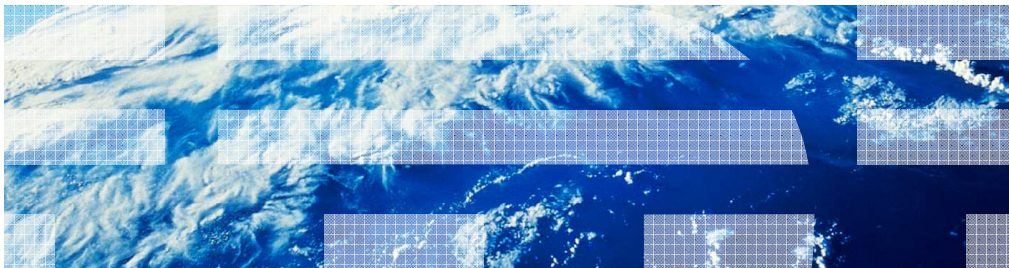

WebSphere Commerce V7 Feature Pack 2

Search overview and architecture



This presentation provides an introduction to the WebSphere Commerce Search solution capabilities and architecture.

Table of contents

- Solution overview
- Architecture overview
- Customization

This presentation begins with an overview of the WebSphere Commerce search solution capabilities. Following the overview is a discussion of the solution architecture. This presentation concludes with some example customization scenarios.

Solution overview

This section introduces the WebSphere Commerce search solution.

Storefront search

- Full-text search
- Rich feature set
 - Auto suggest
 - Spell correction
 - Phrase search
 - Wildcards
 - Advanced search

Search Results

Your search for "cofee" produced 0 results.

Did you mean: [coffee](#) [come](#) [cover](#) [cafe](#) [coffeecup](#)

You can refine your search using the [Advanced search](#) options.

Your search for **coffee** produced 22 results.

Suggested words
are present in the
index

col
c
cotton
coffee
combed
comfy
cozy
cool
contemporary
corduroy
comfort
coffeemaker

Within the storefront, several new features are available. Some examples are shown on this slide including spell correction and auto suggest for search terms. Shoppers can search on single or multi-word phrases and search terms can include wildcards. An advanced search page is available to specify additional search constraints.

Basic search results

The screenshot shows a web page with a navigation bar at the top containing 'Furniture', 'Tableware', 'Kitchenware', and 'Apparel'. A 'United States E' dropdown menu is visible on the right. Below the navigation bar is a 'Customer Support' link with a person icon. The main content area is titled 'Search Results' and states: 'Your search for coffee maker produced 36 results. You can refine your search using the Advanced search options.' Below this, a pagination bar shows 'Displaying products 13 - 24 of 36' and navigation arrows. Three product listings are shown:

Product Name	Description	Price	Action
Coffeecup with a Silver Trim	A coffeecup with a silver trim.	\$4.99	Add to Cart
Digital 12 cup Coffee Maker, Blue	This blue stainless steel coffee maker will compliment any modern kitchen.	\$99.99	Add to Cart
Digital 12 cup Coffee Maker, Green	This green stainless steel coffee maker will compliment any modern kitchen.	\$99.99	Add to Cart

At the bottom of the page, there is a page number '5', the text 'Search overview and architecture', and the copyright notice '© 2011 IBM Corporation'.

This screen capture shows the search results page for the Madisons store without WebSphere Commerce search enabled. This search runs against the WebSphere Commerce database and returns matching products results only.

The screenshot shows a search results page for 'coffee' on a WebSphere Commerce site. The page features a navigation bar with categories like Furniture, Tableware, and Kitchenware. A breadcrumb trail at the top indicates the path: Home | coffee | Coffee Makers. On the left, there are facets for Brand (Sharpson, Kitchen's Best, AromaStar, Enzi) and Price (Less than 100, Between 100 and 200, Between 400 and 500). The main content area displays search results for 'coffee', showing three products: 'Coffee and Espresso Bar' (\$499.95), 'AromaStar Deluxe' (\$79.99), and 'AromaStar Steam Frother System' (\$149.99). Each product listing includes a small image, the product name, a short description, and an 'Add to Cart' button. The search term 'coffee' is highlighted in red in the product descriptions. Additionally, there are links to user manuals for the products, which are highlighted as unstructured content. A 'Customer Support' link is also visible on the left side of the page.

This screen capture shows the Madisons search results page when WebSphere Commerce search is used. In addition to the list of matching products, various facets are also returned from the search query. This facet information makes it easy for the shopper to narrow down the initial results to find what they are interested in. Facets previously selected by the shopper are shown in the breadcrumb trail and can be removed to return to more general search results. Highlighted search terms allow shoppers to quickly identify where in the product name or short description their search term occurs. The WebSphere Commerce integrated search also locates unstructured content such as user manuals and includes those documents in the search results list view.

Search results summary

- Scoped by
 - Store catalog
 - Store path (extended sites)
 - Contract entitlement
- Faceted search results
- Search term highlighting
- Interactive bread crumb trail
- Multiple views
 - Grid view
 - Details view
- Unstructured content results in details view
- Displayed price can be indexed or calculated
- Configurable sort options
- Results can be dragged and dropped
- Results can be bookmarked

This slide summarizes the search result features available with WebSphere Commerce search. When a search is run, additional constraints are applied based on the shopper's context. Search results are scoped by the store's catalog and, in the case of extended sites, the parent catalog. Any contract entitlement rules are also applied.

On the previous slide, you saw examples of search facets, term highlighting, the interactive bread crumb trail and unstructured content results in the details view. An alternative view, called grid view, is also available and allows more results to be displayed on a page reducing the total number of results pages.

When designing your search results page, you can decide whether to display an index product price or a calculated one. You can also configure the sort options available to the shopper.

For convenience, shoppers can drag search results and bookmark search results and return to them at a later time.

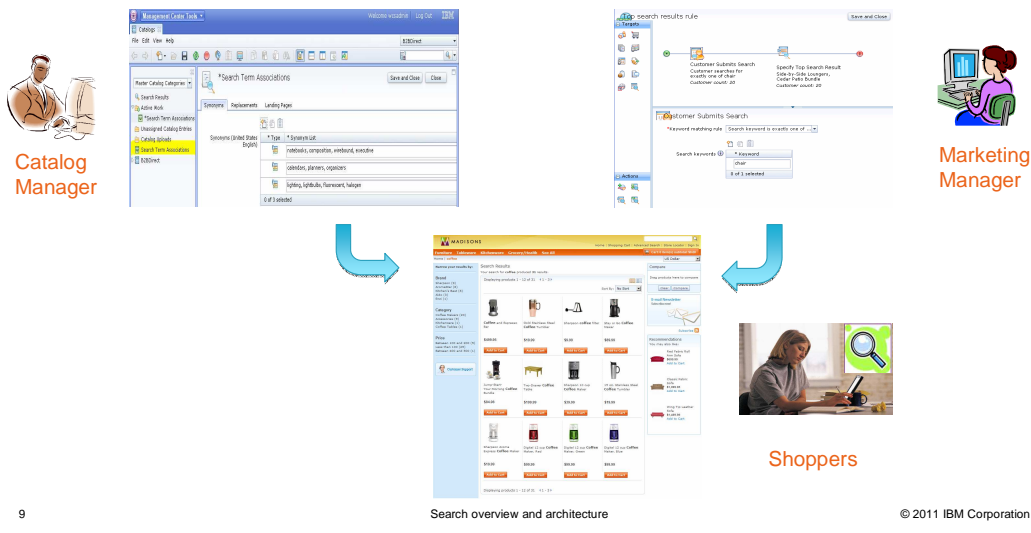
Search based navigation

- By keyword
 - Quick search and advanced search
- By category
 - Browse entitled products
- By facet
 - Select attributes to drill down

The integrated search solution offers shoppers three ways to navigate your store. The first is search by a keyword as shown on the previous slides. The second is by browsing the catalog. Search queries are used to build the catalog hierarchy so only entitled categories and products are shown during navigation. The third is by facet selection. Shoppers can drill down on selected product attributes to find exactly what they are looking for. Facet selection can be combined with either keyword searches or category browsing.

Search management tools

- Search Term Association Tool
 - Synonym terms
 - Replacement terms
 - Landing Pages
- Search Marketing Tool
 - Product Promotions
 - Cross Sell and Up Sell
 - Search Results Ranking and Sorting



In addition to the new storefront search features, the WebSphere Commerce search solution provides even more control over search results through a new set of search management tools. The search management tools are split into two groups. The search term association tool is part of the Management Center Catalogs tool. This tool allows catalog managers to define synonyms and replacement terms for common search terms. Synonyms and replacement terms help ensure shoppers find the products they are looking for even if they use different terms to describe a product than your store catalog does. Landing pages allow you to bypass the search results page and redirect shoppers to a specific store page based on their search term. Landing pages can be used to direct shoppers to store information such as address or shipping information. They can also be used to redirect to brand or category specific pages within the store.

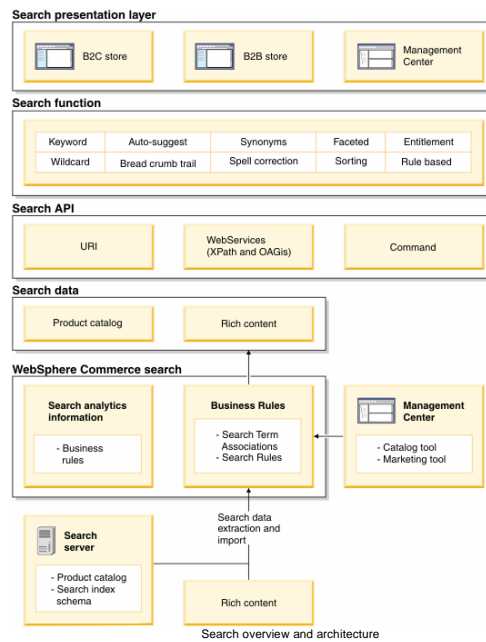
The second group of integrated search tools are part of the Management Center Marketing tool. Using these tools, Marketing Managers can define business rules that influence the content and ranking of search results within the store. For example, you can define rules that rank products from certain categories or manufacturers higher in the search results. You can also set specific products to always be the top results. Beyond impacting the search results themselves, you can also use search results and behavior to drive other marketing activities such as displaying cross sell recommendations based on current search results.

The combination of search term associations and search marketing rules determine the final search results displayed to the shopper. You can learn more about these tools in the search tools overview presentation in the business user interface section.

Architecture overview

This section covers an overview of the WebSphere Commerce search architecture.

WebSphere Commerce search solution



11

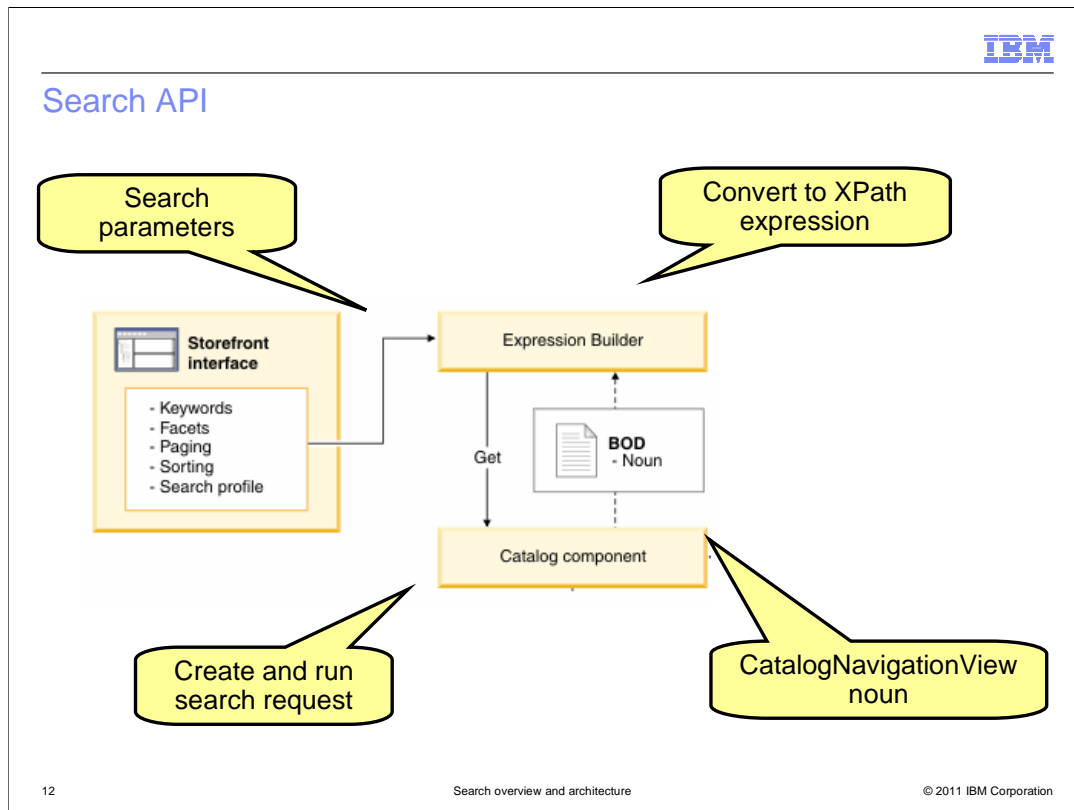
© 2011 IBM Corporation

The WebSphere Commerce search solution has many layers. The previous section of the presentation focused on the first two layers shown in this diagram, the presentation layer and the search functions. The search API provides a way for the storefront to communicate with the WebSphere Commerce search engine. The search engine processes the request and returns search data such as matching catalog entries and rich content back to the UI. This API is discussed next.

Following the API overview is the WebSphere Commerce search architecture. The WebSphere Commerce search implementation is responsible for applying business rules and processing queries against the search index. The business rules are defined using the Catalog and Marketing tools in Management Center. As mentioned in the previous section, you can learn more about these tools in the search tools overview presentation in the business user interface section.

The architecture overview concludes with an introduction to the search server technology and search index design.

Search API



The search API is responsible for capturing the details about a search request, initiating the request and displaying the search results back to the shopper. Many shopper actions within the store can trigger a search query to run. For example typing a term in the store search field, browsing a catalog and drilling down on a facet. When the shopper triggers a search, the parameters associated with their context and the search request itself are captured and passed to an expression builder. The expression builder converts the search query into an XPath expression and uses a get service call to signal the catalog component to create and run the search request. The response business object document (BOD) contains the CatalogNavigationView noun. This noun contains the results of the search query and is used by the storefront to display the search results.

Search input parameters

- Stored as control parameters in the XPath expression
- Represented internally as a SelectionCriteria object
- Available parameters
 - Access profile
 - Search profile
 - Search filters (includes search term)
 - Search type
 - Sort order
 - Metadata
 - Custom search expression

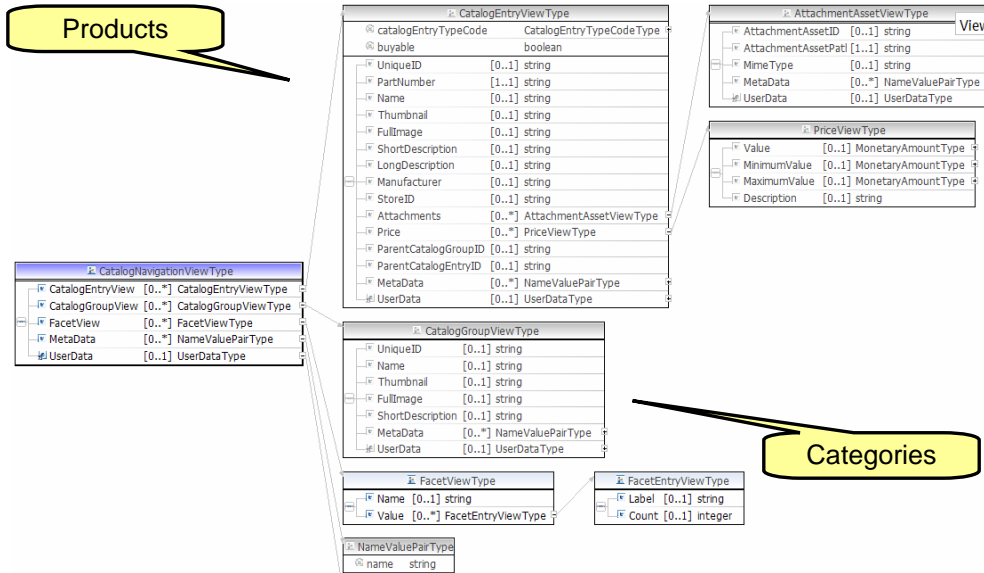
The search parameters collected from the storefront are stored as control parameters in the XPath expression. Internally, the set of parameters are represented as a SelectionCriteria object.

This slide lists a summary of the available parameters. The name and usage of each parameter is described in the WebSphere Commerce Information Center. The access profile is commonly represented as a control parameter in an XPath expression. The search-specific parameters include the search profile, filters, type, sort order, metadata and optionally a custom search expression. The search profile contains common details for each query initiated from a specific page in the store. The search filters contain the details about the current search such as the search term, included or excluded categories, and selected facets. Search types can be exact, none, all, or any. Sort order defines how the search results should be sorted. Metadata contains information about the shoppers current context such as facets already selected. If needed, a custom search expression can also be passed as a parameter to the search.

Sample XPath expression

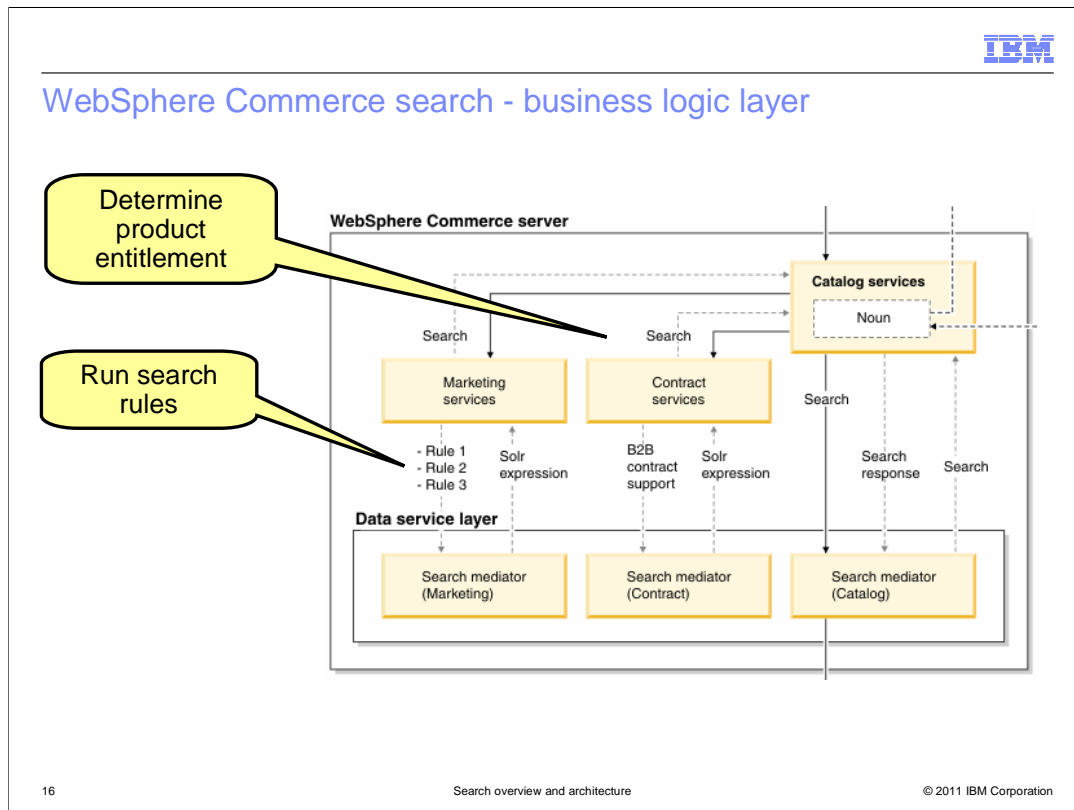
This slide shows an example of a storefront search query using the advanced search page. The shopper is searching for the term coffee and excluding the term table. They are looking for results between \$100 and \$200 located in the coffee makers category. The XPath expression with all of the corresponding control parameters is shown on the bottom of the slide.

CatalogNavigationView noun



The results of a search are returned to the storefront as a **CatalogNavigationView** noun.

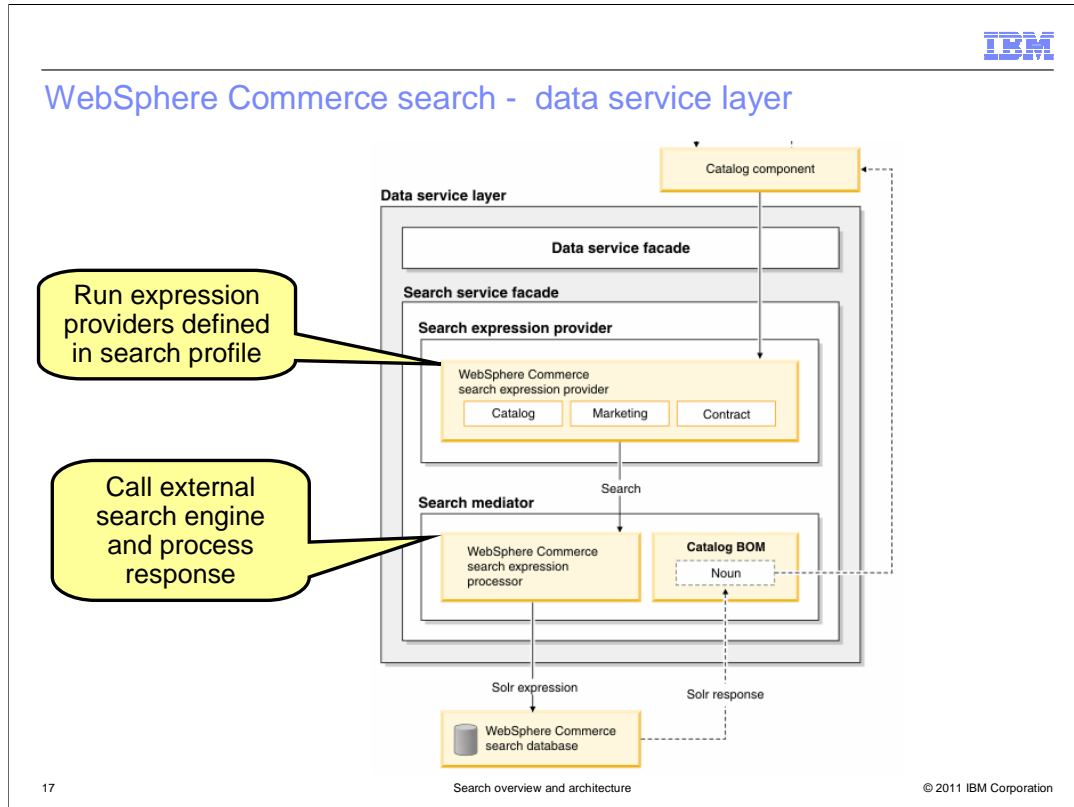
WebSphere Commerce search - business logic layer



The next two slides look at the WebSphere Commerce search architecture in more detail. First the business logic layer and then the data service layer.

When an XPath search request is received from the storefront, the business logic layer examines the control parameters and applies any necessary logic. The catalog services control the processing of the search request. The XPath control parameters are converted into Solr expression fragments. Context information such as store path and catalog ID are also added to the Solr query. The marketing services check whether any configured search rules match the current request and if so, the actions defined in the rule are converted to Solr expressions and added to the query. To ensure the search results only contain products the shopper is entitled to buy, the contract services check the catalog filter for the current contract. The catalog filter constraints are returned as a Solr expression and added to the query. When all of the business rules have been processed, the catalog services trigger the Solr query to be run on the search server.

WebSphere Commerce search - data service layer



At the data service layer, a new façade has been added for handling search requests. The search service façade is responsible for all of the search engine specific logic. The search profile, which was passed in from the storefront as a control parameter, defines a set of expression providers that are called in sequence to build the search engine specific query. Each expression provider takes the XPath form of the query as input and returns a Solr expression fragment as output. To integrate with a search engine other than Solr, you need to define expression providers that convert the XPath to the query language of your chosen search engine. Once the Solr query is fully formed and verified, the search mediator passes the query to the external search engine to be run. It then builds a CatalogNavigationView noun from the Solr response object. If you are integrating with a search engine other than Solr, you will need to provide a search mediator to invoke your external search engine and process its response.

Underlying technology

- Apache Solr v1.4
 - Enterprise search server based on Lucene
 - Powerful full-text search
 - Highly scalable
- Apache Tika v0.4
 - Toolkit for detecting and extracting metadata and structured text content from various documents
 - Used to index product attachments

The WebSphere Commerce search solution includes two open source projects. Apache Solr is an enterprise search server based on the Lucene search libraries. It provides a powerful full-text search and is scalable to match your search needs. The WebSphere Commerce search solution uses version 1.4 of Apache Solr. The second open source project used is Apache Tika. Tika provides indexing capability for rich text documents and is used to build the unstructured content index for product attachments. The WebSphere Commerce search solution uses version 0.4 of Apache Tika.

Solr terminology

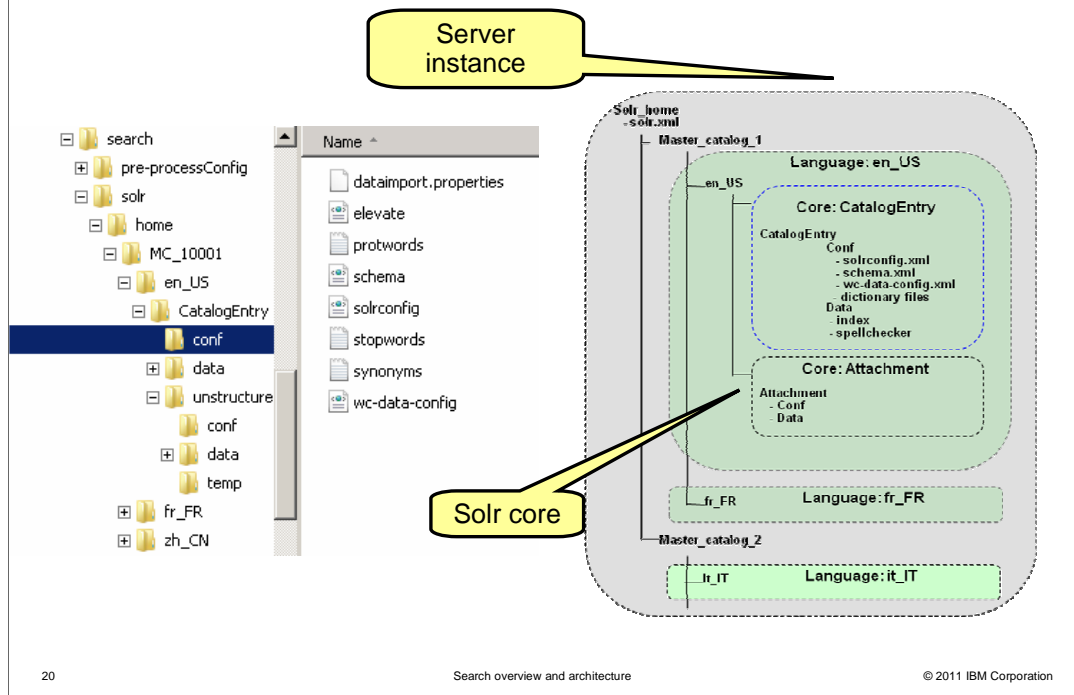
- Server instance
 - Self-contained run time environment
- Core
 - Allows one Solr instance to have multiple configurations and indexes
- Index
 - Searchable collections of items called documents which represent catalog entries
- Schema
 - Defines the index fields and data type of the fields
- Data import handler (DIH)
 - Imports data from relational database to Solr index

This slide summarizes some Solr terms that are used in the rest of this presentation. A Solr server instance is the complete run time environment. In most production systems this is equivalent to one WebSphere Application Server node. You can learn more about deployment options in the search configuration and administration overview presentation in the store administration section.

A Solr core contains a search index and related configuration information. One Solr server instance can contain multiple cores and therefore multiple indexes. The index is a searchable collection of items called documents. Each document in the index represents a catalog entry. Documents are composed of a set of attributes called fields. For example, a catalog entry document has fields for part number, name and description. The name and data type of each field is defined in the index schema.

The Solr data import handler provides a configuration driven way to import data from relational database or XML into a Solr index. The WebSphere Commerce search solution provides a command line utility to invoke the data import handler to load data from your WebSphere Commerce database into the Solr index. This utility is covered in the search configuration and administration overview presentation.

Solr core structure



20

Search overview and architecture

© 2011 IBM Corporation

Since search is language sensitive, you need the documents in the index to contain only words in the same language as the search term. To support this separation, each language supported by your store requires two cores in the Solr server instance. One for catalog documents and one for unstructured content documents specific to that language. In the diagram on the right, these are represented by the CatalogEntry core and the Attachment core for the English language. If your store also supports French, another pair of cores are needed. Each core is also specific to a master catalog. If you have a second master catalog, as shown in the diagram, new cores are required for each language it supports.

On the left of the slide is the directory structure for the structured and unstructured cores for a single master catalog and language. Each language directory under the master catalog has its own pair of cores.

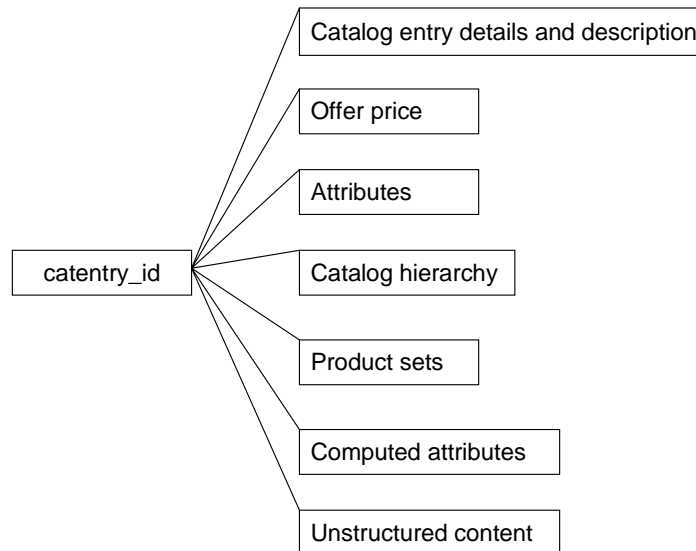
Main index definition files

- solr.xml
 - List of configured cores
- schema.xml
 - Field type definitions
 - Fields to index
- solrconfig.xml
 - Config parameters for the index
- wc-data-config.xml
 - Queries used by the Data Import Handler

A list of all configured cores can be found in the solr.xml file in the Solr home directory. This directory structure was shown on the previous slide. The index for each core is defined by a file called schema.xml. This file contains two main sections. The first contains all the type definitions for the data stored in the index. The second section defines the field names and corresponding types that make up the index. Additional data in this file include the unique key for the index and the default query operator. For the WebSphere Commerce search solution, the unique key for each document in the index is the catalog entry ID. The default query operator is OR. The solrconfig.xml file contains Solr configurations options. The wc-data-config.xml file defines the queries that are used by the Data Import Handler to build the search index.

The two main files you will need to work with are schema.xml and wc-data-config.xml. These two files are responsible for defining and loading the index used by the starter stores. You can use this index as a starting point and modify it based on your business needs. Some index customization scenarios are discussed in the customization section of this presentation.

CatalogEntry index design



22

Search overview and architecture

© 2011 IBM Corporation

This slide summarizes the data contained in the CatalogEntry index used by the Madisons and Elite starter stores. Each document in the index represents one catalog entry in the WebSphere Commerce database. The description information stored corresponds to the language for the index.

Additional information related to each catalog entry is also stored in the index. The offer price and product attributes can be used to create facets that are displayed in the storefront to help shoppers narrow down the initial search results. The full hierarchy of the master catalog and all sales catalogs is included so that search results can be scoped to the catalog the shopper is currently viewing. This data also supports search requests in extended sites stores. Product sets are included to support contract entitlement in search requests. Only products the shopper is currently entitled to see based on their current store and contract are returned in the search results. Examples of some computed attributes are provided in the starter store index schema but are not initially loaded.

Computed attributes are used to cache a version of real time data so that it can be used in determining the search results. Examples of computed attributes include customer product ratings, best selling products and profit margins. The unstructured content field is used to look up catalog entries that correspond to records in the unstructured content index.

You can learn how to extend the starter store index by following the tutorials described in the next section. You can learn more about populating the index in the search configuration and administration overview presentation in the store administration section.

Customization

This section introduces the WebSphere Commerce search customization scenarios.

Main customization scenarios

- Index extension
 - Index additional columns from WebSphere Commerce base tables
 - Index custom extension tables
 - Index external data sources
- Search management UI extension
 - Additional filters
 - New business rules
 - Override Solr synonym expansion
- Store front extension
 - Facet attributes
 - Unstructured content

Search customization scenarios fall into three main groups. The first is index extension. This includes adding new information into the index such as base table fields that are not indexed by default, custom extension tables and data from external sources.

The second customization group is extension to the Management Center search management tools. This can include Management Center UI updates and new expression builder classes. Some examples include adding search filters to existing rules, adding new business rules and overriding Solr synonym expansion with a custom synonym expansion method.

The final customization group is storefront extension. This can include customizing the display of facet information or displaying new unstructured content.

Tutorial: Adding catalog entry warranty information to the search

- Extend the search schema
 - schema.xml
- Extend the Data Import Handler mapping
 - wc-data-config.xml
- Rebuild the search index
 - di-buildindex.bat -masterCatalogId masterCatalogId
- Add a new search profile to search against the custom fields
 - Update wc-search.xml to define the new query
 - Update /Snippets/Search/SearchSetup.jspf to use the new profile in the store
- Extend the business object mediator for CatalogNavigationView to map the new fields into user data
 - com.ibm.commerce.catalog-fep\wc-business-object-mediator.xml

The tutorials section of the WebSphere Commerce Information Center contains two new tutorials on the WebSphere Commerce search solution. The first of these tutorials demonstrates how to add data stored in custom tables to your search index. This slide summarizes the high level steps covered by the tutorial.

The first step is to extend the index schema to include the new fields. Since the fields have a one-to-one mapping with a catalog entry, no changes are needed to the preprocessing scripts, but the wc-data-config file is updated to extend the queries used by the data import handler. Once the index extensions are ready, the search index is rebuilt to include the new data.

The tutorial continues with an extension to the search profile to add a custom field to the list of index fields queried when a search is run. The storefront is updated to use the new search profile.

The final step is to extend the business object mediator for the CatalogNavigationView noun to map the new fields returned by the Solr query into the user data fields of the noun.

Example: Bypass Solr synonym expansion

- Create a new expression provider
 - Use SearchFacadeClient to look up synonyms
 - Append synonyms to search terms
 - Sample code available in Information Center
- Create a new search profile to use the expression provider

This example of a search management customization is documented in the WebSphere Commerce Information Center but not as a tutorial.

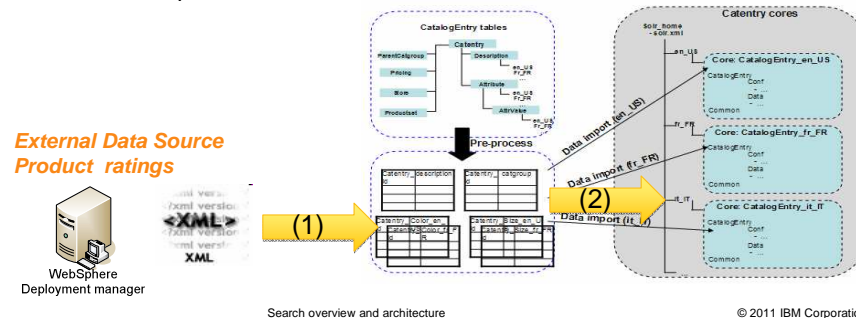
If you want to bypass Solr's synonym expansion and provide your own this is possible by creating a new expression provider. The SearchFacadeClient can be used to look up synonyms defined in the search term association tools and your expression provider determines how those synonyms should be expanded in the Solr search query. For example, you can support multi-word synonyms or expand synonyms to include plurals of each term. The Information Center provides sample code for a custom expression builder.

Once you have completed your expression builder, you need to create a new search profile that includes it.

Tutorial: Indexing ratings data

- Load product rating information into database table
 - Modify pre-process configuration file
 - Develop code to load data from XML file to the flat table

- Index the data from the database table
 - Run 'di-preprocess' script
 - Update wc-data-config.xml file
 - Run 'di-build index' script



The second tutorial provided in the WebSphere Commerce Information Center touches on all three types of search customization. It demonstrates how to index product ratings data and use the data in search business rules and storefront search results.

The tutorial starts by extending the preprocessor to create and load a flat table for ratings data. Some sample code is also provided to load the XML formatted rating source file into the flat table. The sample schema provides a field for ratings data so no index schema extension is needed but the wc-data-config file must be updated to load the data from the new flat table into the index. When the index customization steps are complete, the preprocess and index building script are run and the ratings data is loaded into the search index.

Create search sorting rule for customer rating

- Update 'Order Search Result' definition file

```
<PropertyValue displayName="{extCatalogResources.displayProduct_ranking}" value="customerRanking"/>
```

- Create a new Search Rule in Management Center

Order Search Result

* Action Change how initial search results are ranked
 Change how initial search results are sorted

Sort criteria Override previously defined search ordering

* Sort by	* Sequence
Ranking	Descending

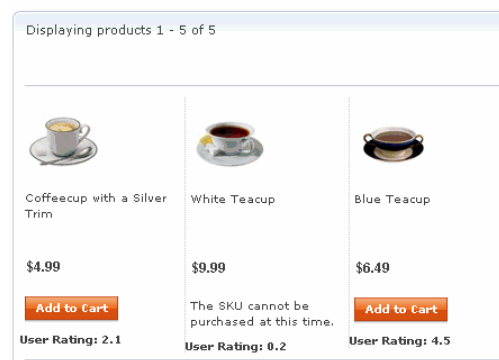
0 of 1 selected

In the next part of the tutorial, Management Center is updated to allow product ranking to be used as a sort option in the Change Search Result Order search rule action. A new option is added to the sort dropdown that maps to the name of the index field.

Sorting product ratings in search result

- Show the product ratings in storefront
 - Update BOM file
 - Add JSP code snippets

Product rating



The final step is to update the Madisons store to display the ranking information in the search results. As in the other tutorial, the business object mediator is updated to return the results in the CatalogNavigationView noun.

Trace summary

- New component for search term association services
 - com.ibm.commerce.search.*
- Search runtime trace
 - com.ibm.commerce.catalog.*
 - com.ibm.commerce.marketing.*
 - com.ibm.commerce.foundation.*

The WebSphere Commerce search solution adds one new trace component for the services dealing with search term association management. The majority of the search runtime trace statements can be viewed by enabling the existing catalog, marketing and foundation trace components.

References

- Search framework
 - <http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/concepts/csdsearch.htm>
- Adding catalog entry warranty information in WebSphere Commerce search
 - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/tutorial/tsd_search1_intro.htm
- Indexing ratings data in WebSphere Commerce search from an external source
 - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/tutorial/tsd_search2_intro.htm
- Using the GetSearchTermAssociation service to bypass Solr synonym expansion in WebSphere Commerce search
 - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.management-center_customization.doc/tasks/tpnsearchstabyp.htm

This slide contains some useful references.

Summary

- Solution overview
- Architecture overview
- Customization

The presentation began with an overview of the WebSphere Commerce search solution capabilities. Following the overview was a discussion of the solution architecture. This presentation concluded with some example customization scenarios.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback about SearchOverviewAndArchitecture.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20SearchOverviewAndArchitecture.ppt)

This module is also available in PDF format at: [../SearchOverviewAndArchitecture.pdf](..../SearchOverviewAndArchitecture.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

JSP, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.