

## WebSphere Commerce V7 Feature Pack 3

### WebSphere Commerce search architecture updates



© 2011 IBM Corporation

This presentation provides an overview of the architecture updates for the WebSphere Commerce search solution in Version 7 Feature Pack 3. You should be familiar with the existing WebSphere Commerce search solution before viewing this presentation.

---

## Table of contents

- Solution overview
- Architecture updates
- Customization

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

## ***Solution overview***

This section introduces the WebSphere Commerce search solution enhancements.

## WebSphere Commerce Search

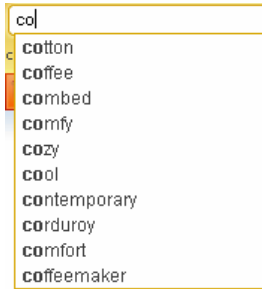
- For shoppers
  - New auto suggest options
    - Category, Brand, Articles
  - Site content search results
- For business users
  - Search Term Association tool
    - New statistics for top hits and top misses
  - Management Center access to Coremetrics reports
  - Marketing tool
    - A / B testing for search rules

WebSphere Commerce search was introduced in Feature Pack 2 and some new capability has been added in Feature Pack 3 for both shoppers and business users. The first improvement for shoppers is a redesigned auto suggest drop down that can group suggestions by type such as categories, brands and articles. The other improvement is the introduction of site content search results to complement the product results available in Feature Pack 2.

For business users, the Search Term Association tool includes new statistics for viewing top search hits and top search misses. This information can help you define more effective synonyms, replacements terms and landing pages. Coremetrics customers can access more detailed search statistics by launching a Coremetrics report directly from the Search Term Association tool. In the Marketing tool, the search rule builder has been enhanced to include search rule A / B testing. With this new capability, business users can evaluate different search rules against each other and choose the one that provides the best results for the store.

## Enhanced auto suggest

### Feature Pack 2

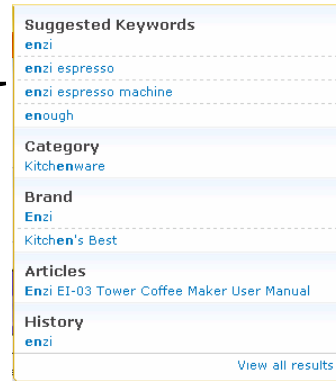


### Feature Pack 3

Multi-word keywords

Suggestion groupings

Search term history







The auto suggest drop down in Feature Pack 3 has several enhancements compared to Feature Pack 2. For standard search term suggestions, both single and multi-word keywords are now supported. In addition to suggesting specific search terms, the auto suggest also provides three different groups of suggestions. The default groups are category, brand and articles. Additional groups can be added as a customization. At the bottom of the auto suggest is a search term history for the shopper. This contains matching search terms used in the current shopping session.

## Separate search results area for site content

Search Results  
 Search for: **"coffee"** . Number of results found: **34**.  
 Products: 31, Articles & Videos: 3



**Products: 31**

Displaying products 1 - 12 of 31    < 1 - 3 >    Sort By: No Sort

|   |   |   |  |
|---|---|---|--|
| <br><b>Coffee and Espresso Bar</b> | <br><b>Stay or Go Coffee Maker</b> | <br><b>Gold Stainless Steel Coffee Tumbler</b> | <br><b>Sharpson coffee filter</b> |
|---|---|---|--|

**Articles & Videos: 3**

Displaying articles & videos 1 - 3 of 3    Sort By: No Sort

|   |
|---|
| <br><b>Coffee Recipe</b><br>HTML             |
| <br><b>How to Brew Coffee Video</b><br>VIDEO |

Another new storefront feature is a separate results area for site content such as articles and videos. The search results summary at the top indicates how many of each result types were found and the site content results are displayed in a separate results section below the catalog results. Note that site content does not have entitlement so all shoppers will see the same results.

## Top search hits

Search Term Associations

Synonyms
Replacements
Landing Pages
Top Search Hits
Top Search Misses

Reporting period: Choose reporting period From 2011/07/08 to 2011/07/08

Keyword:

Show top keywords: All

Language: United States English

Total searching sessions: 21

| Statistics ⓘ | Keyword  | Searching Sessions | Results for Keyword | Synonyms or Replacements  |
|--------------|----------|--------------------|---------------------|---------------------------|
|              | light    | 2 (9.52 %)         | 10                  | light, lamp               |
|              | fruit    | 2 (9.52 %)         | 4                   | banana, apple, strawberry |
|              | coffee   | 2 (9.52 %)         | 31                  |                           |
|              | sharpson | 1 (4.76 %)         | 10                  |                           |
|              | glass    | 1 (4.76 %)         | 7                   |                           |
|              | lamp     | 1 (4.76 %)         | 10                  | light, lamp               |

7
WebSphere Commerce search architecture updates
© 2011 IBM Corporation

The Search Term Association tool includes two new tabs for viewing search statistics, top search hits and top search misses. This slide shows a top search hits tab. The top half of the screen shows the selected report parameters. The bottom half of the screen shows the report details. Statistics gathering is not enabled by default but can be enabled through a simple configuration update.

## A / B testing for search rules

Coffee experiment (Read-Only) Save and Cl...

**Experiment**

General Properties Paths **Statistics**

Total number of unique customers 4

| * Name           | * Winner                 | Unique Custom | Clicks | Click Orders | Click Revenue | Currency |
|------------------|--------------------------|---------------|--------|--------------|---------------|----------|
| Descending price | <input type="checkbox"/> | 2             | 1      | 0            | 0.00          |          |
| Ascending price  | <input type="checkbox"/> | 2             | 2      | 2            | 26.98         | USD      |

0 of 2 selected

8 WebSphere Commerce search architecture updates © 2011 IBM Corporation

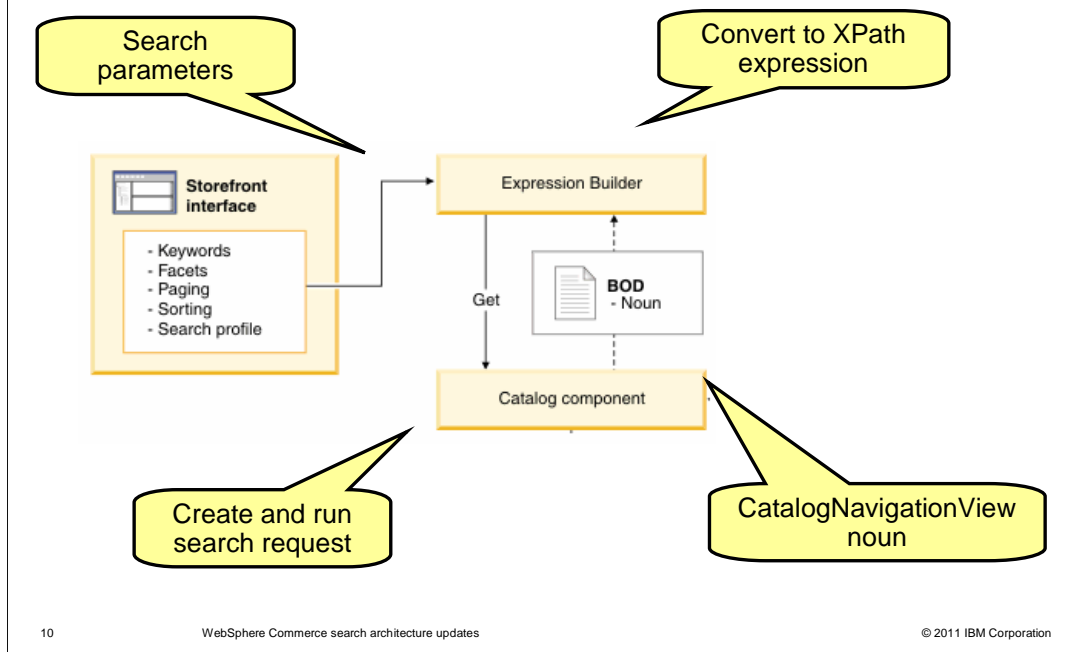
Search rule A / B testing is supported by adding the Experiment element to the search rule builder palette. Here you see an example of a search rule with the Experiment branch added. The experiment branch allows you to add one or more comparison paths to the search rule and collect statistics on the effectiveness of each path. Sample statistics are shown in the lower half of the screen capture. When the experiment is complete, you can select a winning path and that path becomes the search rule used in your store going forward.



## ***Architecture updates***

This section covers the updates to the WebSphere Commerce search solution architecture.

## Search API



10

WebSphere Commerce search architecture updates

© 2011 IBM Corporation

This slide reviews the WebSphere Commerce search API. The 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.

## Expression builders

- Storefront search and navigation
  - **getCatalogNavigationView\***
  - **getCatalogNavigationAttachmentView\***
  - getWebContentView
- SEO
  - getCatalogNavigationCatalogGroupView
- Configurator
  - getCatalogEntrySearchResultsByIDView
  - getCatalogNavigationCatalogEntryView
- Catalog entry
  - Covered in catalog programming model presentation

\* Feature Pack 2 expression builders

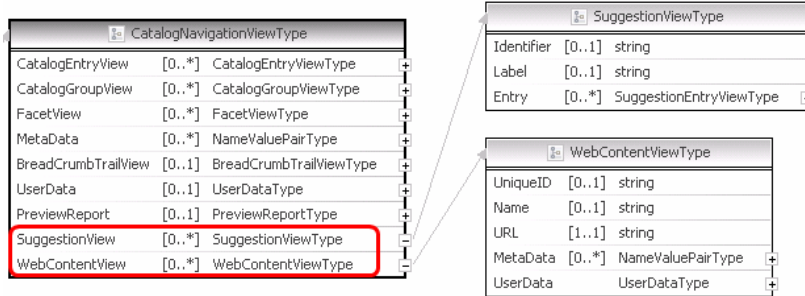
This slide summarizes the CatalogNavigationView expression builders available in Feature Pack 3. The two highlighted in green were part of Feature Pack 2 and the rest are new. The two new storefront capabilities shown previously make use of the storefront search and navigation expression builders.

The auto suggest feature uses the existing getCatalogNavigationView expression builder and a new search profile to initiate the request for navigation suggestions. The navigation suggestions are stored in a new part of the CatalogNavigationView noun shown on the next slide. Once the information is returned to the storefront, it is cached for future use.

For storefront search and navigation there is one new expression provider, getWebContentView. This expression builder and its associated search profile provide the information needed to build a Solr query for site content that is run against the unstructured content index where the site content is stored. Like navigation suggestions, site content results are stored in a new part of the CatalogNavigationView noun shown on the next slide.

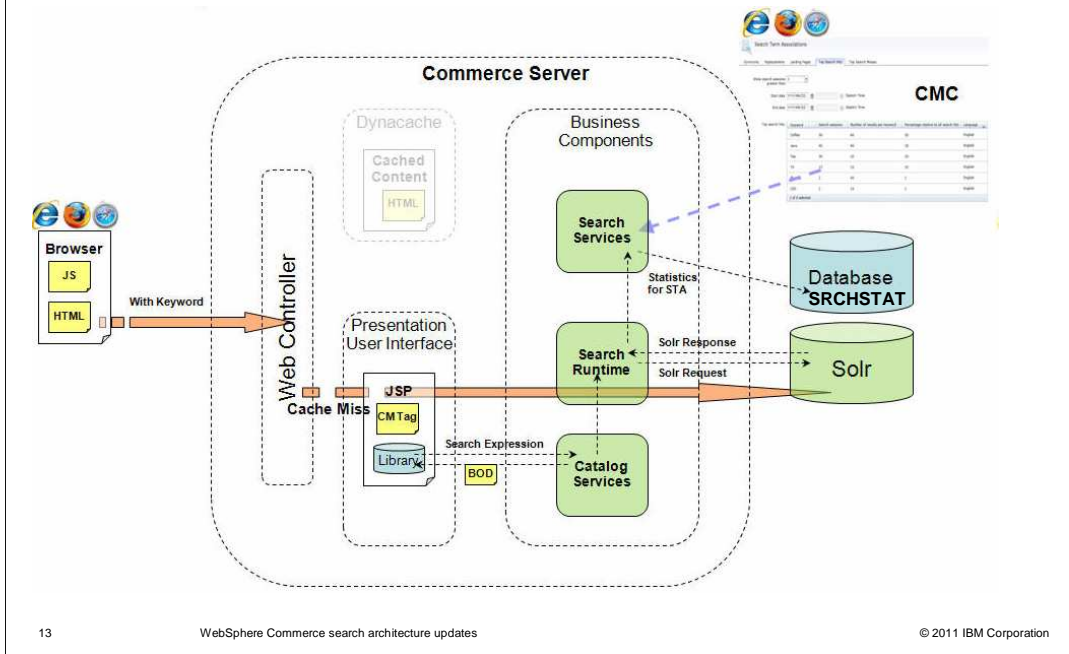
The remaining expression builders are included for reference but are not discussed in detail in this presentation.

## CatalogNavigationView noun changes for search



This slide shows the changes to the CatalogNavigationView noun to support navigation suggestions and site content search results. Each is represented as a new noun part circled in red on the slide. Existing noun parts such as CatalogEntryView and CatalogGroup view have been extended in Feature Pack 3 to support other features such as the new storefront catalog services. These changes are discussed in more detail in the Catalog Programming Model presentation.

## Interaction diagram for search term association statistics



13

WebSphere Commerce search architecture updates

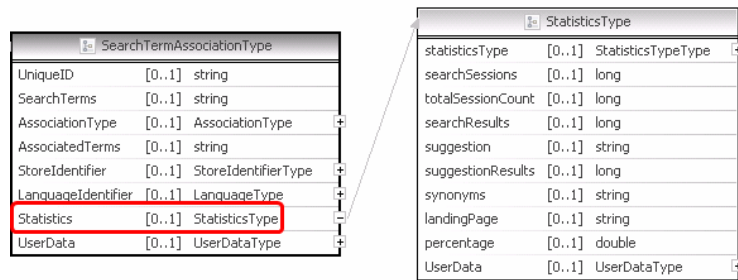
© 2011 IBM Corporation

The presentation now moves to the architecture changes for statistics support. In this diagram you see the flow for a typical shopper search query with STA statistics enabled. The search expression is built up using expression providers and issued against the Solr server. The change for statistics comes when the Solr response is received and processed by the search service façade. At this point, if statistics are enabled, the Search services are invoked to store the details on search hits or search misses in the WebSphere Commerce database. A new table called SRCHSTAT has been created to store the statistics.

Since updating the database on each search can create considerable performance overhead, the search services can buffer statistics in memory until a preconfigured limit is reached. You can set a limit on the number of cached entries and on the maximum time to cache before processing cached entries. When the cached entries are stored in the database, one row is created for each keyword in the cache. This means that after the buffer is processed several times, the statistics for a single keyword might be spread across several rows in the table. When a business user requests a top search hits or top search misses report in Management Center, the results for each keyword are consolidated. The SearchTermAssociation noun has been extended to store the statistics data and is shown on the next slide.

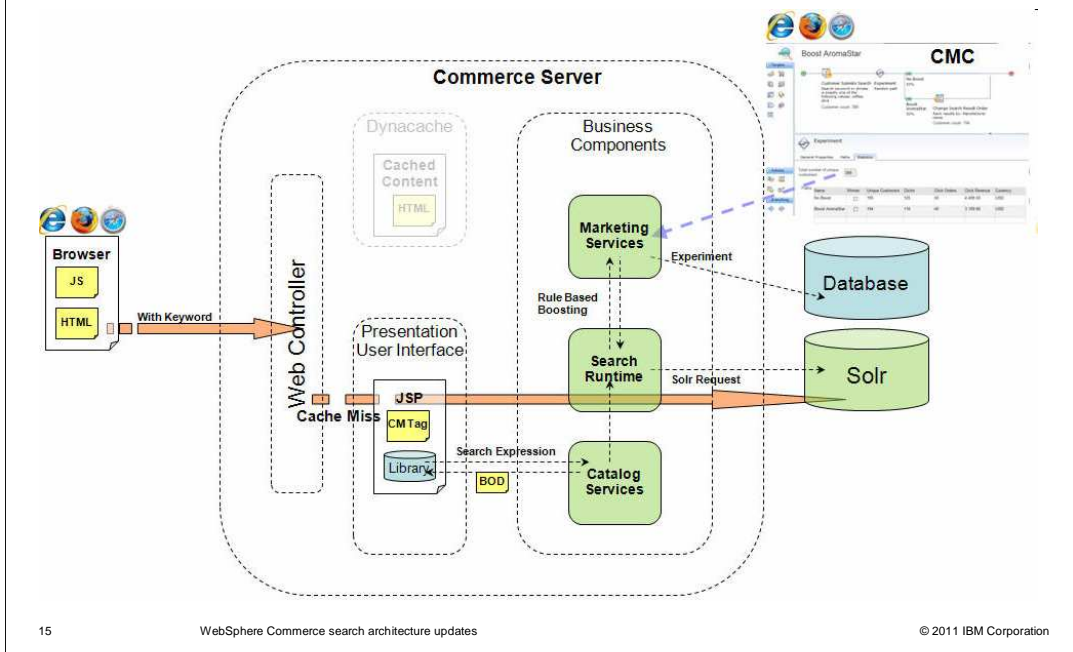
More information on configuring statistics is included in the Search Configuration and Administration Updates presentation.

## Noun changes for SearchTermAssociation



The Statistics noun part of the SearchTermAssociation noun has been added to pass statistics information to the Management Center Search Term Association tool.

## Interaction diagram for search rule A/B testing



This flow diagram shows the changes to a typical search query when a search rule experiment is enabled. As with STA statistics, there is no change in the creation and issuing of the Solr query. When the query results are returned, the experiment identifier and a test element identifier are added to the results in the CatalogNavigationView noun. From there, the existing marketing experiments implementation takes over to track search result clicks and resulting orders in the store. Business users can view the statistics for an ongoing experiment in Management Center as shown earlier in this presentation.

## Underlying technology

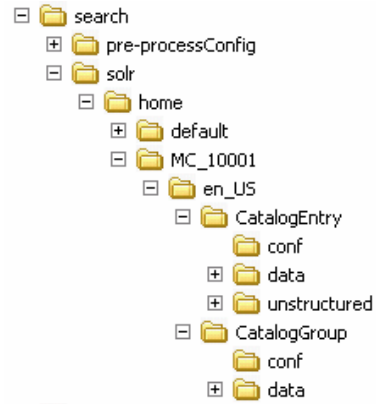
- Apache Solr v1.4
  - Enterprise search server based on Lucene
  - Powerful full-text search
- Apache Tika v0.8
  - Toolkit for detecting and extracting metadata and structured text content from various documents
  - Supported document types
    - Feature Pack 2: PDF, Word, RTF
    - Feature Pack 3: HTML, metadata for binary content types (mp3, video)
- Apache Droids
  - Web crawler to crawl your store and save pages as HTML

This slide shows the open source projects included in the WebSphere Commerce search solution. The supported version of Solr has not changed from Feature Pack 2 to Feature Pack 3. The Tika libraries have been upgraded from 0.4 to 0.8 to include support for parsing HTML and extracting metadata from binary content types such as mp3 and video. The new package in Feature Pack 3 is Apache Droids. The libraries from this project add support for crawling your site to index site content. The capability is discussed in more detail in the coming slides.



## CatalogGroup index

- Paired with CatalogEntry index
  - One index per master catalog and language
- Provides category results for auto-suggest
- Preprocessing table
  - TI\_CGCATALOG\_1
    - Contains category to catalog mapping
- Created and populated using existing command line utilities
- No delta update



The next three slides review the Solr indexes created by default when you enable WebSphere Commerce search in Feature Pack 3. The CatalogGroup index is new for Feature Pack 3. It contains data that is used to build the category section of the auto suggest drop down. A CatalogGroup index is created for each CatalogEntry index in your site. That is, one index is created for each combination of master catalog and language. In order to build the CatalogGroup index, one new pre-processing table is added. The creation and population of the CatalogGroup index is provided by the same command line utilities as the CatalogEntry index. Updates to these utilities are covered in the configuration section. The CatalogGroup index does not have delta update support.

## CatalogEntry index additions

- CATENTDESC
  - longdescription
- CATGPENREL
  - sequence
- SEOURL
  - seo\_token\_ntk
- CATENTSUBS
  - disallowRecOrder
  - subscripType
- New preprocessing tables
  - TI\_CATGPENREL\_1
  - TI\_SEOURL\_1\_1

The CatalogEntry index contains a few new fields in Feature Pack 3. These fields have been added to support other Feature Pack 3 solutions such as search engine optimization and the new storefront catalog services. To load the new data, two new pre-processing tables have been added.

## Unstructured content index changes

- attachmentrel\_id changes from long to string
- Attachment content
  - attachmentrel\_id is atchast\_id (attachment asset)
  - attachment\_id is atchrel\_id (attachment relationship)
  - catentry\_id is related catalog entry

| Field            | ITSVopf0LB | Norm | Value                   |
|------------------|------------|------|-------------------------|
| attachment_id    | --S----0-B | ---  | 00 00 00 00 00 00 04 28 |
| attachmentrel_id | I-S---f0-- | ---  | 1054                    |
| catentry_id      | --S----0-B | ---  | 00 00 00 00 00 00 27 4b |

- HTML content
  - attachmentrel\_id has "HTML\_" prefix followed by content URL
  - attachment\_id and catentry\_id are -1

| Field        | ITSVopf0LB | Norm | Value                         |
|--------------|------------|------|-------------------------------|
| attachment_i | --S----0-B | ---  | ff ff ff ff ff ff ff          |
| attachmentre | I-S---f0-- | ---  | HTML_StaticContentRecipe.html |
| catentry_id  | --S----0-B | ---  | ff ff ff ff ff ff ff          |

The changes to the unstructured content index have been made to support the addition of store site content to this index. The attachmentrel\_id field type has changed from long to string to support the ID format for site content. For product attachments, this value is the primary key for the attachment asset. For site content, it is the prefix HTML followed by the site content URL. Also, the attachment\_id and catentry\_id fields used for product attachments do not have meaning for site content and are set to -1.

## Indexing site content

- Two part process decouples crawling site content from indexing site content
- Part 1: Prepare manifest file
  - Manual process
    - Files must already exist as HTML
  - Automate using Droids web crawler
    - Convert JSP files to HTML
- Part 2: Index content listed in manifest file
  - New request handler for DIH
    - Tika indexes HTML files

The building of the site content documents in the unstructured content index is a two part process that involves different open source tools. An intermediate document, known as a manifest file, is a list of all the site content to be indexed for the site. The first step in the indexing process is to build this file. If you have only a few files and they already exist as HTML documents, you might choose to build the manifest file manually. If you have a large amount of site content spread throughout your site or the content is not stored as HTML, you can use the Apache Droids web crawler to automate the creation of the manifest file. In addition to building the manifest file, the web crawler can create local copies of the site content as HTML to prepare it for indexing. This process is examined in more detail on a coming slide.

The second part of the indexing process is to parse the HTML and populate the index. A new request handler has been created for the data import handler that uses the manifest file as the source of documents to index. The Apache Tika libraries are used to parse the HTML file.

## Manifest file

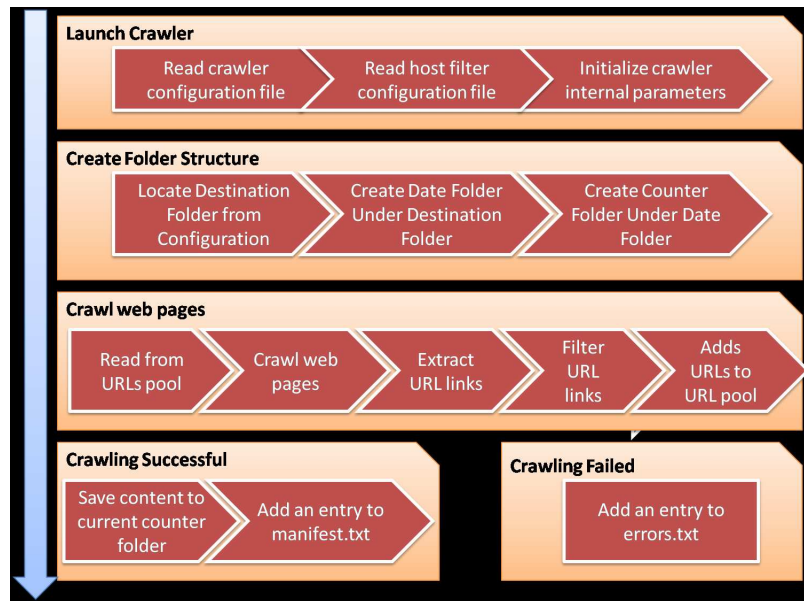
- CSV file
- Populated manually or by web crawler
- Columns
  - Index\*
  - URL\*
  - Saved file name\*
  - Mime type
  - Encoding
  - Delta
  - Version number

| Index | URL   | Saved file name                           | Mime type | Encoding   | Delta | Version Number |
|-------|---|---|-----------|------------|-------|----------------|
| 0     | http://9.115.33.158/webapp/wcs/stores/servlet/Madisons/index.jsp        | 7295f4ba-c8b2-4b84-88c7-d89351e3ded4.html | text/html | ISO-8859-1 | A     | 2              |
| 1     | http://9.115.33.158/webapp/wcs/stores/servlet/TopCategories_10001_10001 | 77493904-ec33-4b1a-a83f-174704541259.html | text/html | UTF-8      | S     | 2              |

\*Required values

This slide summarizes the structure of the manifest file. The manifest is a comma-separated value (CSV) file that can be created manually or automatically by the web crawler. Column headings are not required. The index, URL and saved file name values are required for both manually created and automatically generated files. The remaining columns are included in the automatically generated file but are optional in a manually created file. The last two columns, delta and version number are included to support delta updates for site content.

## Web crawler flow diagram



22

WebSphere Commerce search architecture updates

© 2011 IBM Corporation

This slide describes the flow of the web crawler. When the crawler is launched, it first reads its associated configuration files and initializes internal parameters. The configuration files contain information like the starting URL, destination folder to save HTML to, and regular expressions that define URLs and host names to include or exclude from crawling. These are described further in the configuration section. The next step in the process is to create the folder structure for the output of the parser. The content in the destination folder is what the Tika parser is run against. The files in the counter folder keep track of the crawler's process. The crawling process is shown in line three of the diagram. A URL is read from the pool to be processed and that page is crawled and saved locally as HTML. Any URL links on the page are extracted and compared to the regular expression filter defined in the configuration file. If the URL matches a format that should be crawled, it is added to the URL pool. If the crawling of a page is successful the counter data is updated and an entry for the HTML file is added to the manifest. If crawling fails, the error is logged in the errors.txt file.

# ***Customization***

This section introduces some new customization scenarios.

## Add a new auto suggest grouping

- Create a new suggestion handler
  - Implement NavigationSuggestion
  - Extend AbstractNavigationSuggestion

| Each handler will implement this interface:  | Each handler will extend this class:   |
|--|--|
| <pre>public interface NavigationSuggestion {     public SuggestionViewType populateSuggestion(String profile,         CatalogNavigationViewType catalogNavigationView);     public Map getProperties();     public void setProperties(Map properties); }</pre> | <pre>public abstract class AbstractNavigationSuggestion implements     NavigationSuggestion {     private Map iProperties;     public Map getProperties() {         return iProperties;     }     public void setProperties(Map properties) {         iProperties = properties;     }     public abstract SuggestionViewType populateSuggestion(         String profile, CatalogNavigationViewType catalogNavigationView); }</pre> |

- Add suggestion handler to navigation search profile
- Update the auto suggest storefront code

The enhanced auto suggest drop down in Feature Pack 3 is designed to be extended with additional types of suggestions if needed. To create a new group, you need to implement a new navigation handler for the search result filter. Your navigation handler is responsible for populating the NavigationView noun part with your custom suggestions. Each new navigation handler must implement the NavigationSuggestion interface and extend the AbstractNavigationSuggestion class shown on the slide. Next, you register the new navigation handler in the navigation search profile as shown on the next slide. You might also need to make some storefront updates to specify the link each suggestion will redirect the shopper to.



## Navigation suggestion search profile

```
<_config:profile indexName="CatalogEntry" name="IBM_findNavigationSuggestions">
  <_config:query>
    <_config:provider classname="com.ibm.commerce.catalog.facade.server.services.search.expression.solr.SolrSearchProfileNameValidator"/>
    <_config:provider classname="com.ibm.commerce.catalog.facade.server.services.search.expression.solr.SolrSearchIndexNameValidator"/>
    <_config:provider
      classname="com.ibm.commerce.catalog.facade.server.services.search.expression.solr.SolrSearchNoSearchResultsExpressionProvider"/>
    <_config:provider classname="com.ibm.commerce.catalog.facade.server.services.search.expression.solr.SolrSearchExpressionValidator"/>
  </_config:query>
  <_config:result>
    <_config:filter
      classname="com.ibm.commerce.catalog.facade.server.services.search.metadata.solr.SolrSearchNavigationSuggestionsResultFilter"/>
  </_config:result>
  <_config:navigationSuggestion>
    <_config:suggestion
      handler="com.ibm.commerce.catalog.facade.server.services.search.suggestion.solr.CategoryNavigationSuggestion"
      name="Category"/>
    <_config:suggestion
      handler="com.ibm.commerce.catalog.facade.server.services.search.suggestion.solr.BrandNavigationSuggestion" name="Brand"/>
    <_config:suggestion
      handler="com.ibm.commerce.catalog.facade.server.services.search.suggestion.solr.WebContentNavigationSuggestion"
      name="Articles">
      <_config:param name="totalNumber" value="100"/>
    </_config:suggestion>
  </_config:navigationSuggestion>
</_config:profile>
```

This slide shows an example of a navigation suggestion search profile. To register a custom suggestion grouping, you add a new navigation suggestion handler. The default handlers are shown in purple on the slide.

## Add UserData for search term association statistics

- Default implementation
  - Statistics are spread across multiple database rows
  - Fetch command invokes JDBCQueryService to perform query and uses result set to assemble the Statistics noun part
- To add custom statistics
  - Extend an existing Fetch command to add custom statistics into the UserData field
    - AbstractFetchSearchTermAssociationStatisticsCmdImpl
    - FetchSearchTermAssociationStatisticsCmdImpl
    - FetchSearchTermAssociationStatisticsByKeywordCmdImpl
  - Implement these methods to retrieve your custom statistics
    - getSelectAccessProfile
    - getTotalCountAccessProfile
    - buildQueryParameterMap
    - buildNoun

Adding custom STA statistics through UserData requires some customization due to the way statistics are stored. Since there can be more than one row in the SRCHSTAT table for each keyword, the Get service for STA statistics is implemented a bit differently. It uses a Fetch command to get the SQL template and invokes the JDBCQueryService to retrieve the multi-row result set from database. The Fetch command then combines the results and populates the noun part.

To include custom statistics data in the UserData field, it is necessary to create a custom version of the Fetch command. If you want to completely replace the statistics data, you can extend the abstract command. To keep the existing statistics and add your own data, extend one of the existing implementations listed on the slide. There are four methods you need to override which are also listed.

## Summary

- Solution overview
- Architecture updates
- Customization

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

## References

- Working with search rule experiments
  - <http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center.doc/tasks/tsbsearchxsupert.htm>
- Working with site search statistics
  - <http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center.doc/tasks/tpnsearchstatscont.htm>
- Catalog navigation service for the storefront
  - <http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.developer.doc/concepts/csdsearchbasednavstore.htm>
- Indexing site content with WebSphere Commerce search
  - <http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.developer.doc/concepts/csdmanagesearchsitecontent.htm>

This slide contains some useful references on WebSphere Commerce search.



## 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\\_SearchArchitectureUpdates.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_SearchArchitectureUpdates.ppt)

This module is also available in PDF format at: [../SearchArchitectureUpdates.pdf](http://SearchArchitectureUpdates.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>

Other company, product, or service names may be trademarks or service marks of others.

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.