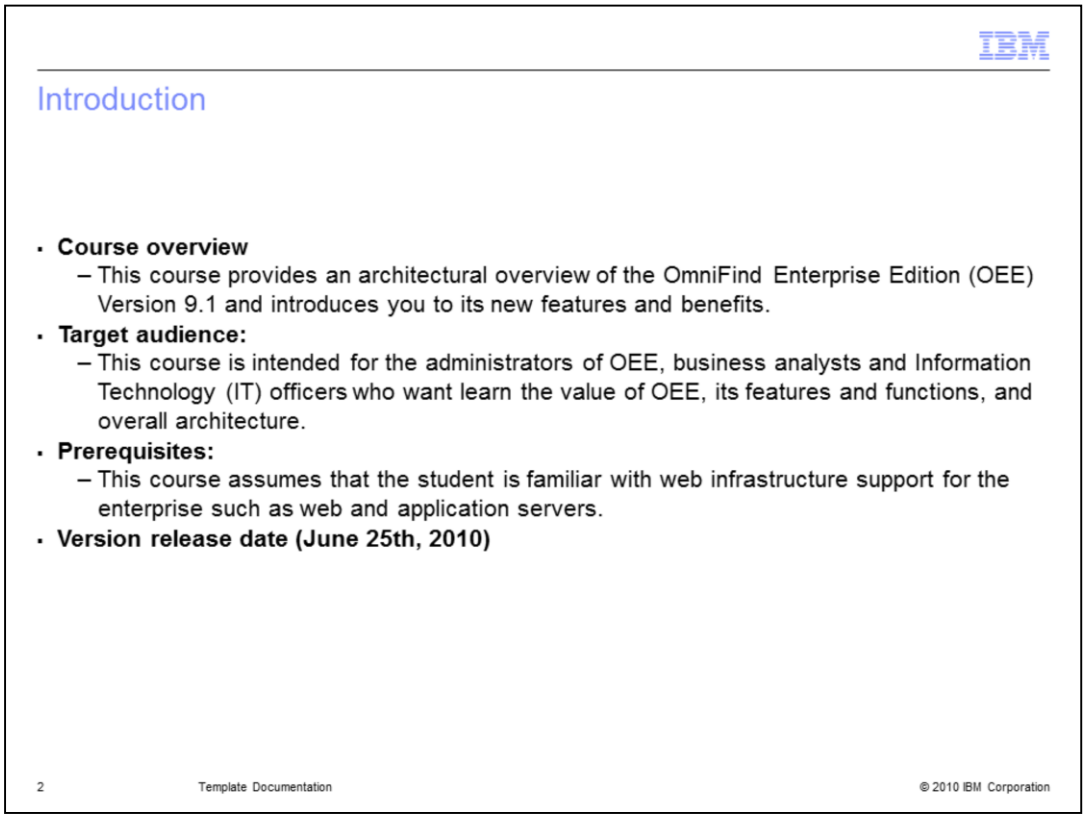
This course provides an architectural overview of the OmniFind® Enterprise Edition (OEE) Version 9.1 and introduces you to its new features and benefits.



This course is intended for administrators of OEE, business analysts, or Information Technology (IT) officers who want to learn the value of OEE, its features and functions, and overall architecture.

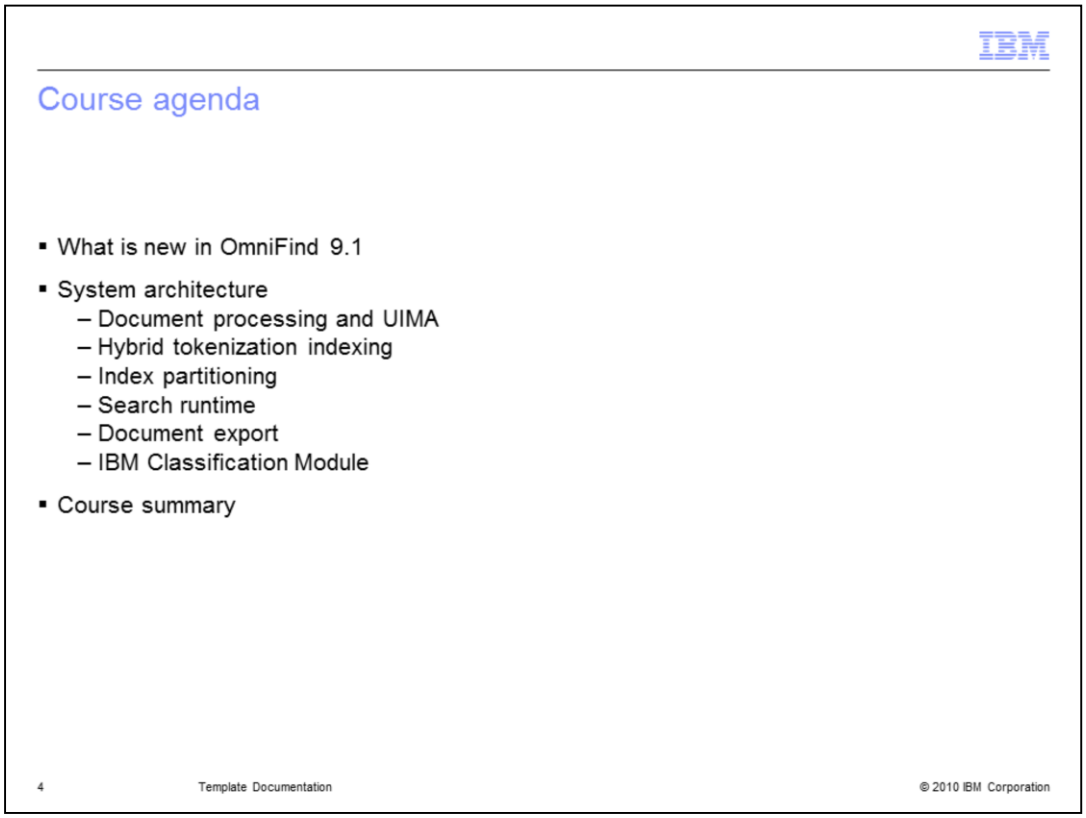
This course assumes that the student is familiar with enterprise web infrastructure support such as web and application servers.

After this course you will be able to:

Describe the major new features and functions of OmniFind V9.1.

Explain the new architecture of OmniFind and its supported configurations.

Describe the benefits of IBM Classification Module's tighter integration with OmniFind V9.1.

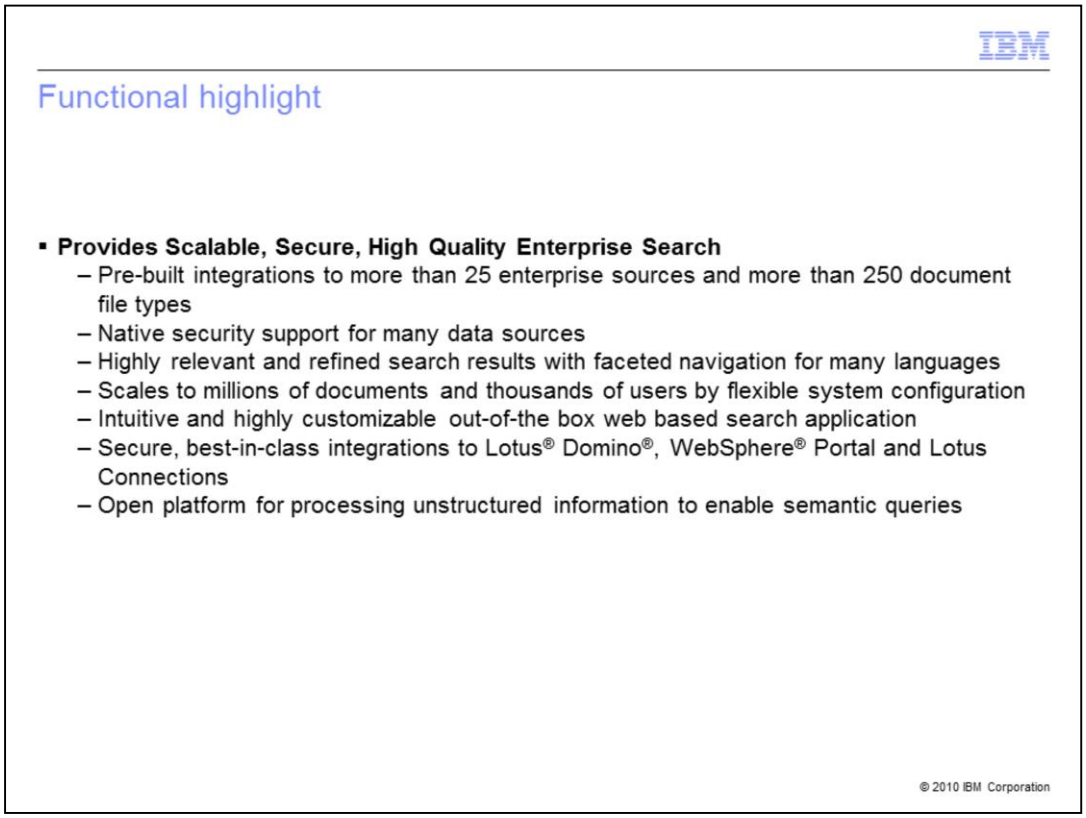


This is the course agenda for this session.

I will start out by talking about what is new in OmniFind V9.1, its benefits and features.

Then I will spend the remainder of the session discussing the various aspects of the overall architecture, the topics of which are listed here.

So let's get started.



Before I talk about what is new in OmniFind, let me briefly go over what it does.

OmniFind provides a high quality enterprise search capability that is both scalable and secure.

OmniFind can crawl and index documents stored in more than 25 different enterprise repositories and more than 250 document file types.

For a majority of those data sources, the native security ACLs are honored by OmniFind.

The results that are returned from a search are highly relevant and can be further filtered easily using faceted navigation for many languages.

As your content grows, so does OmniFind. Scaling to millions of documents and thousands of users by flexible system configuration.

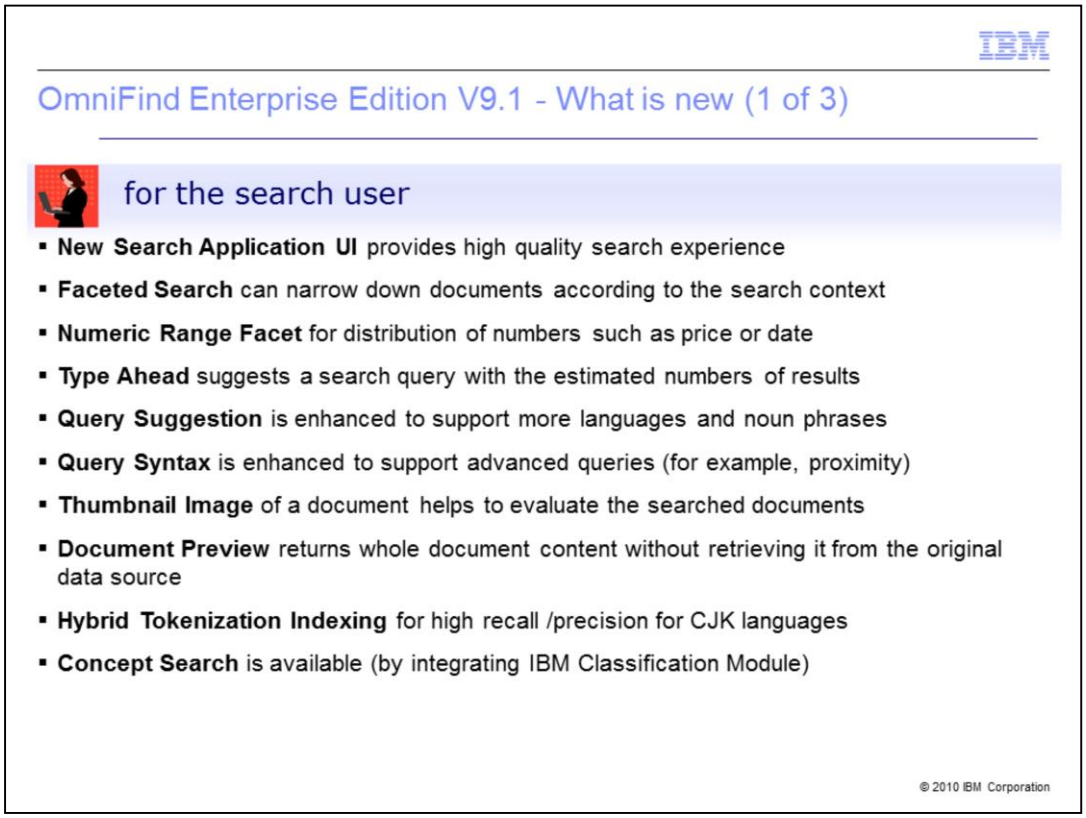
An intuitive and highly customizable ready to use web based search application is provided.

Since this is your enterprise content, you can be rest assured that it remains secure, letting users only search and retrieve documents to which they have access.

Being an IBM product, OmniFind also has best-in-class integrations with Lotus Domino, WebSphere Portal, and Lotus Connections.

And lastly, OmniFind is an open platform for processing unstructured information to enable semantic queries, and you will see this a little later on in the presentation.

So let’s move on to what's new in OmniFind V9.1.



First I am going to review what is new for the end user.

A new and improved search application is provided ready to use as is that is based on Ajax and Dojo providing a rich desktop like search experience.

Incorporated into the new search application is a faceted search that allows you to narrow down documents according to the various dimensions of your data.

As a part of the faceted navigation capability, you have numeric and date range facets that allow you to say things like “show me all the documents between these dates” with a single click. It is also useful for setting up price ranges.

Type Ahead is a new feature that suggests a search query with the estimated numbers of results based on what you are typing into the search box.

Query Suggestion is enhanced to support more languages and noun phrases.

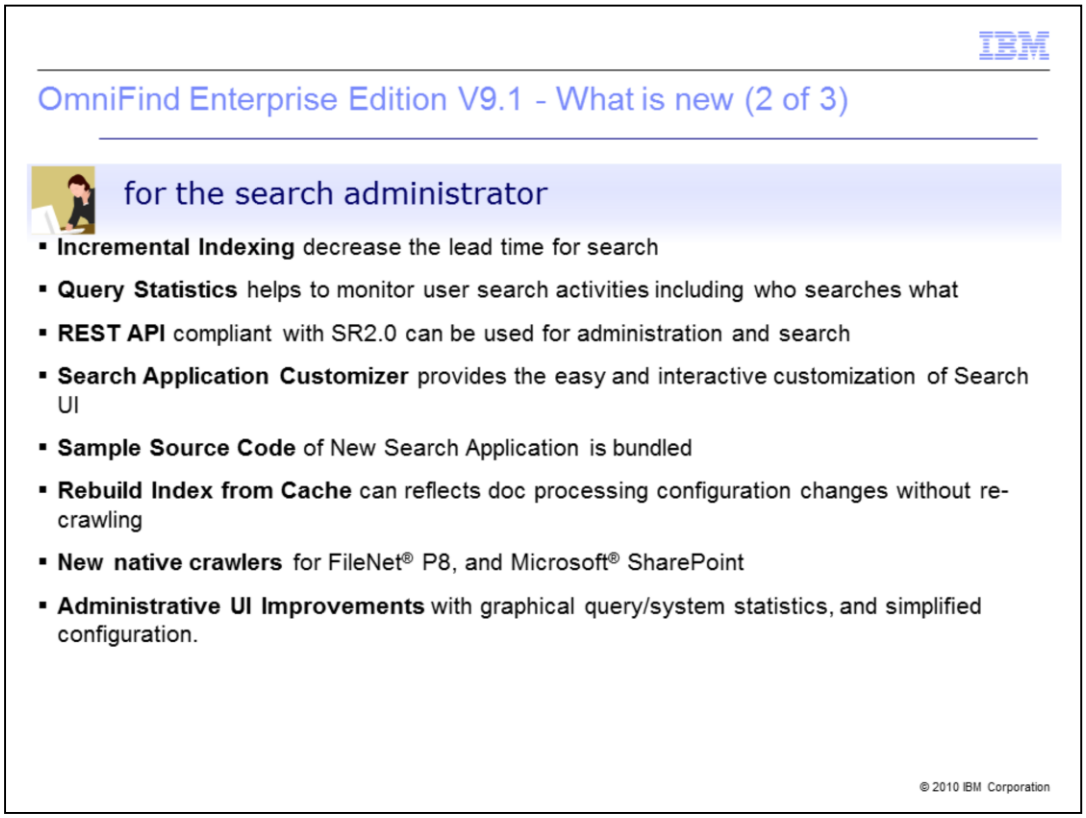
The query syntax has also been enhanced to support advanced queries (for example, proximity searches, X and Y in the same sentence).

A thumbnail Image of a search result document helps to evaluate the searched documents.

Document Preview returns the whole document content without retrieving it from the original data source.

And a new hybrid tokenization indexing scheme has been employed that increases precision and recall for Chinese, Japanese, and Korean languages.

And lastly with the integration of the IBM Classification Module, you can now perform concept based searches whereby the words used in the query do not necessarily need to be found in the document but rather the document is indeed about the concept you expressed.



Now let's take a look at what is new for the search administrator.

First OmniFind now supports incremental indexing which decreases the lead time for search. As each index increment is completed it is made available for search. No longer do you need to wait for the entire build to complete.

More robust query statistics are provided to help you monitor user search activities including who searches what.

Also a new REST API is available as an alternative to using SIAPI for building customized administration and search applications.

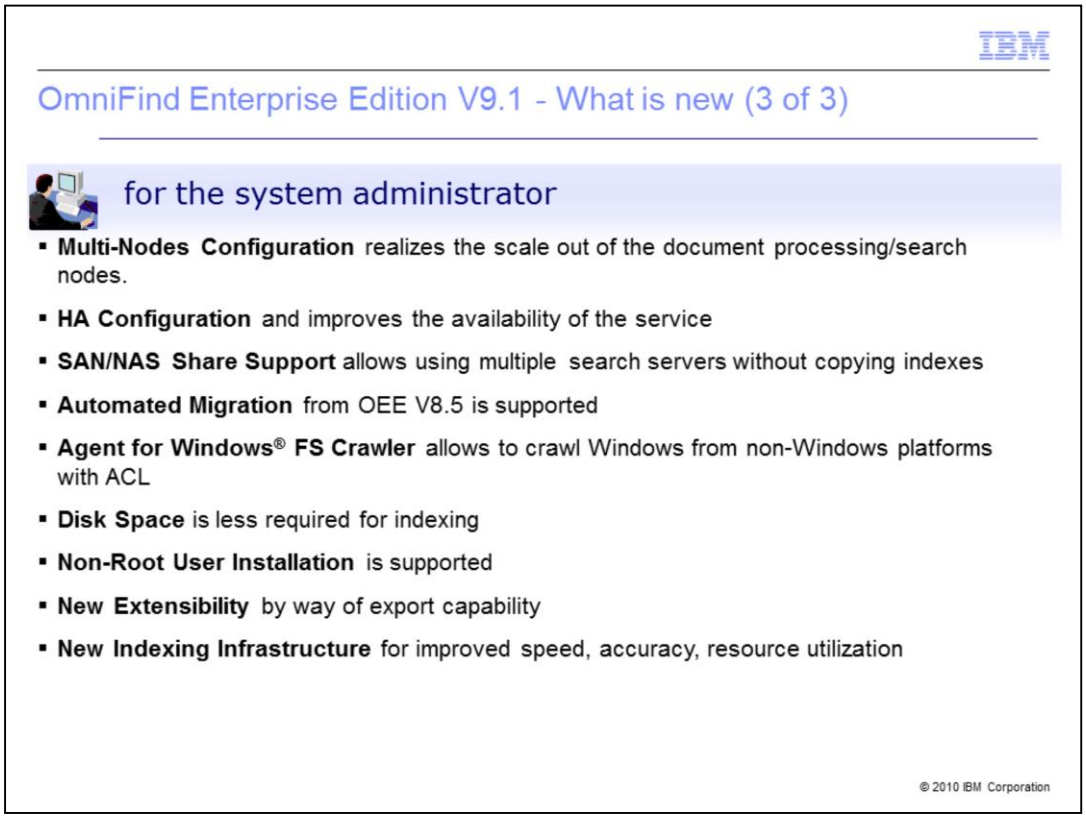
As with the old search application, a search application customizer is also available to provide easy and interactive customization of the new Search Application.

Source code of the new Search Application is also bundled with the product so that you can customize the application any way you want.

A new option to rebuild the index from a document cache avoids recrawling documents when only configuration changes have been made.

New native crawlers for FileNet P8, and Microsoft SharePoint are available.

And general improvements to the administrative user interface with graphical query and system statistics, simplified configuration, and much more.



And lastly, some new features for the System Administrator.

Multiple servers are now supported for the scale out of the document processing function and search function: Two of the typically most resource intensive operations.

High availability configuration is now supported to help guarantee 24x7 operation of service.

SAN/NAS share support allows using multiple search servers without copying the indexes.

An automated migration utility from OEE V8.5 is now supported.

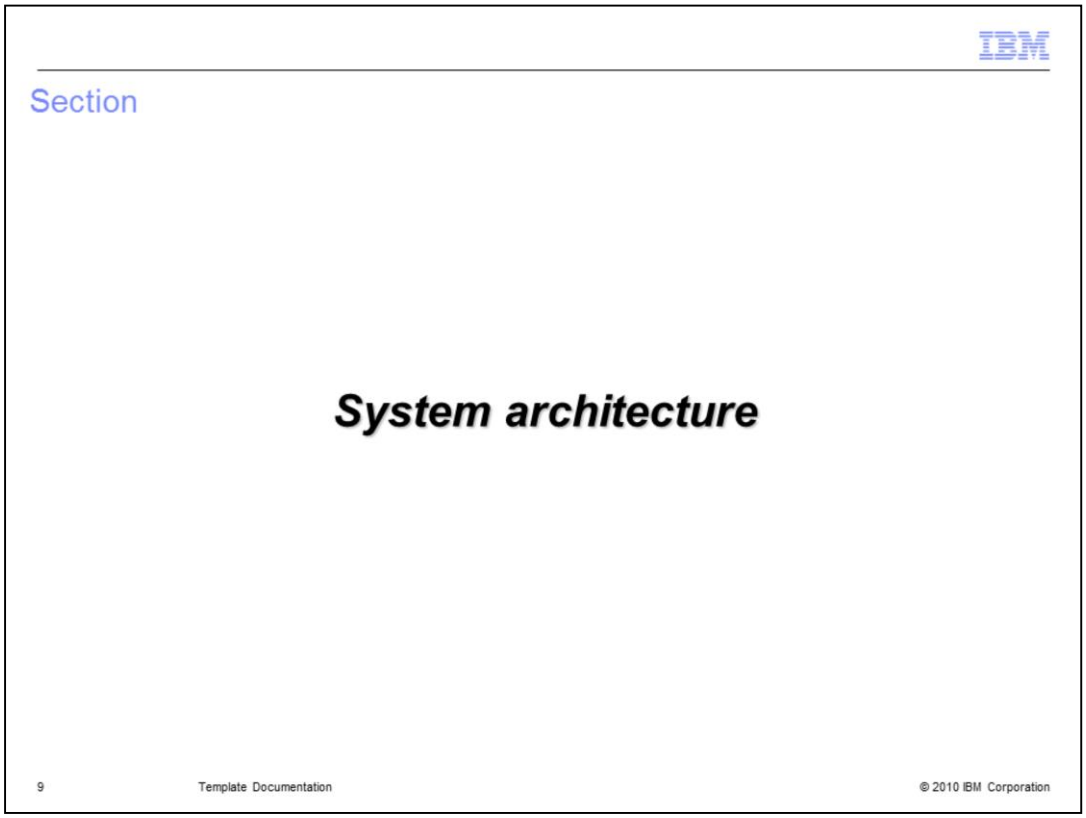
The Windows file system crawler can now be installed remotely onto a windows machine allowing to crawl Windows file systems from non-Windows platforms with ACL.

Indexing now requires a dramatic reduction in disk space.

Non-root user installation is supported for UNIX® based platforms.

Documents and search results can now be exported out of OmniFind to the file system for use by other applications.

Finally, but just as important, a new indexing Infrastructure has been adopted for improved speed, accuracy, resource utilization.



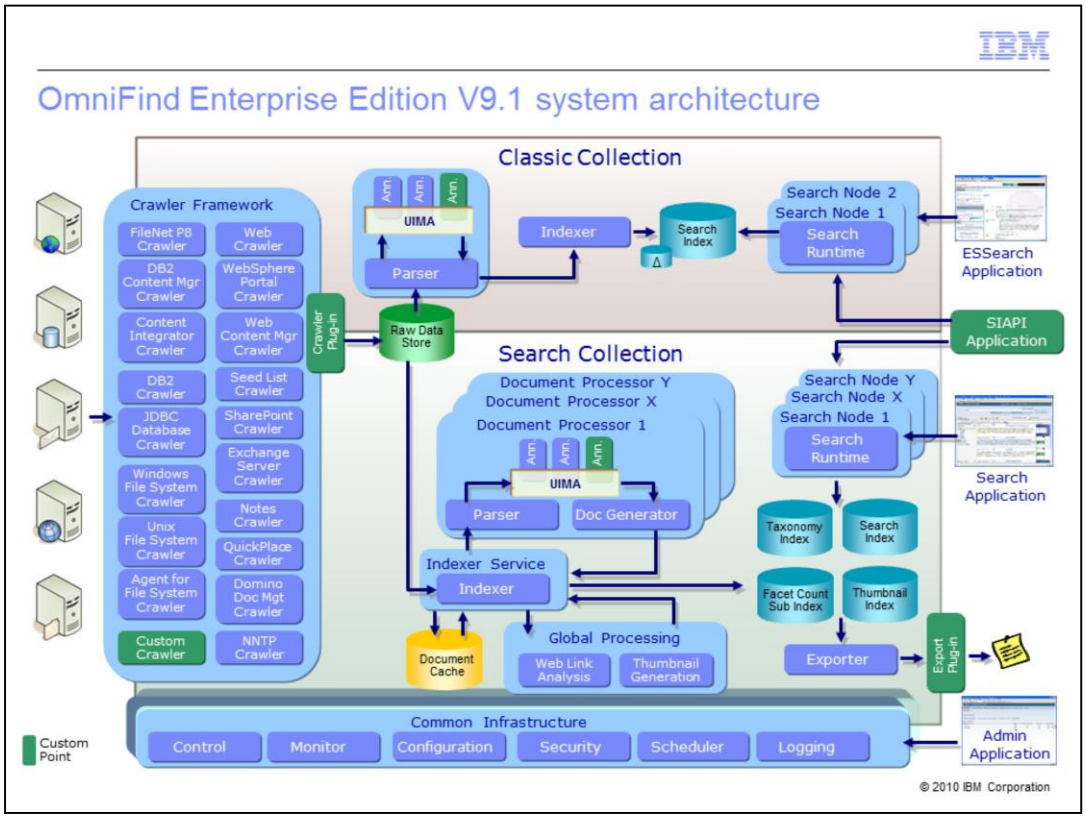
And that does it for what is new in OmniFind V9.1.

You can see that there is quite a lot of new function and capability.

You will learn more about these new functions later in this training series.

Now on to the system architecture of OmniFind.

A lot has changed and been added so let's get started



While a little busy this chart shows the overall system architecture of OmniFind V9.1.

It seems complex because there are actually two architectures. One supporting the original indexing scheme. The second supporting the new and improved indexing scheme.

The components in the shaded gray box at the top labeled Classic Collection is the previous architecture employed for OmniFind versions prior to 9.1.

The components in the light blue shaded box in the center labeled “Search Collection” depict the new indexing scheme.

The new indexing scheme based on open source technology has been adopted for improved speed, accuracy, resource utilization.

Both the old and new indexing architectures use UIMA for applying text analytics, the same crawler framework for accessing content, and the same overall administrative infrastructure for control, monitoring, configuration, and more.

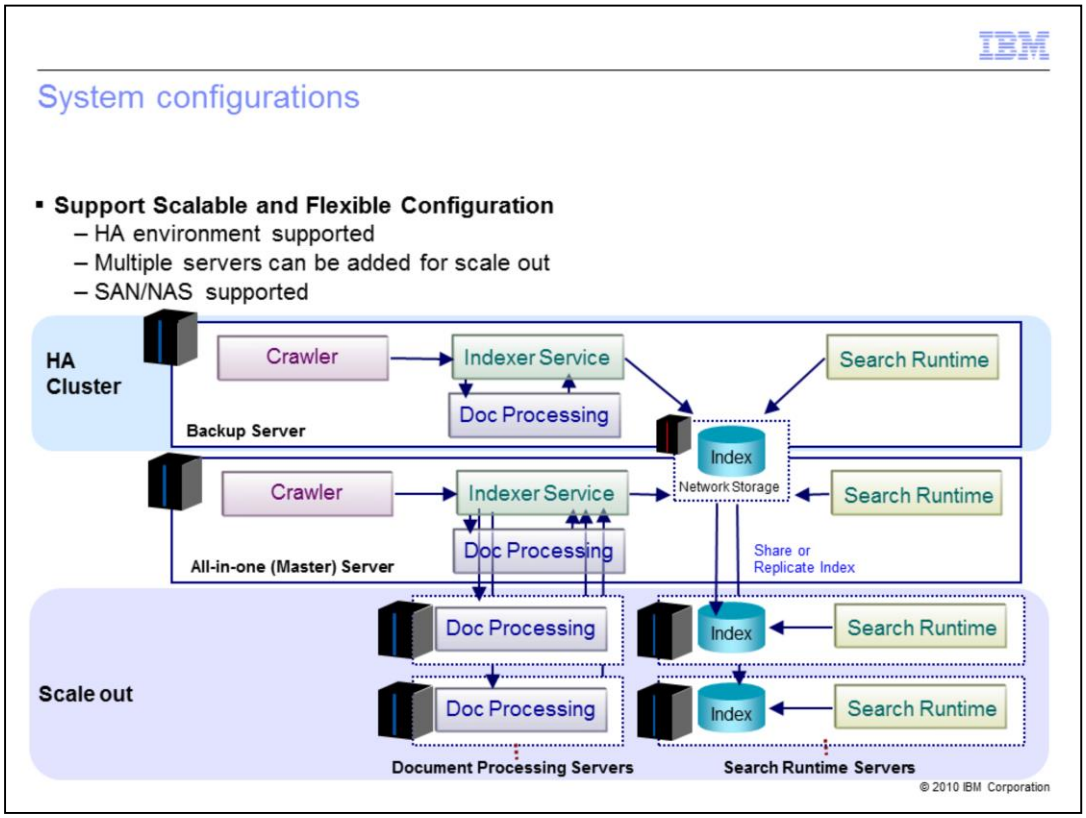
Note that SIAPI can also be used for both types of collections so your customized search applications will not need to change.

The old ESSearchApplication is still supported but only works with the classic collections.

A new Ajax/Dojo based search application is provided with new search features but only works with the new search collection type.

In the new collection type architecture, there are quite a few enhancements which I will review in upcoming charts. But from this diagram you can see the implementation of a Document Cache that lets you rebuild an index for configuration changes without recrawling.

And you also notice the capability to export documents and search results for consumption by external applications.



Now we are going to take a closer look at the system architecture for the new search collection type

This diagram shows the three major enhancements namely:

High availability support.

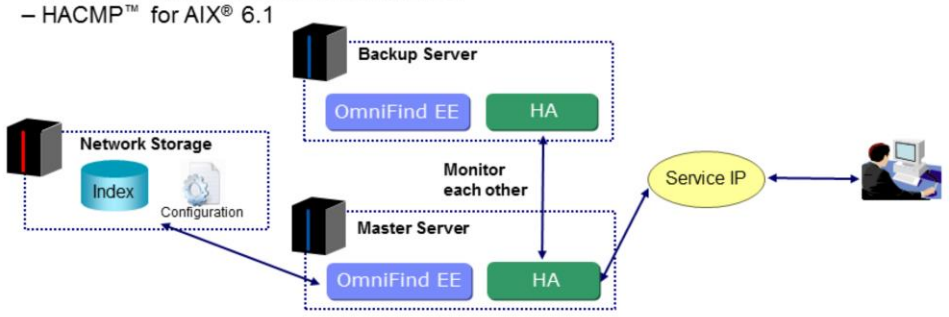
Improved scale out for document processing and search runtimes.

And SAN and NAS shared drive support.

So let's take a closer look at each of these.

HA environment support

- **Takeover failed server by HA software within few minuets automatically without any user interaction**
  - All search and index activity will be handed over switching
  - Running crawlers are restarted on switched node
  - It is not started after switched to avoid exceeding expected duration
- **Supported Environment**
  - Microsoft Cluster server 2003 and 2008
  - HACMP™ for AIX® 6.1



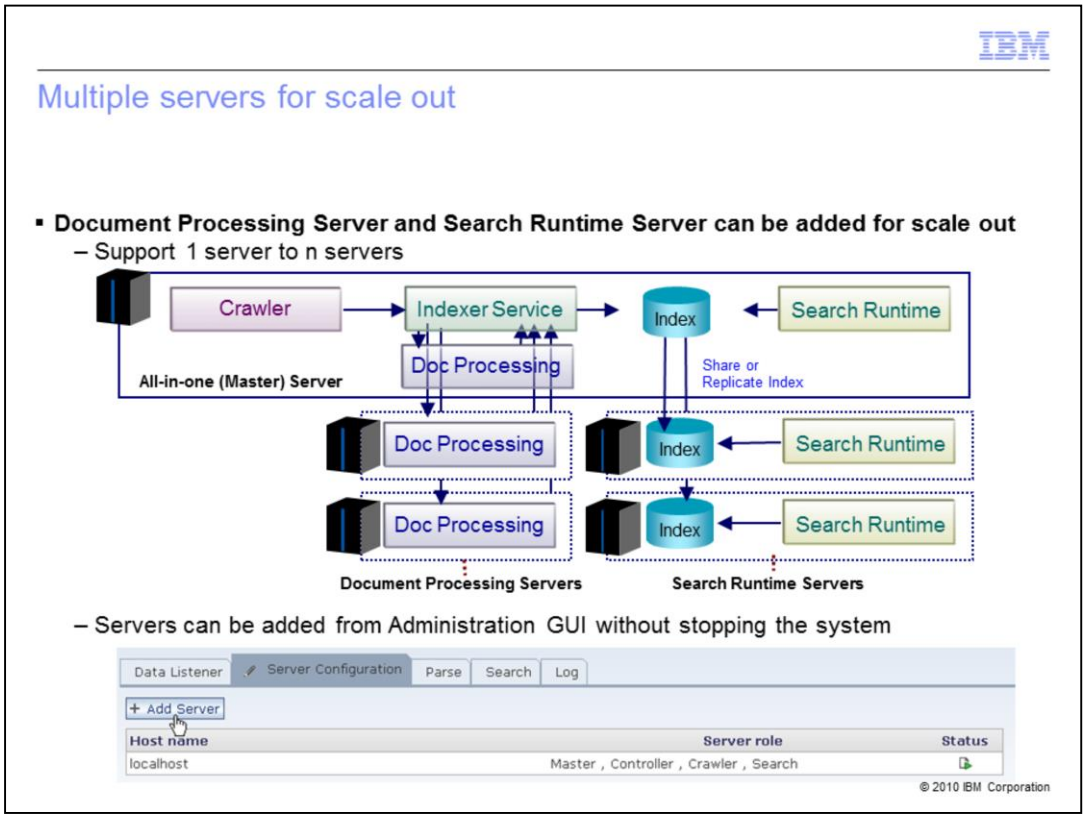
High availability helps ensure 24x7 operation in the case of a failed server in your environment.

In an HA environment, you have a duplicate environment of your OmniFind setup on standby ready to go in the case of a detected failure in your production environment.

The HA software will automatically switch between the two environments when the failure is detected.

Previously running crawlers will be restarted on the switched node.

Note that high availability is only supported on Windows Server 2003 and 2008, and AIX using HACMP.



If you remember in OmniFind 8.5, the system configurations supported were 1-2-4 servers, two servers for search and one for indexing.

Now with the new architecture and indexing scheme you can have any number of servers for your search runtimes and multiple servers for document processing.

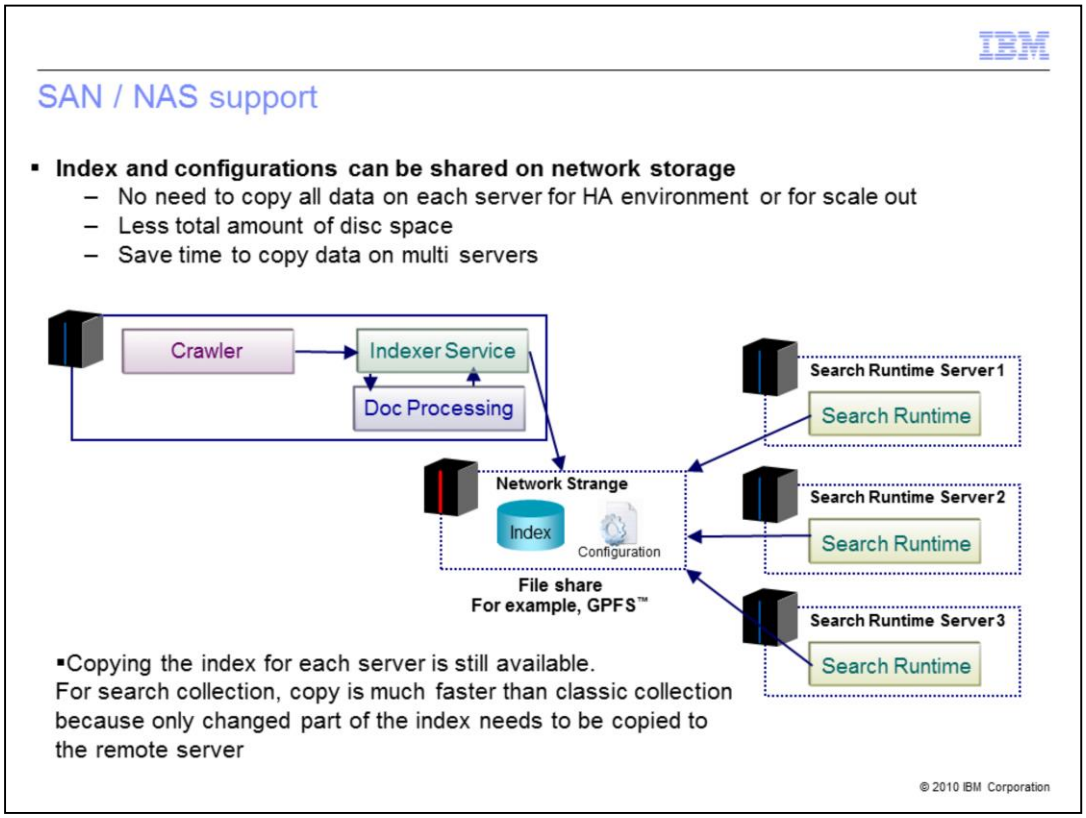
For each search runtime sever added to the system, a copy of the index is made to that server. We will talk about shared disk in just a moment.

In the old architecture, the indexing component and the document processing UIMA pipeline were inseparable and had to run on the same server.

Now in the new architecture you have separated out the document processing component to one or more servers to handle the increased load of the text analytics being applied.

The main index remains on the master server with the documents to be processed and distributed in a round robin fashion by the master server.

Note that for both types of scale out, you can add these servers while the system is running. There is no need to restart the system. The system will automatically detect the presence of the newly added server and start using it.

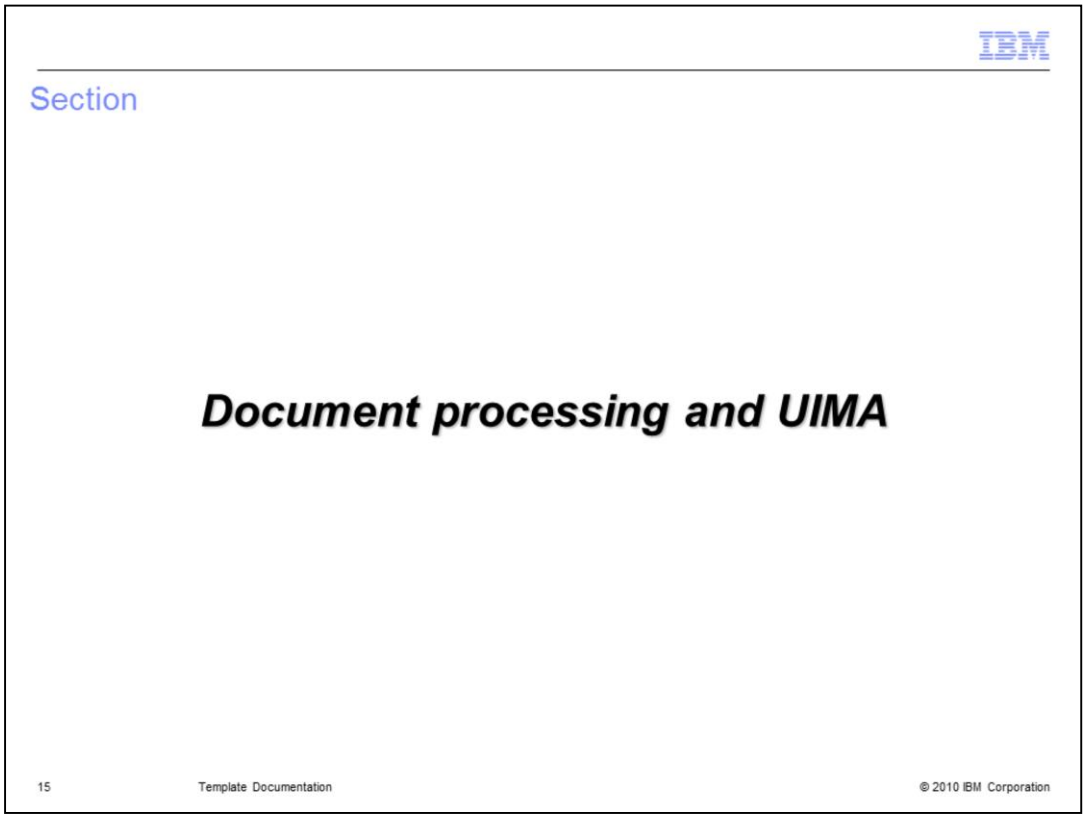


With the new architecture, we have introduced support for SAN and NAS which allows the index to be shared across multiple servers, thus saving precious disk space.

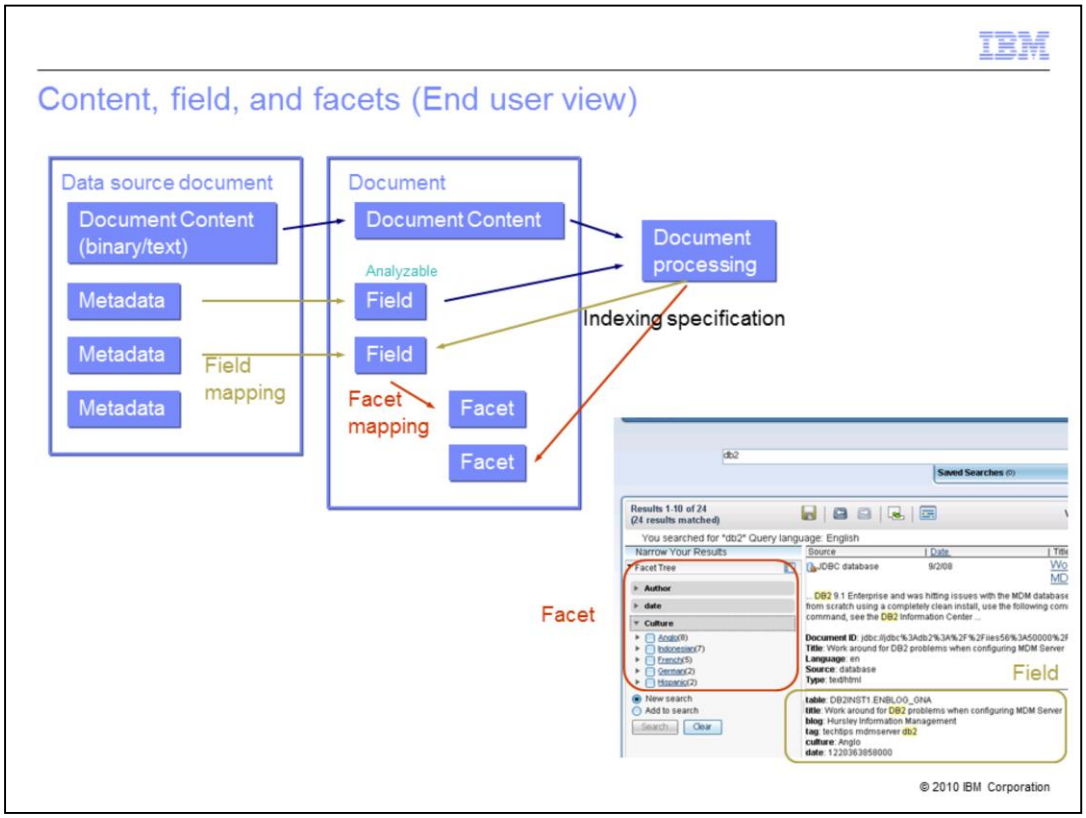
So with scaled out search servers this can be a big savings as well as saving time since an index copy is no longer required.

Note that copying the index for each server is still available.
For search collection, the copy operation is much faster than the classic collection. This is because only the changed part of the index needs to be copied to the remote server.

SAN/NAS is required for high availability.



Now let’s take a closer look at the changes to document processing and UIMA.



Now the logical document model within OmniFind has been enhanced with the concept of facets.

Facets are used to filter and narrow down documents in the new search application and in this way perform as a kind of guide navigation through your search results.

Facets are populated with values from search fields that are mapped to the facet.

Facets can also be populated directly from a text analytic annotator in the UIMA document processing pipeline.

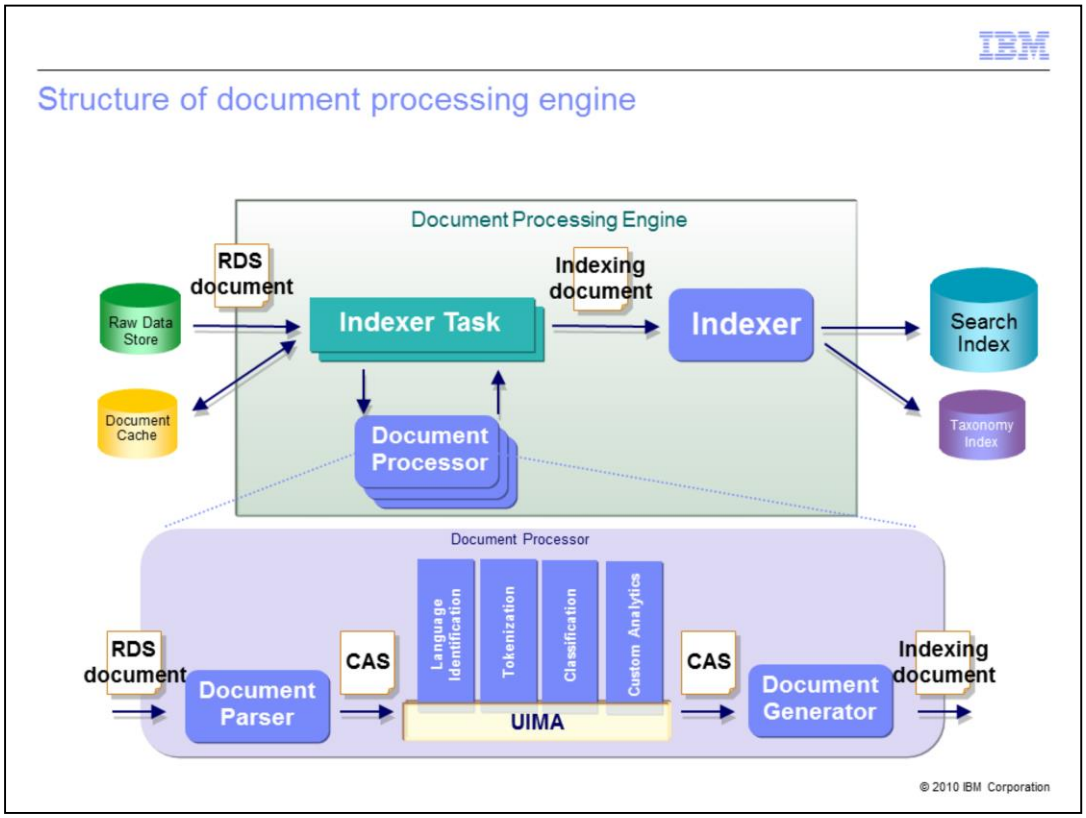
Remember that search fields obtain their values from native fields from crawled documents or from an annotator as well.

So in the screen snapshot shown here we see the search fields listed in the brown rectangle in the details view of the search results.

Facets are shown in a tree fashion in the panel to the left.

The facets and their counts are updated for each query submitted reflecting the statistics for the current result set.

So in the example, we searched for documents containing the term “DB2®” in them. Note that there is a field in these documents called “Culture” which was mapped to the facet with the same name. Looking at the expanded facet for culture we can see the number of documents that exist in the current result set for each culture.



As I mentioned before during the scale out discussion, the document processing component has been separated out from the indexing component allowing it to run in its own server.

This chart shows the separation.

The diagram at the bottom is the blow up of what is inside a document processor.

It receives a raw data store document as input.

The document parser extracts the text from the document using Stellent, if necessary, and converts it into a Common Analysis Structure (in XML).

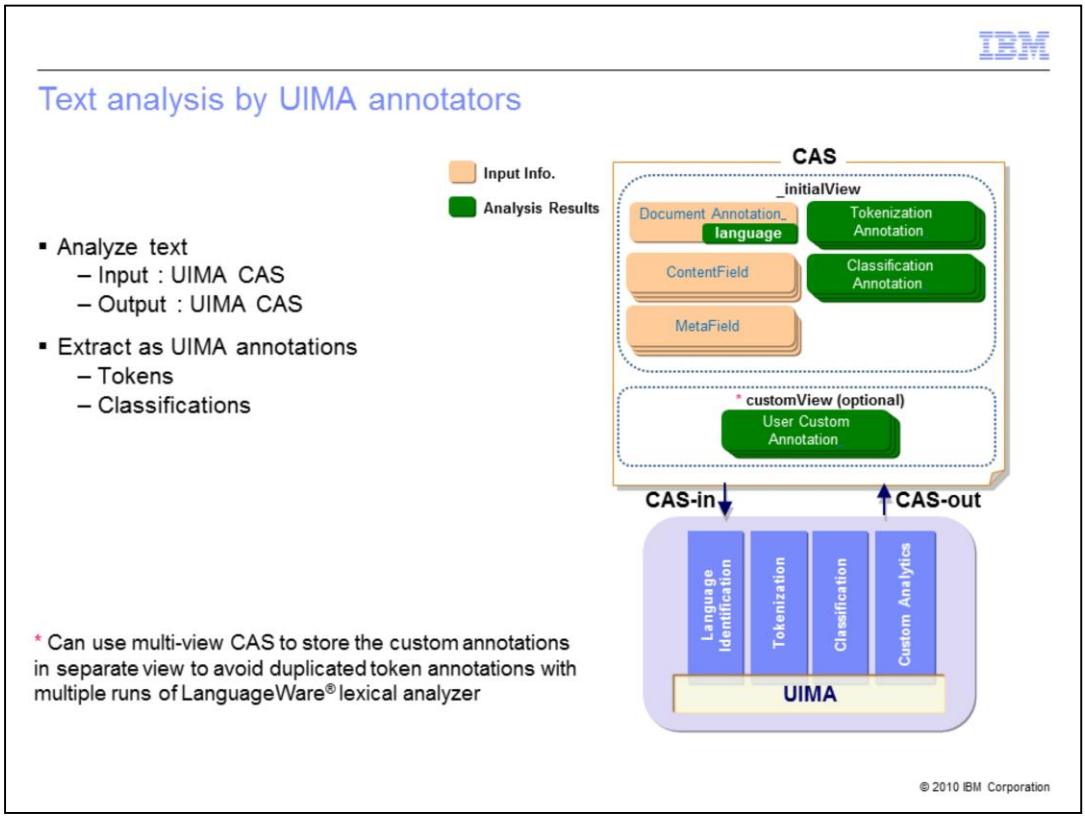
The CAS is then fed through the UIMA pipeline as before.

Coming out of the UIMA pipeline is the resulting CAS with annotations which is then converted into a Lucene indexing document by the document generator.

As for the indexing component at the top of the chart there are a few new things.

First a document cache has been added that allows you to not have to recrawl documents when only configuration changes have been made to the index. Only a rebuild needs to occur.

Also the index itself is based on the open source Lucene index with IBM extensions. IBM is a committer to Lucene and contributes back selected enhancements to Lucene.



OmniFind has added a new feature to help protect the integrity of the CAS as it moves through the UIMA pipeline.

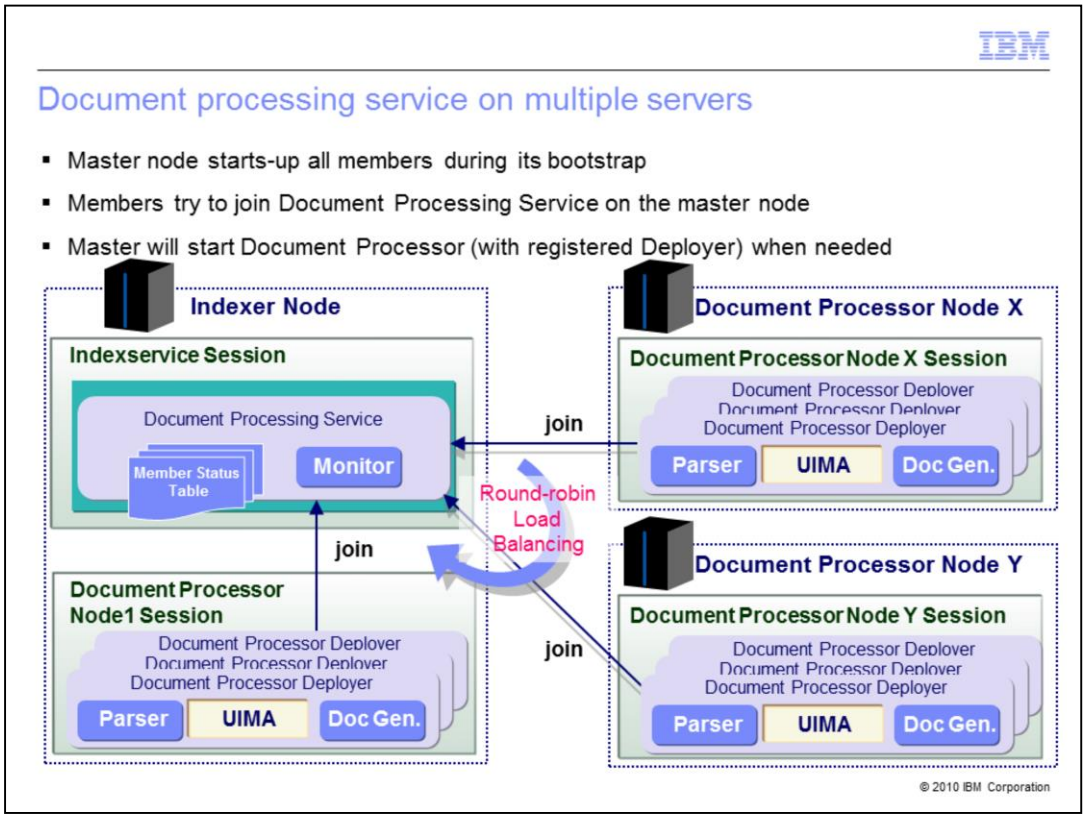
Namely you now can select from two views of the CAS.

The first and initial view allows an annotator to see and potentially alter all annotations made by previous annotators in the pipeline. This is generally useful when you want to use the work done previously productively to say identify the language of the document or reuse parts of speech tagging.

But at the same time this can also allow an annotator to alter previous annotations either maliciously or accidentally.

To protect against this potential integrity threat, the administrator can select the second or custom view which creates an entirely separate namespace for the annotations.

The advantage is that the custom annotations are now kept separate from the system generated annotations. The disadvantage is that the custom annotator will not be able to see or take advantage of any previous work and annotations done by the system.



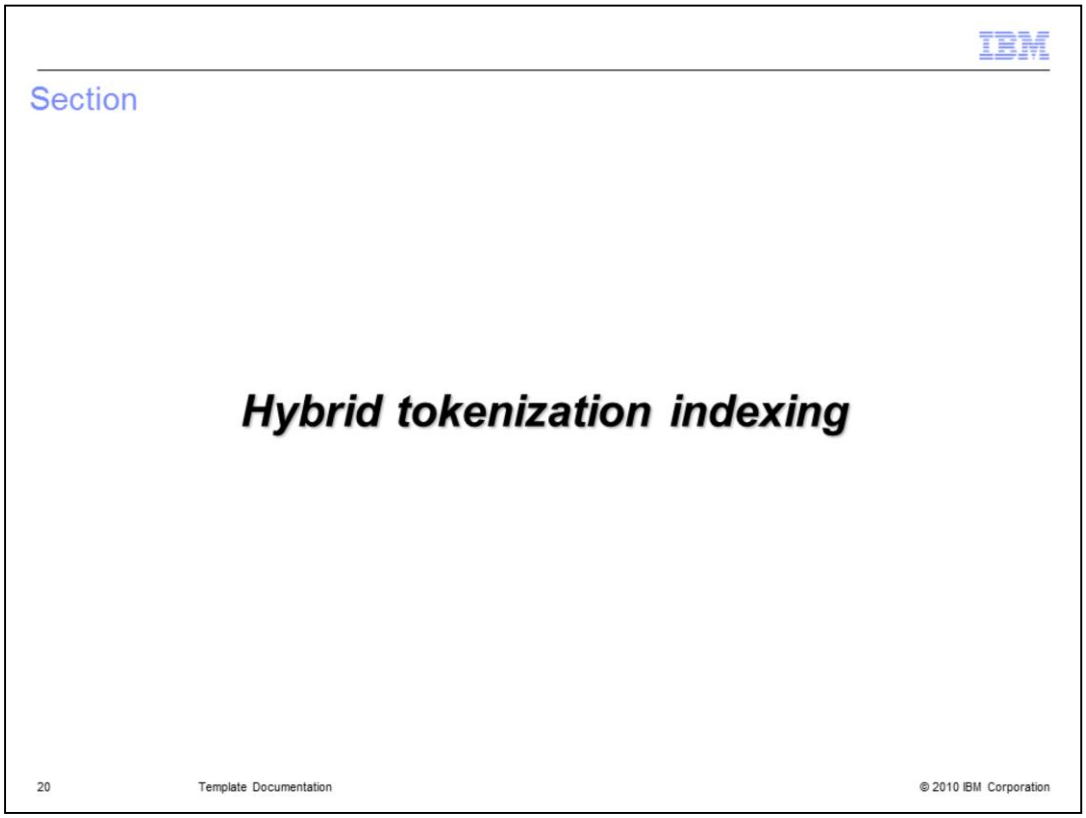
And as I previously mentioned, the document processing service can now run on any number of separate servers for load balancing.

The Master node will start-up all document processor services and servers at system start-up time.

You can add new document processor servers while the system is up and running.

Once registered, the Master server will start the document processor on the new server when needed.

The load balancing is done in a round robin fashion.



So now let's take a look at the new hybrid tokenization scheme for indexing.

Hybrid tokenization indexing

- Both morphological analysis and N-gram segmentation can be enabled *within a single collection* for *good precision* and *high recall*
  - High quality search with linguistic capabilities
    - It will help users to get more precious results, by searching for linguistically equivalent words, such as variant and inflected forms.
  - High recall search with language-neutral rules of segmentation
    - Even if morphological analysis does not determine words as expected (this often occurs for some languages), users can still get some results if documents have exactly the same character sequences as a given query.
- Effective for Asian languages (Chinese, Japanese, Korean)

In previous versions of OmniFind you had to choose whether morphological analysis or n-gram segmentation were to be used for the entire collection. It was one or the other.

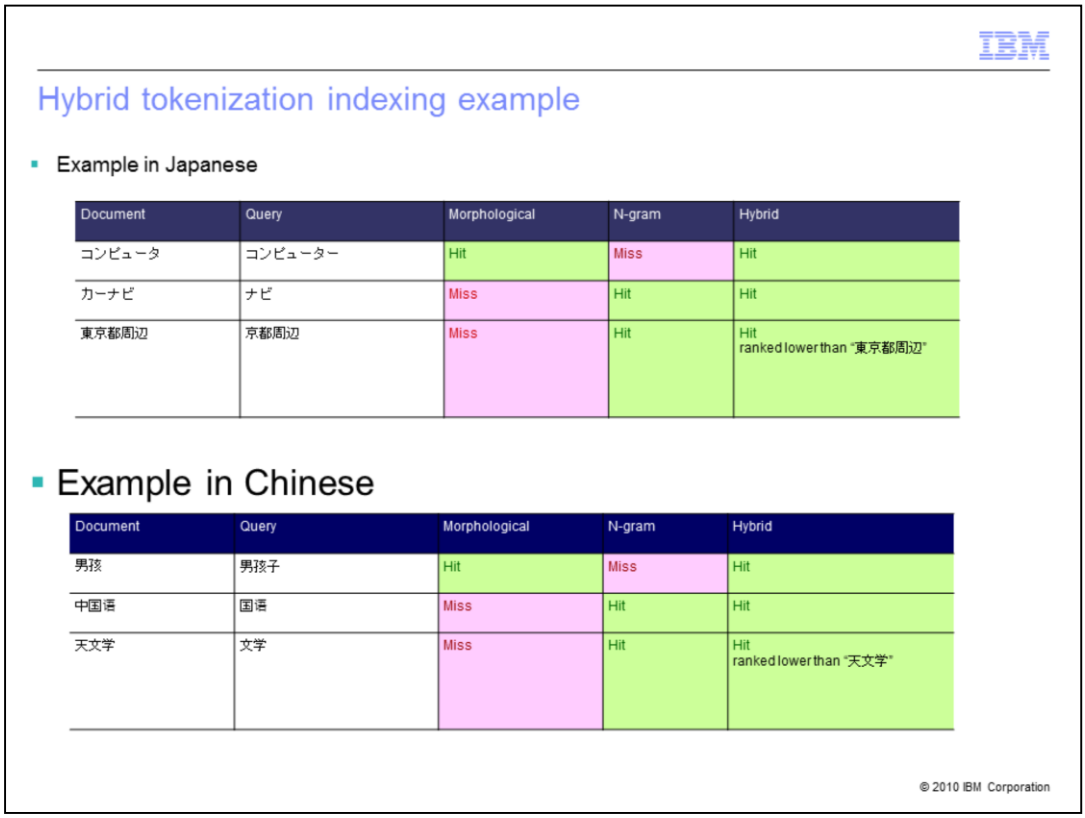
Morphological analysis among other things uses white space characters to identify words in text and to then break them down into their root form or lemmas.

N-gram is the technology used for parsing double byte languages where words are symbols with no white space characters to separate them.

Now both methods can be employed in a single collection and is referred to as hybrid tokenization.

The benefit is improved precision and recall of the search results.

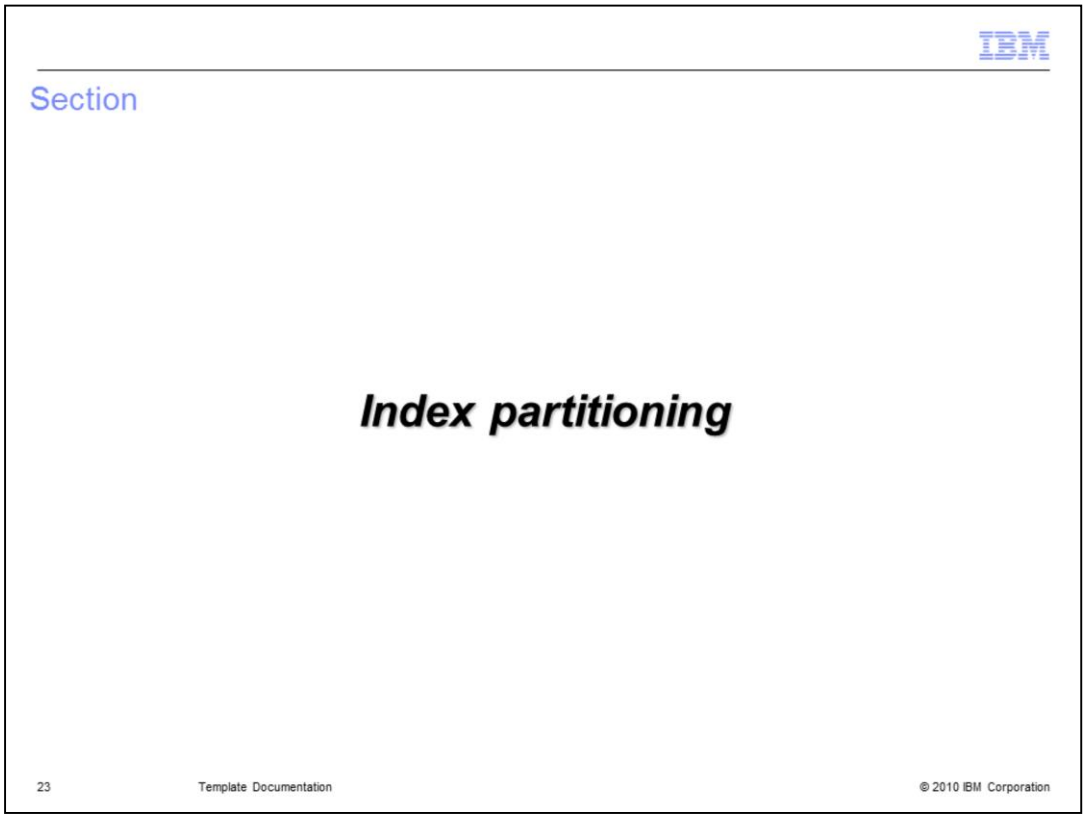
It is most effective for Asian languages such as Chinese, Japanese, and Korean.



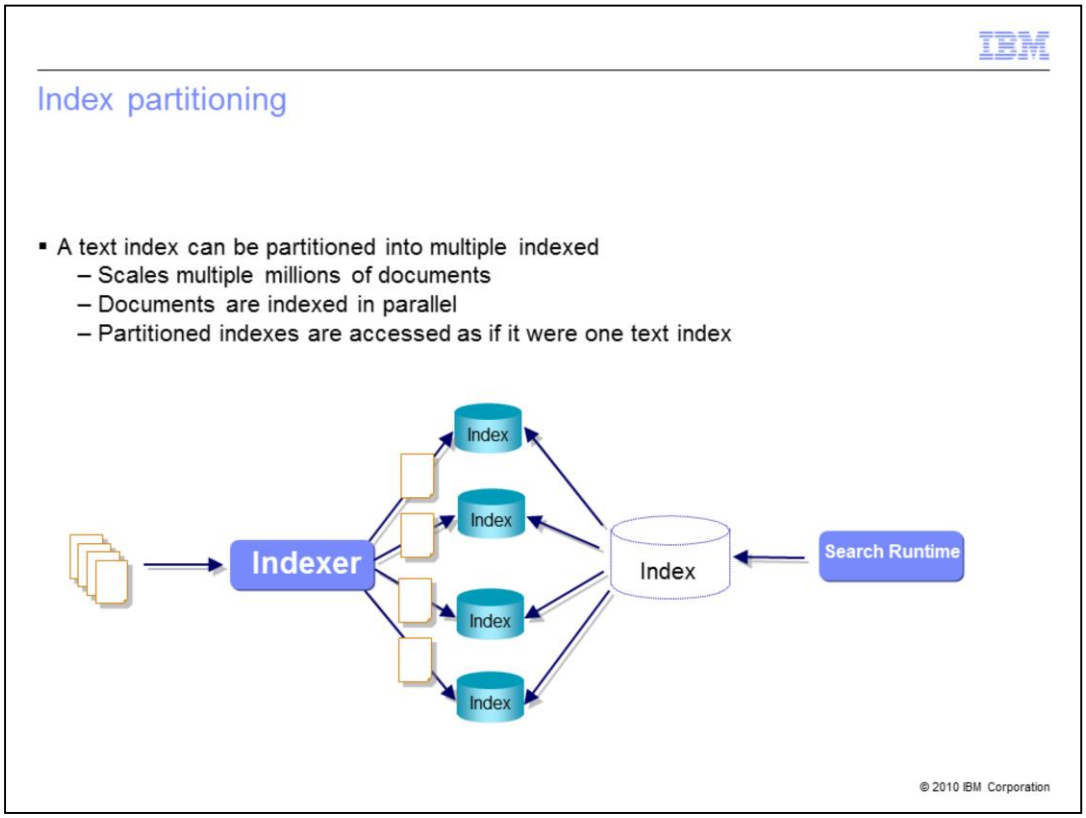
Here are some examples where the precision is improved.

Now these examples are in Japanese and Chinese which I do not expect you to understand.

But if you look at the hit and miss indicators for each type you see that the hybrid approach results in the correct match for all cases.



Now let’s take a look at index partitioning.



Index partitioning is the practice whereby a logical collection can be partitioned into multiple indexes.

This is useful when the number of documents in an index exceeds approximately 20 million, the point at which a physical index starts to be constrained by system resources.

A partitioned index (and the number of partitions) must be planned for and configured at collection creation time. The parameters of which cannot be changed once created.

Documents are indexed in parallel and distributed to their appropriate partition based on a URI hashing algorithm.

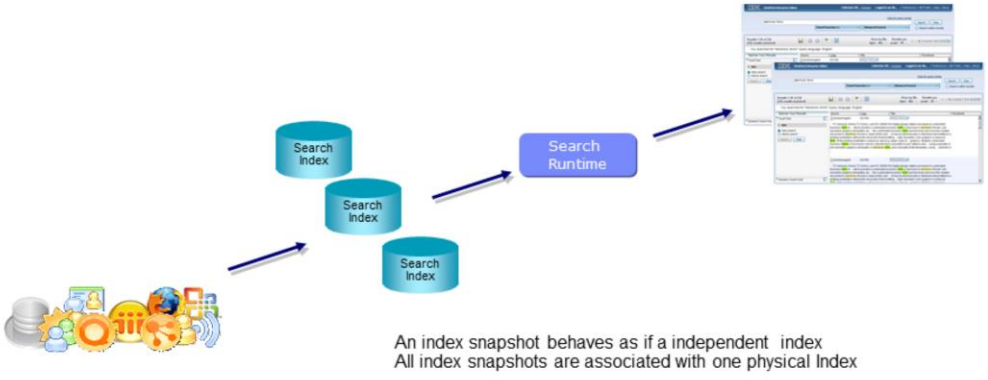
Federation is used to search the multiple partitions in parallel aggregating the results into a single results set.

So from the end user perspective it appears as a single logical collection.

Now let's take a look at some improvements gained in the search runtime.

Search the latest documents with short latency

- Newly inserted/updated documents are reflected to search results incrementally in a short period
  - The reorganization of index is not needed
  - Search Runtime detects index updates in background



An index snapshot behaves as if a independent index
All index snapshots are associated with one physical Index

One of the benefits of using the Lucene index as the base for OmniFind's new collection type is that newly inserted or updated documents are reflected in the search results in a much shorter period of time.

No longer do you need to wait for the entire collection to be built.

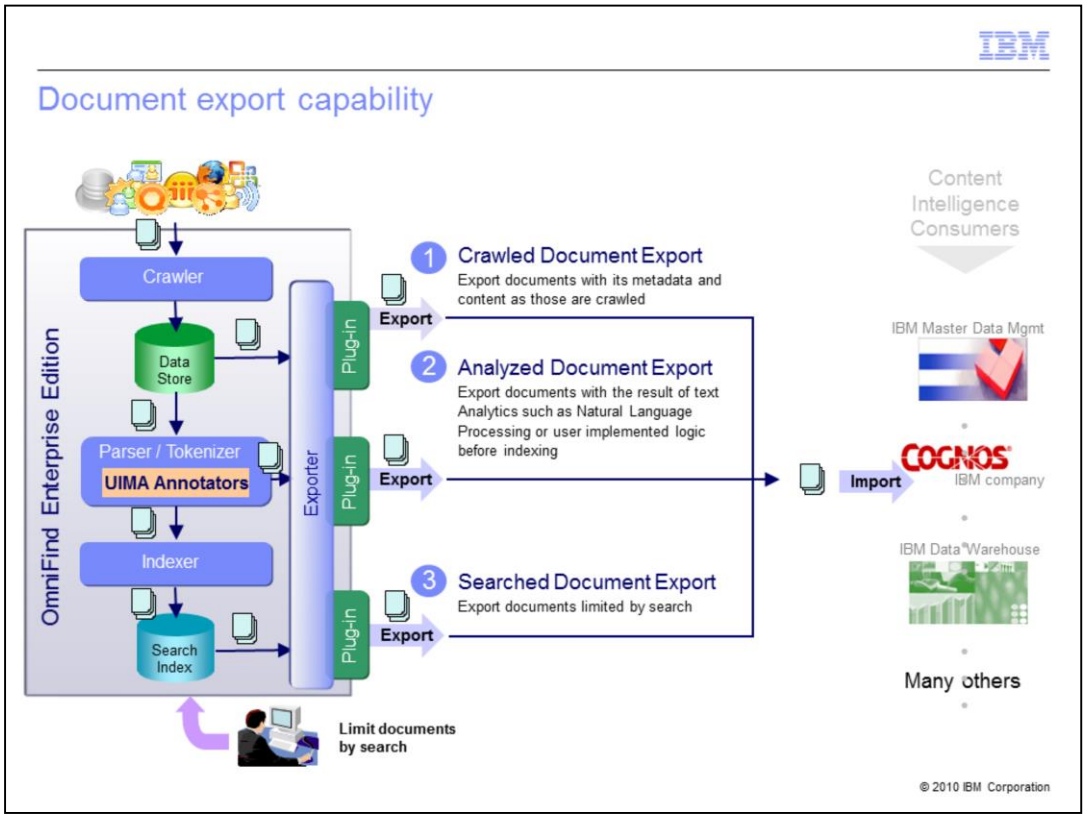
This is because Lucene uses an incremental indexing strategy.

As soon as an index increment reaches a certain size it is closed and a new increment starts.

The closed index is immutable so it can be search right away.

The search runtimes detect these index updates in the background and make the newly “closed” increments available for search when ready.

Now let's talk about document export.



OmniFind now provides a document export capability that allows other external tools and applications to use the results of your search.

Actually you can export more than just your search results.

There are three export points as shown in this chart where export can occur.

The first is after documents have been crawled. Here the content of the crawled document, its original binary form, and any crawled metadata is exported.

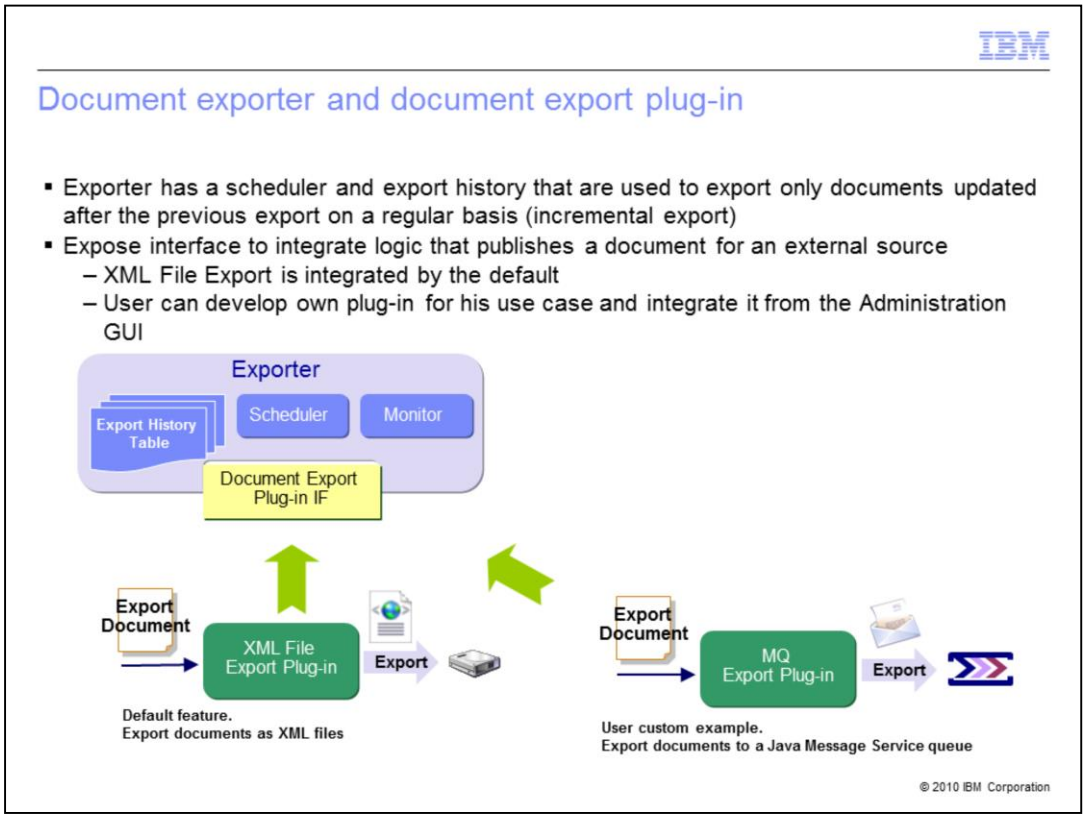
The second point is after the document has been analyzed. Here you are able to also export any annotations that were added by the text analytics.

And lastly you can export the results of a search.

The exported content is deposited in the file system as XML and the binary form of the document stored in a separate file.

Note that exports can be scheduled on a recurring basis.

At each of these points, you can extend the export functionality by writing your own export plug-in module which I will describe in more detail on the next chart.



The exporter is the OmniFind component that keeps track of the number and time of scheduled exports and is responsible for invoking the appropriate export plug-in to perform the actual export.

An export history is maintained so that only new and updated documents since the previous export run are exported. This is a kind of incremental export.

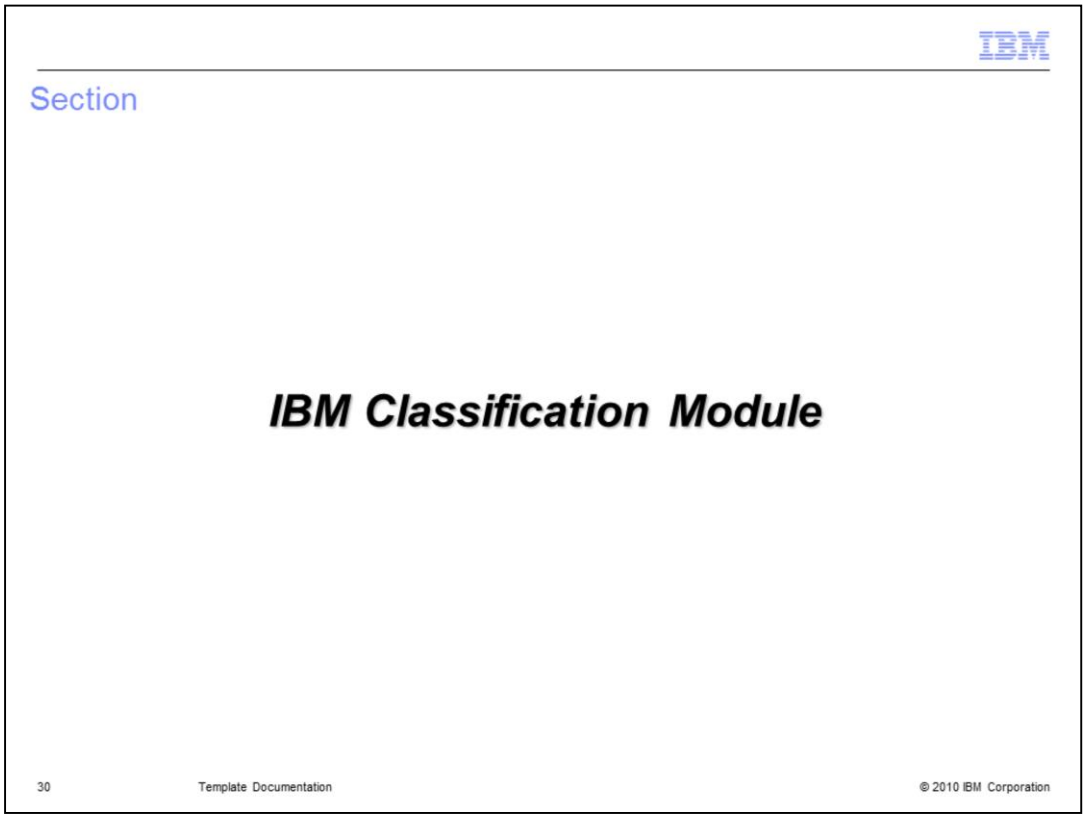
By default OmniFind invokes the XML file system export plug-in to export the documents as shown in the bottom left of the chart.

But you replace this export plug-in with your own logic written in Java™ that publishes the documents to an external source.

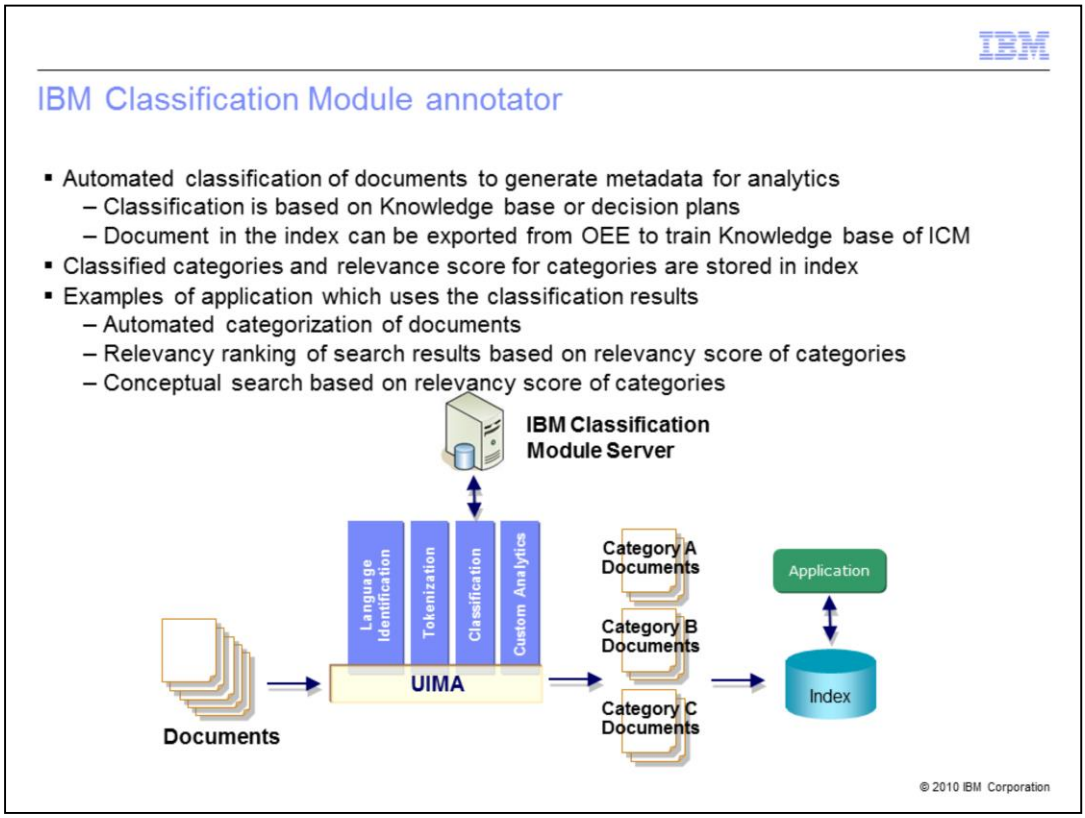
In the lower right corner of the chart shows an example custom plug-in that exports the documents to MQ Series. Or you might write one that exports the documents and metadata into a relational database.

Different plug-ins can be used at the three stages identified earlier.

And different plug-ins can be used across different collections.



And lastly, I am going to talk about the IBM Classification Module and its tighter integration with OmniFind.



The IBM Classification Module is primarily used to automatically classify your documents based on sample documents it had been trained on and associated to administrator defined categories.

Documents in the collection can be searched for and exported to assist in building your training set for ICM.

In ICM you build your knowledge base and decision plan based on those training documents.

Once built, the prepackaged ICM annotator can be configured to use that Knowledge Base and Decision Plan in the UIMA pipeline. The result being documents annotated with a category metadata field.

You also have the option of storing the associated score with the category field into the OmniFind index.

Besides the benefit of automated classification of documents just described, there are two other capabilities that can be optionally enabled.

The ranking score of a search result document can now be augmented with its categorization score by how closely it matches the query categorization score. In this case it is better to use clustering on the documents to build the knowledge base.

The next benefit is that it can enable conceptual search determined by how well the category generated by the query matches the category assigned to the document. Again clustering is more preferable here.

The advantage of concept based searching is that the words in your query do not necessarily need to be found in the document.

You have completed this course and now can:

Describe the major new features and functions of OmniFind V9.1.

Explain the new architecture of OmniFind and its supported configurations.

Describe the benefits of IBM's Classification Module tighter integration with OmniFind V9.1.

Trademarks, disclaimer, and copyright information