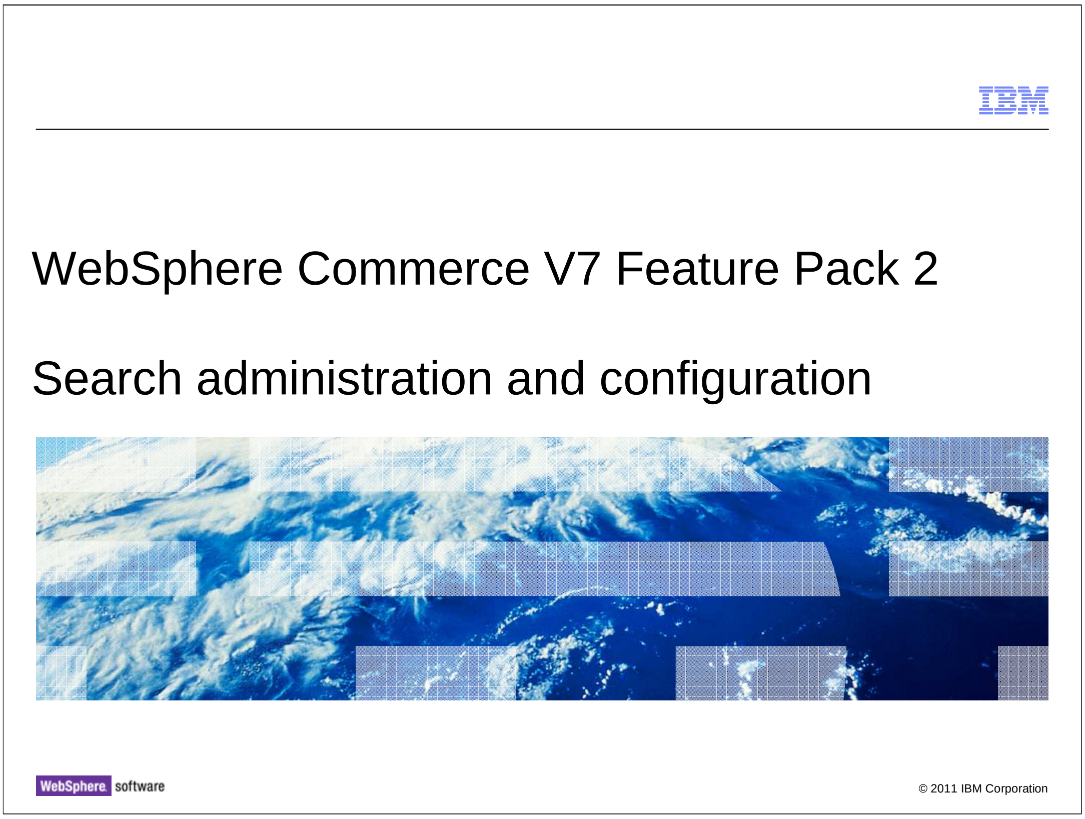


WebSphere Commerce V7 Feature Pack 2

Search administration and configuration

This presentation provides an overview of the administration and configuration steps for the WebSphere Commerce Search solution. You should be familiar with the WebSphere Commerce Search solution before viewing this presentation. An overview of the search solution can be found in the Programming Model section.

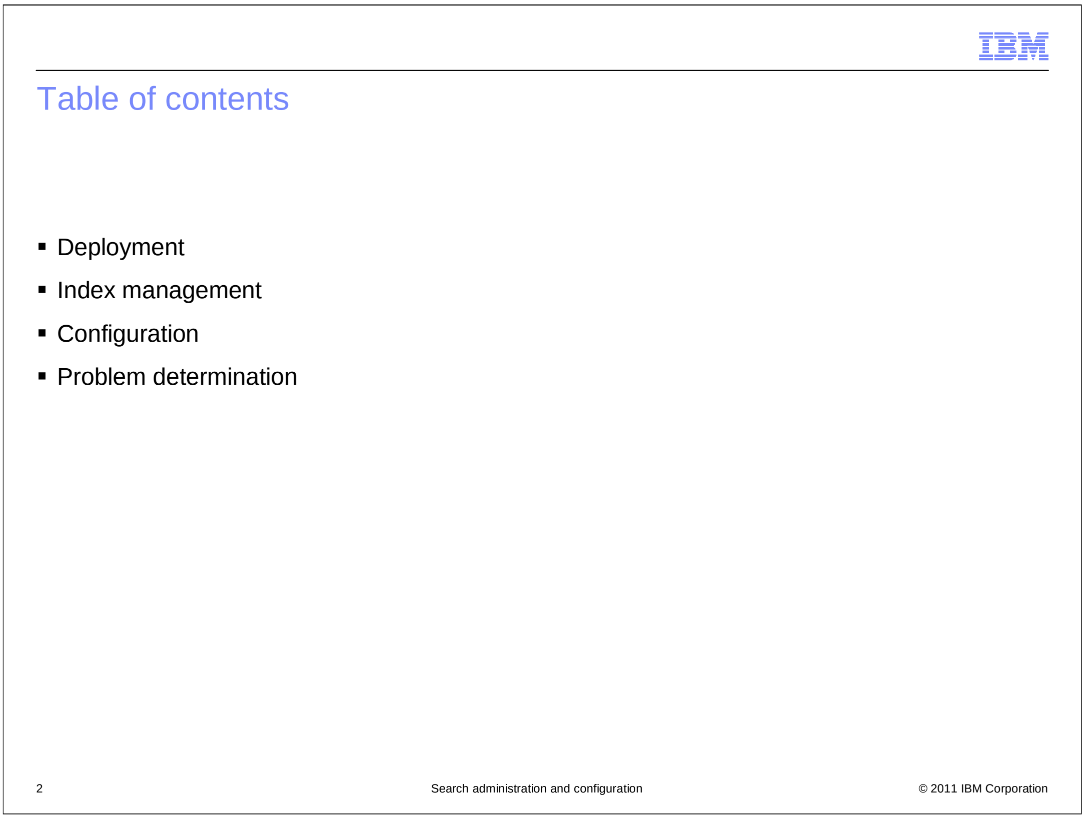


Table of contents

- Deployment
- Index management
- Configuration
- Problem determination

2 Search administration and configuration

The presentation begins with an overview of the steps for deploying WebSphere Commerce Search. Following deployment is an overview of index management. The next section introduces the configuration options and the presentation concludes with some problem determination tips.

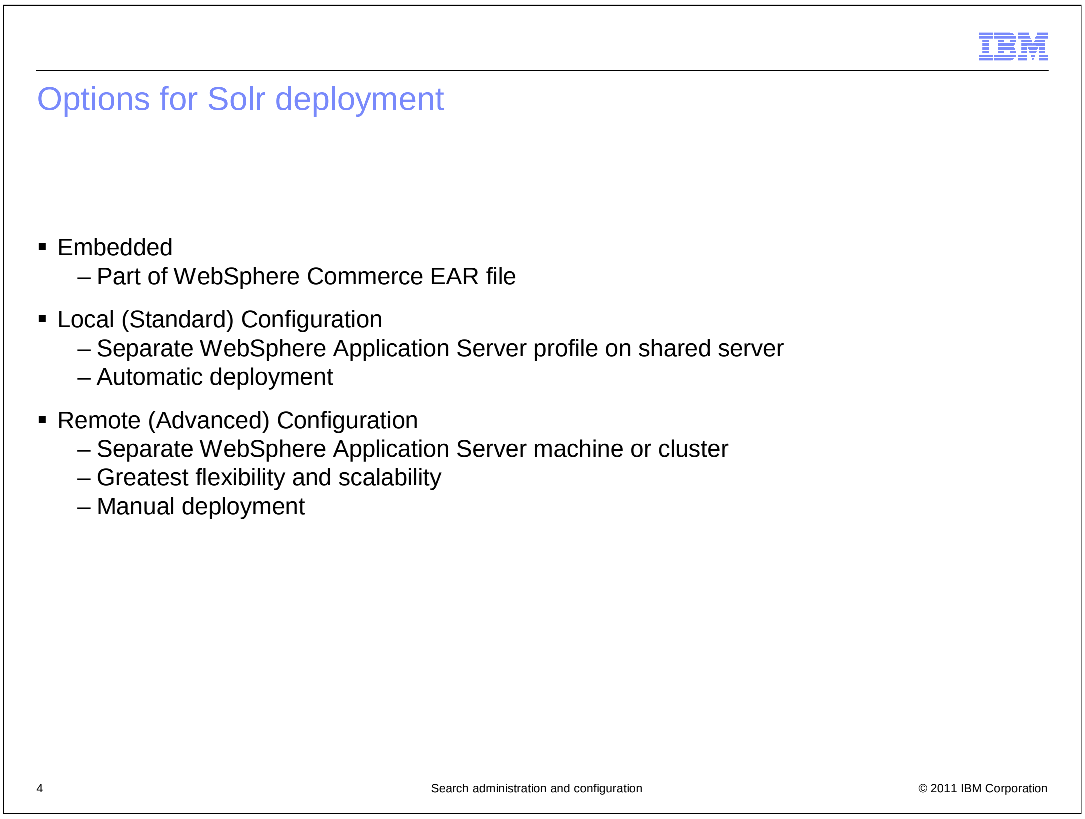
Section

Deployment

3

Search administration and configuration

This section covers deployment of WebSphere Commerce Search.



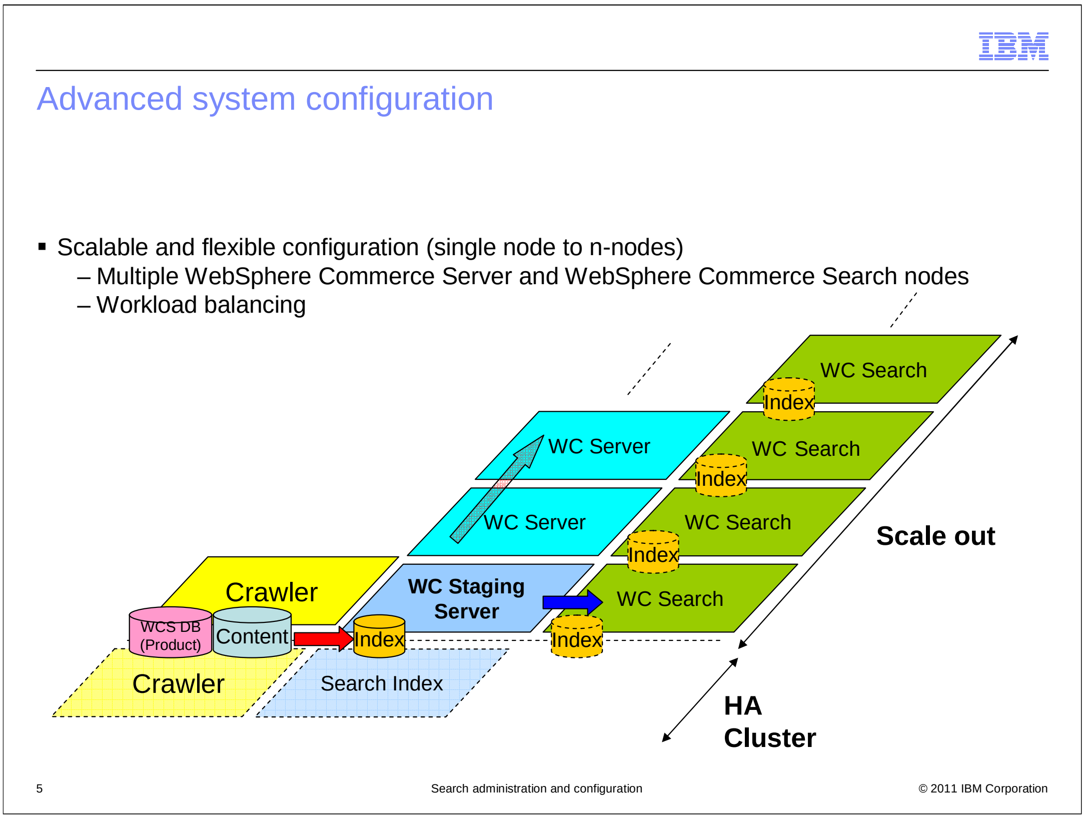
Options for Solr deployment

- Embedded
  - Part of WebSphere Commerce EAR file
- Local (Standard) Configuration
  - Separate WebSphere Application Server profile on shared server
  - Automatic deployment
- Remote (Advanced) Configuration
  - Separate WebSphere Application Server machine or cluster
  - Greatest flexibility and scalability
  - Manual deployment

4 Search administration and configuration © 2011 IBM Corporation

Solr does not come with its own installer. To simplify the deployment of the Solr war file, WebSphere Commerce provides scripts that assist with deployment. There are three deployment options for your WebSphere Commerce search solution. The first option, embedded mode, applies to WebSphere Commerce Developer only. In this case it becomes part of your WebSphere Commerce EAR file and runs directly on your WebSphere Commerce test server.

The other two deployment options are standard and advanced. The standard deployment is suitable for small sites with fairly low traffic volumes. In this case, Solr is configured to run in another profile on your existing WebSphere Application Server . Similar to the embedded configuration, this option can be automatically configured during feature enablement. The advanced deployment option is used in most production scenarios. In this scenario, Solr is manually deployed on it’s own WebSphere Application Server providing greater flexibility and scalability for your solution. Although it is a manual configuration process, scripts are provided to assist in the deployment.

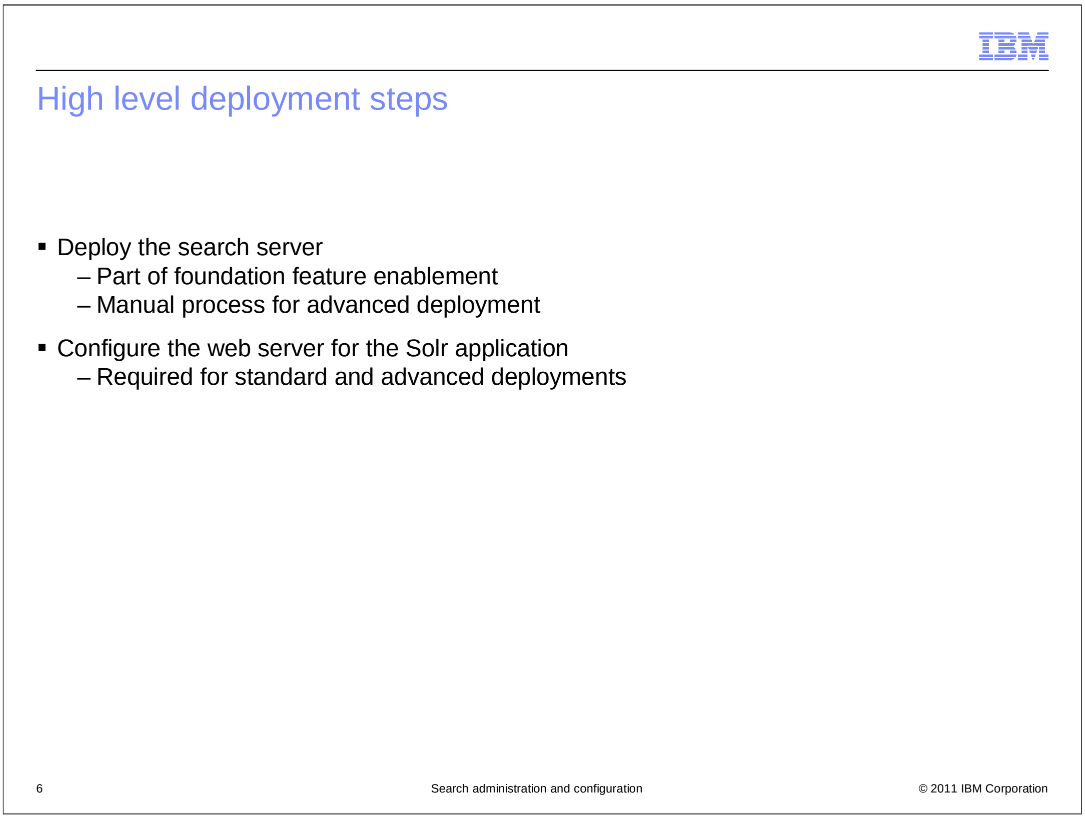


Advanced system configuration

- Scalable and flexible configuration (single node to n-nodes)
  - Multiple WebSphere Commerce Server and WebSphere Commerce Search nodes
  - Workload balancing

5 Search administration and configuration © 2011 IBM Corporation

This diagram shows how the advanced configuration gives you the most flexibility and scalability in your search solution. You can have multiple WebSphere Commerce server nodes and WebSphere Commerce search nodes with a workload balancer managing traffic to the servers. Your search index is built in a staging environment along with your WebSphere Commerce application. When updates are ready for production, the WebSphere Commerce application and data are propagated to the WebSphere Commerce production nodes and the Solr index is replicated to the production master index. From there it is replicated across the WebSphere Commerce search nodes.



High level deployment steps

- Deploy the search server
  - Part of foundation feature enablement
  - Manual process for advanced deployment
- Configure the web server for the Solr application
  - Required for standard and advanced deployments

6 Search administration and configuration © 2011 IBM Corporation

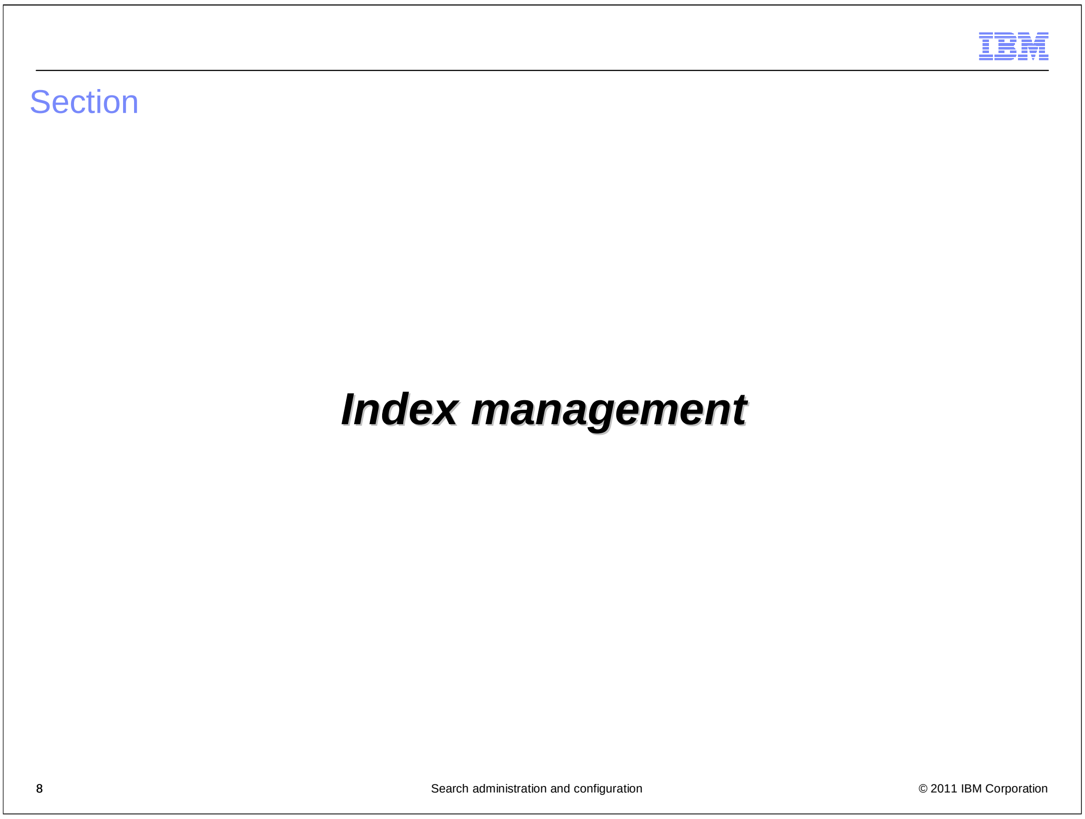
This slide outlines the high level steps for deploying your search server. The Administering WebSphere Commerce search section of the WebSphere Commerce Information Center contains the details. The first step is to deploy the Solr war file. For embedded and standard deployments, this is included in foundation feature enablement. For advanced deployment it is a manual process with scripts provided. The next step for standard and advanced deployments is to configure the web server for the Solr application. WebSphere Commerce must be able to send search results to the search server.

After deployment

- Publish your store
  - Starter store code in enhancements sar
- Create the search indexes for the master catalog
  - setupSearchIndex.bat -masterCatalogId masterCatalogId
  - Two-part process for advanced deployment
- Enable search based navigation
  - New store function selection option under Catalog tab

Search-based navigation ☑

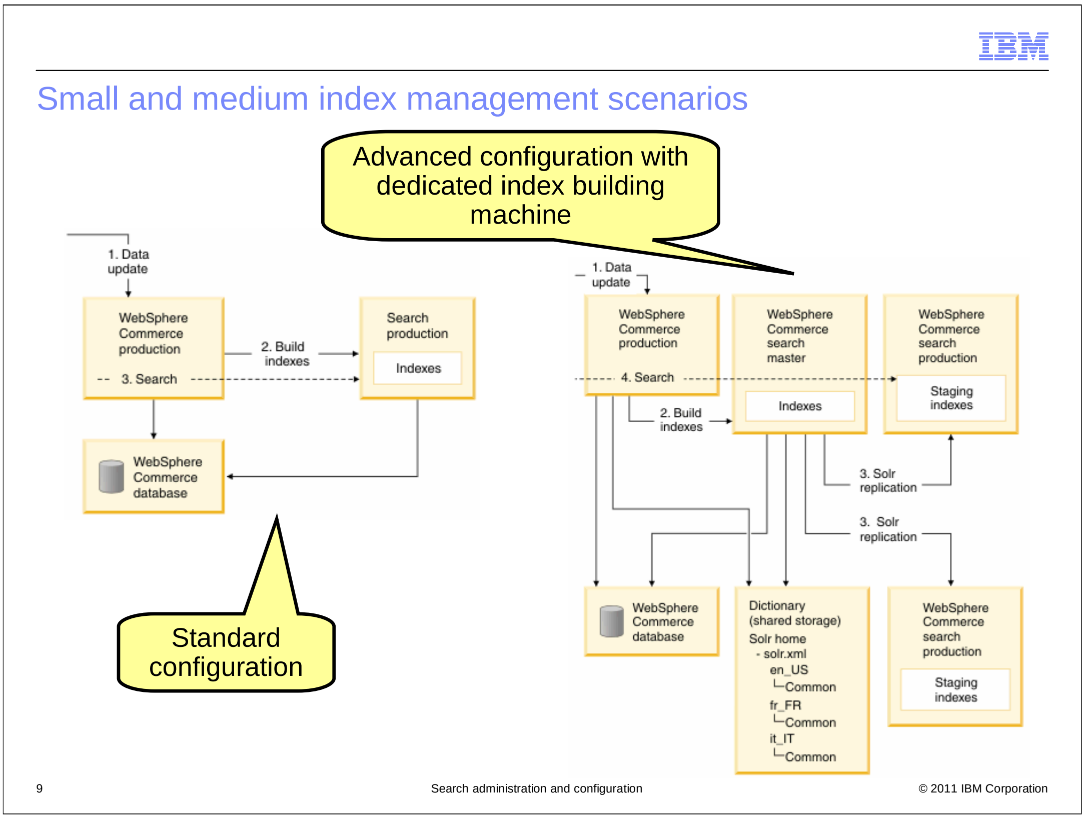
After deploying Solr, the next step is to publish your store. The enhancement sar files provided with feature pack 2 contain examples of the storefront code needed to call the catalog search services. Once your store is published, take note of the master catalog ID. The setupSearchIndex script provided with feature pack 2 identifies the languages supported by your store and creates each required index. There is a search index created for each combination of master catalog and supported language. With the advanced deployment option, creating the search index is a two part process. The script is run on both the WebSphere Commerce server and the search server using different parameters. The final step is to enable the search-based navigation option for your store. This store function selection option is enabled using the Store Management tool. It is located on the Catalog tab.



Section

Index management

This section covers index management.



Small and medium index management scenarios

Advanced configuration with dedicated index building machine

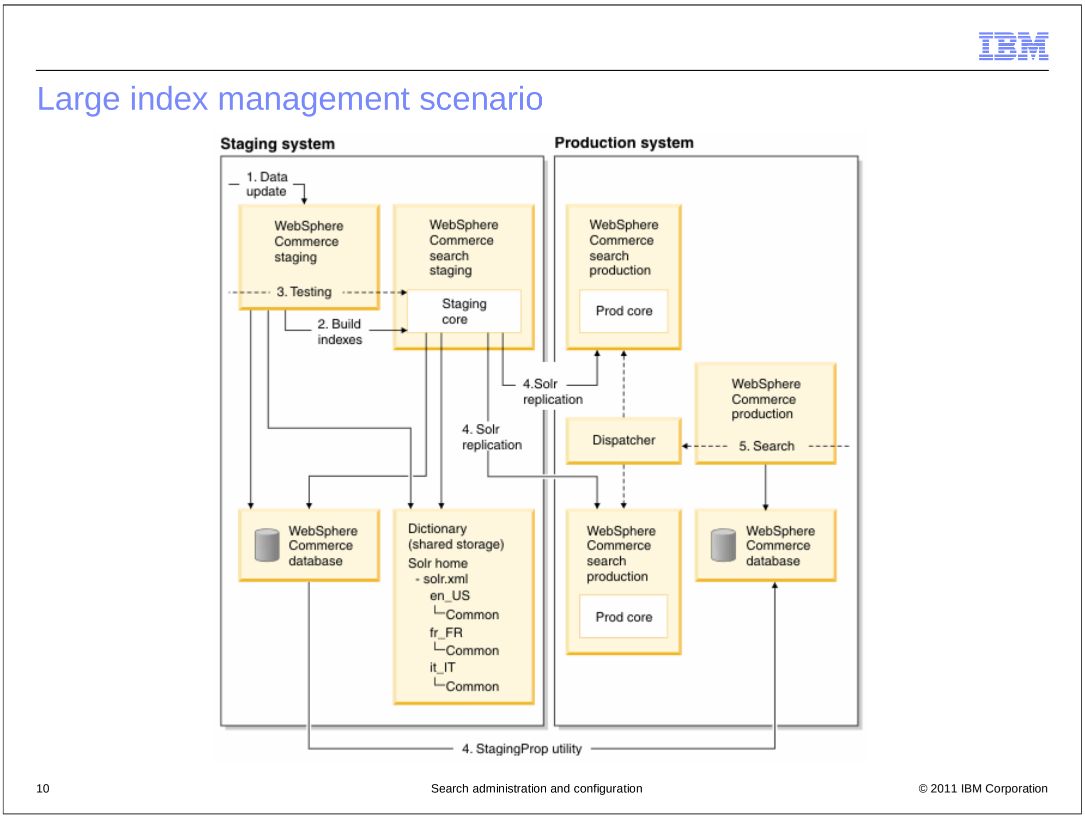
Standard configuration

1. Data update
WebSphere Commerce production
2. Build indexes
3. Search
Search production
Indexes
WebSphere Commerce database
1. Data update
WebSphere Commerce production
4. Search
2. Build indexes
WebSphere Commerce search master
Indexes
WebSphere Commerce search production
Staging indexes
3. Solr replication
3. Solr replication
WebSphere Commerce database
Dictionary (shared storage)
Solr home
- solr.xml
en_US
└Common
fr_FR
└Common
it_IT
└Common
WebSphere Commerce search production
Staging indexes

9 Search administration and configuration © 2011 IBM Corporation

The standard deployment configuration, with WebSphere Commerce and Solr sharing the same application server, is appropriate for managing small indexes. When data is updated in WebSphere Commerce, the production search index must be updated. Performing index updating activities on the production index can slow the performance of customer searches. There is also a delay in updates being reflected in the search index.

The improve performance, small and medium deployments can choose to include a dedicated index building machine. This requires the advanced deployment option where Solr is installed on a separate application server. When data is updated in WebSphere Commerce, the search master machine updates the index and then uses Solr index replication to update the production search servers. While there is no performance hit to production, there is still a delay in updates being reflected in the search index.

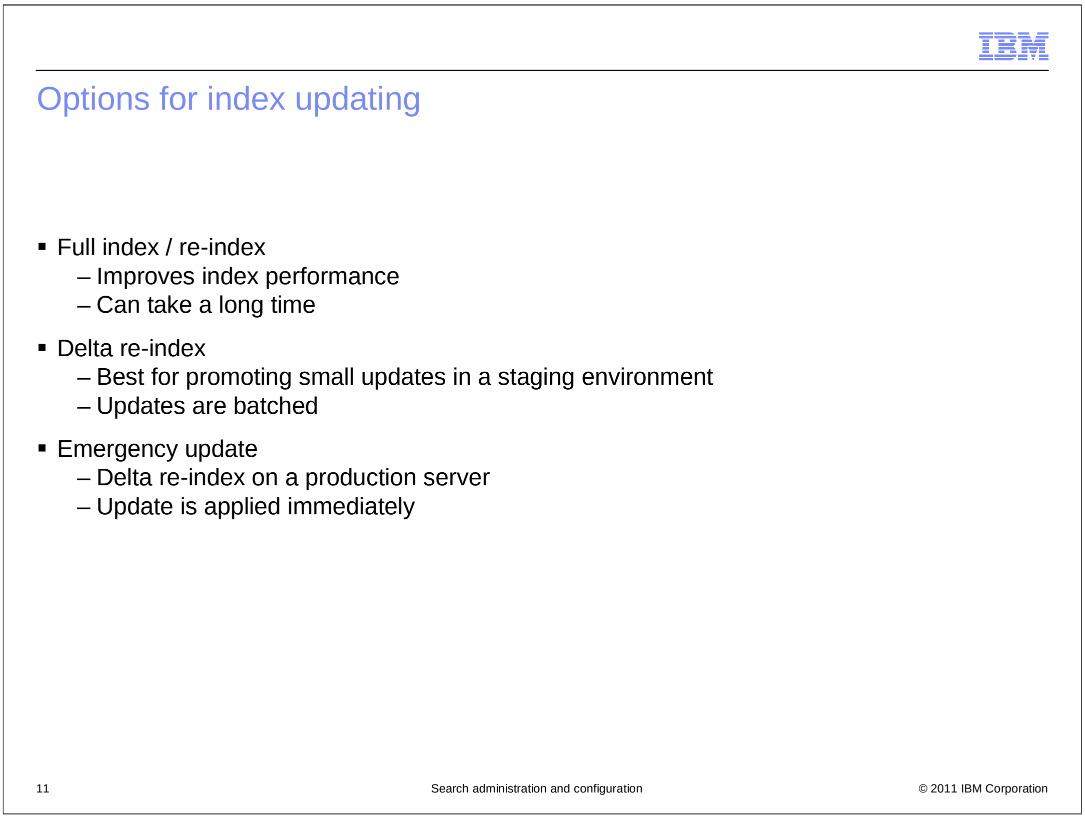


Large index management scenario

Staging system
Production system
1. Data update
WebSphere Commerce staging
WebSphere Commerce search staging
3. Testing
Staging core
2. Build indexes
4.Solr replication
4. Solr replication
WebSphere Commerce search production
Prod core
WebSphere Commerce production
Dispatcher
5. Search
WebSphere Commerce database
Dictionary (shared storage)
Solr home
- solr.xml
en_US
└Common
fr_FR
└Common
it_IT
└Common
WebSphere Commerce search production
Prod core
WebSphere Commerce database
4. StagingProp utility

10 Search administration and configuration © 2011 IBM Corporation

For large index scenarios, a full staging system is recommended. All data updates and index building activities are performed within the staging environment. When the updates are tested and ready for production, the WebSphere Commerce updates are promoted to production using the StagingProp utility and the search index is replicated to the production search servers using Solr replication. There is no performance hit to the production system and the WebSphere Commerce database and search index are always synchronized.



Options for index updating

- Full index / re-index
  - Improves index performance
  - Can take a long time
- Delta re-index
  - Best for promoting small updates in a staging environment
  - Updates are batched
- Emergency update
  - Delta re-index on a production server
  - Update is applied immediately

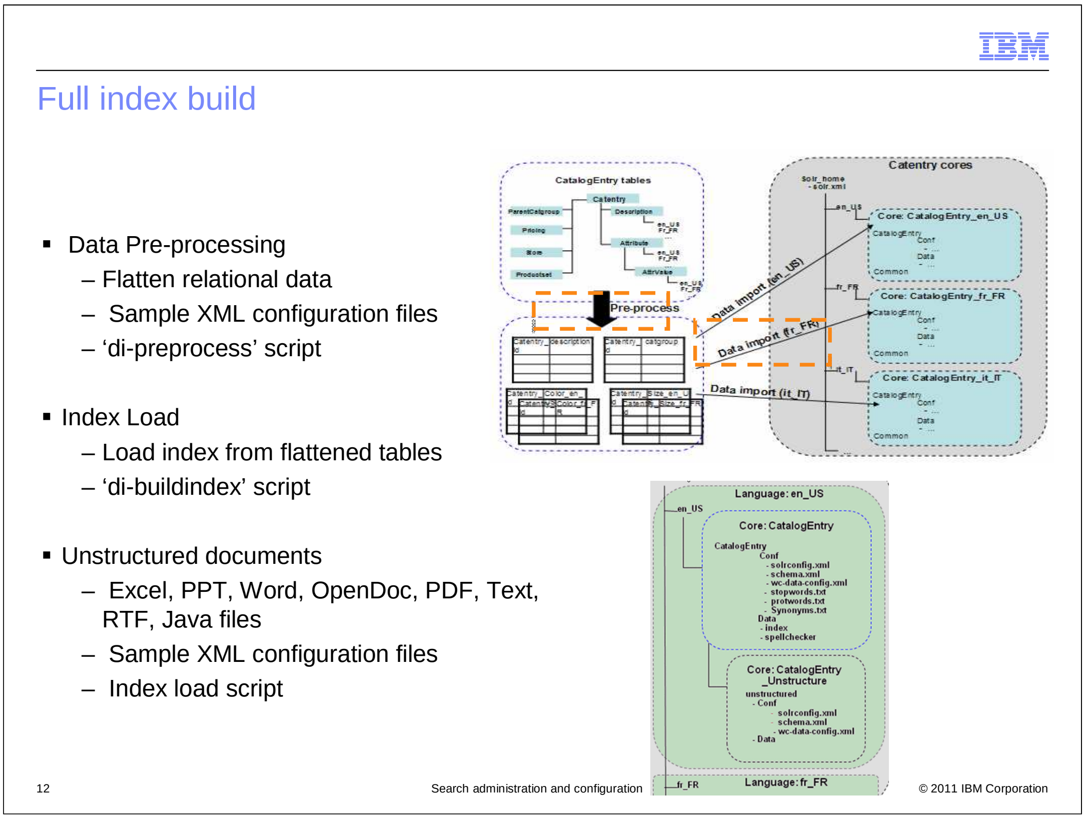
11 Search administration and configuration © 2011 IBM Corporation

The method for updating an index depends on the type of update being made. This slide outlines three index updating options.

Full indexing or re-indexing of the catalog takes the longest time but is required in certain scenarios. When you first load your index, or make significant changes to your catalog data, a full index update is required. If you have an authoring server you also need to perform a full index update to pick up workspace updates. Rebuilding your index occasionally, even if large changes haven’t been made, is recommended to keep it performing at its best.

In between full index builds, delta re-indexing can be used to pick up small updates and make them quickly available in your staging index for testing. This takes less time than fully rebuilding the index, but should be limited to simple updates on small amounts of data. Delta updates are batched and processed when an index synchronization is requested.

In situations where data in a production index is incorrect and must be fixed you can perform an emergency update. This is essentially a delta re-index applied to your production environment. The index being updated is in your production environment, but is not a live machine. The update is applied to the index immediately rather than waiting for a batch job to run. Once the update is complete, the index is replicated across your live production indexes.



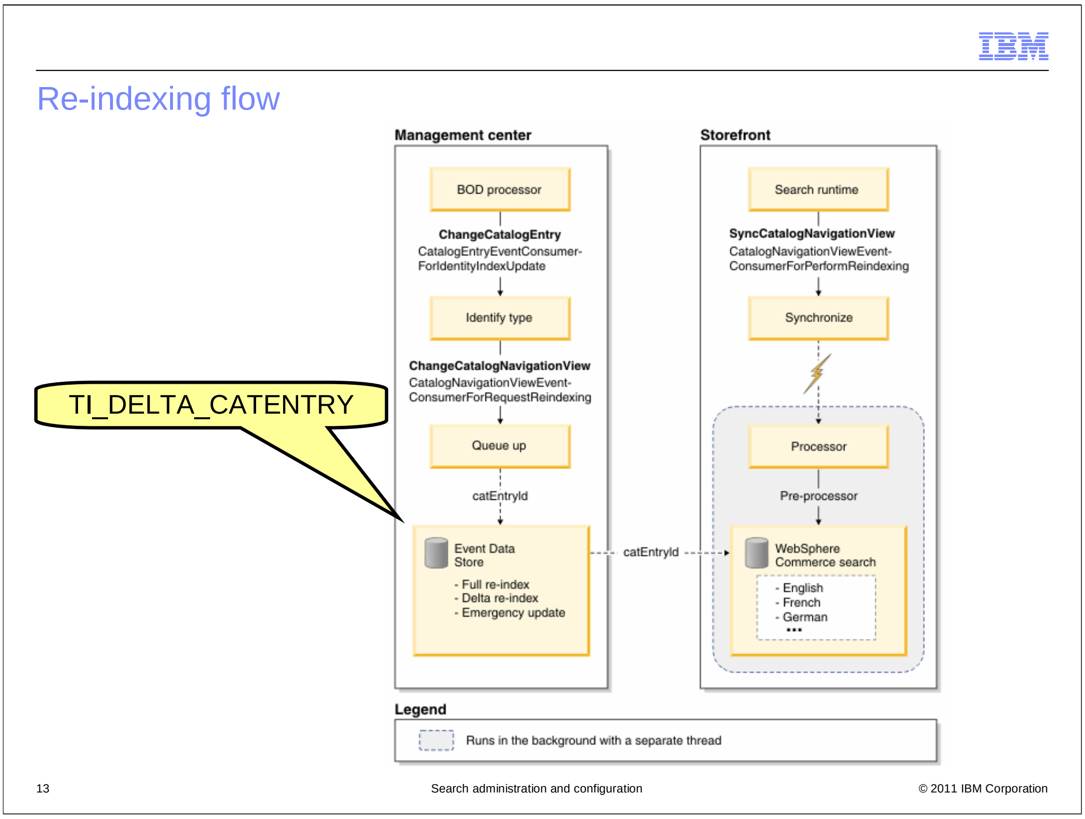
Full index build

- Data Pre-processing
  - Flatten relational data
  - Sample XML configuration files
  - 'di-preprocess' script
- Index Load
  - Load index from flattened tables
  - 'di-buildindex' script
- Unstructured documents
  - Excel, PPT, Word, OpenDoc, PDF, Text, RTF, Java files
  - Sample XML configuration files
  - Index load script

Full index building begins with a pre-processing step. Pre-processing is used to flatten the relational data in your WebSphere Commerce database and reduce the time needed to load the index. This pre-processing step typically takes longer to complete than the index loading step. XML configuration files are used to define the flat table structure and which data requires pre-processing. These files are discussed in more detail in the configuration section of this presentation. The di-preprocess script is provided to perform the pre-processing.

Once the data has been flattened, it is ready to load into the index. The di-buildindex script calls the Solr data import handler to load the index. Indexes for each language and master catalog are loaded.

Unstructured content can only be loaded using the full index process. A variety of document types are supported by WebSphere Commerce search and the solution can be extended to support additional document types. Unstructured content also undergoes a preprocessing step and the index is loaded by the same di-buildindex script used for the catalog index load.



Re-indexing flow

TI_DELTA_CATENTRY

Management center

BOD processor

ChangeCatalogEntry
CatalogEntryEventConsumer-ForIdentityIndexUpdate

Identify type

ChangeCatalogNavigationView
CatalogNavigationViewEvent-ConsumerForRequestReindexing

Queue up

catEntryId

Event Data Store

- Full re-index
- Delta re-index
- Emergency update

catEntryId

Storefront

Search runtime

SyncCatalogNavigationView
CatalogNavigationViewEvent-ConsumerForPerformReindexing

Synchronize

Processor

Pre-processor

WebSphere Commerce search

- English
- French
- German
...

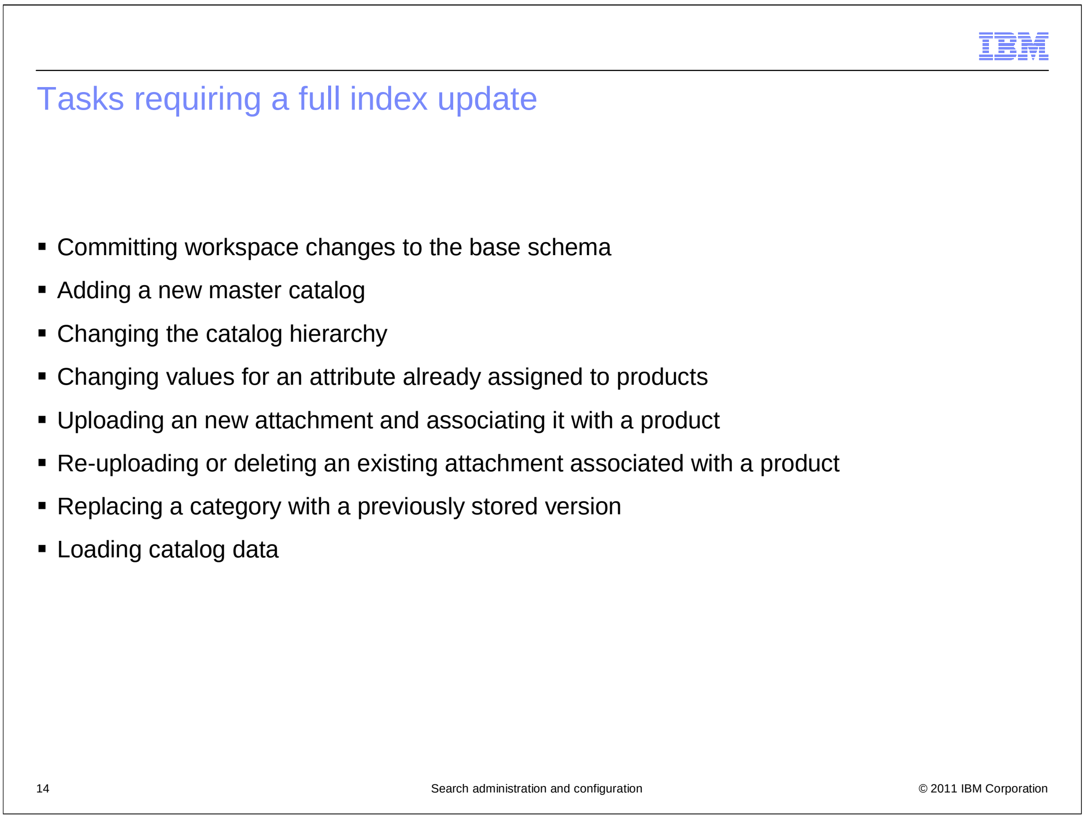
Legend

Runs in the background with a separate thread

13 Search administration and configuration © 2011 IBM Corporation

The diagram on this slide shows the re-indexing flow. When you update an existing product in Management Center, an event called ChangeCatalogEntry is issued. A pre-configured event listener receives the event and determines which type of re-indexing is required. Next, a ChangeCatalogNavigationView event is issued for the product. The catalog entry ID and type of re-indexing required are added to the queue for the next index synchronization.

When you preview your change from the storefront, the first search request triggers an index synchronization event. Catalog entries flagged for delta re-indexing are re-indexed through a separate background process. In store preview mode, you can see the status of the re-indexing.

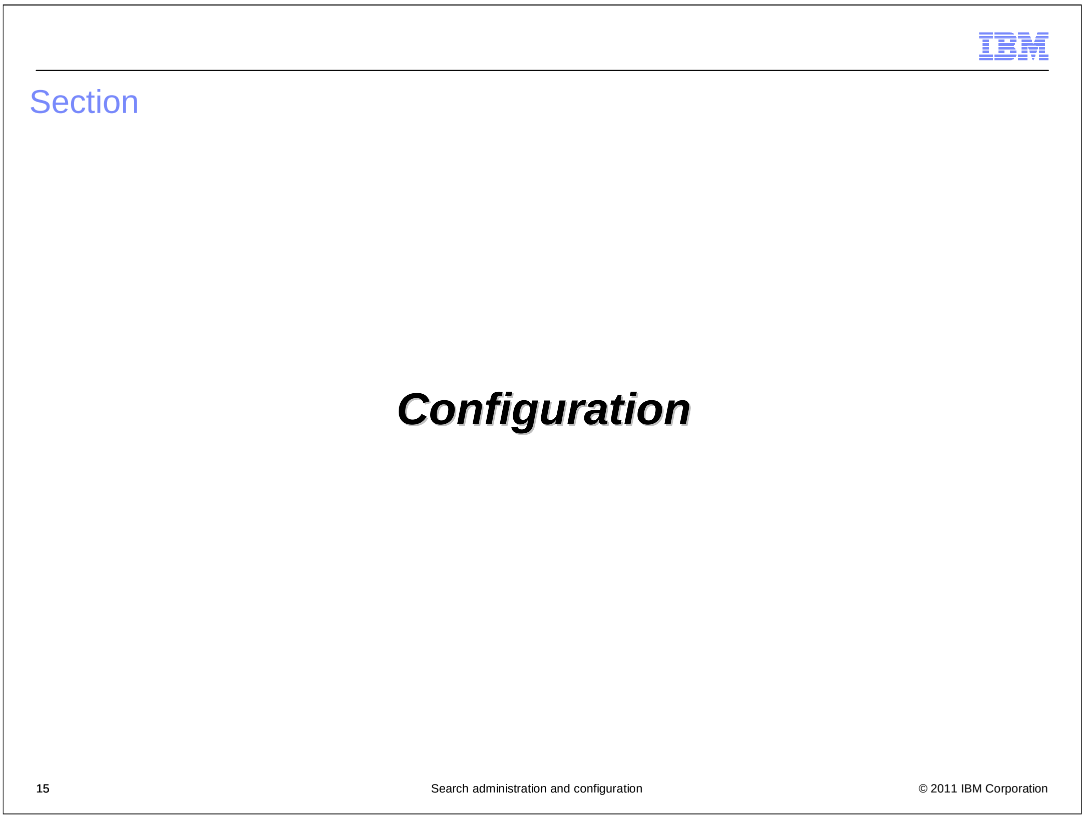


Tasks requiring a full index update

- Committing workspace changes to the base schema
- Adding a new master catalog
- Changing the catalog hierarchy
- Changing values for an attribute already assigned to products
- Uploading an new attachment and associating it with a product
- Re-uploading or deleting an existing attachment associated with a product
- Replacing a category with a previously stored version
- Loading catalog data

14 Search administration and configuration © 2011 IBM Corporation

This slide summarizes the type of business tasks that require a full index update. In general, any task that affects more than one catalog entry requires a full re-index. Also any tasks related to unstructured content require a full re-index. In feature pack 2, delta re-indexing is not supported for changes committed by a workspace into the base schema. To test your workspace changes, a full re-index must be performed.



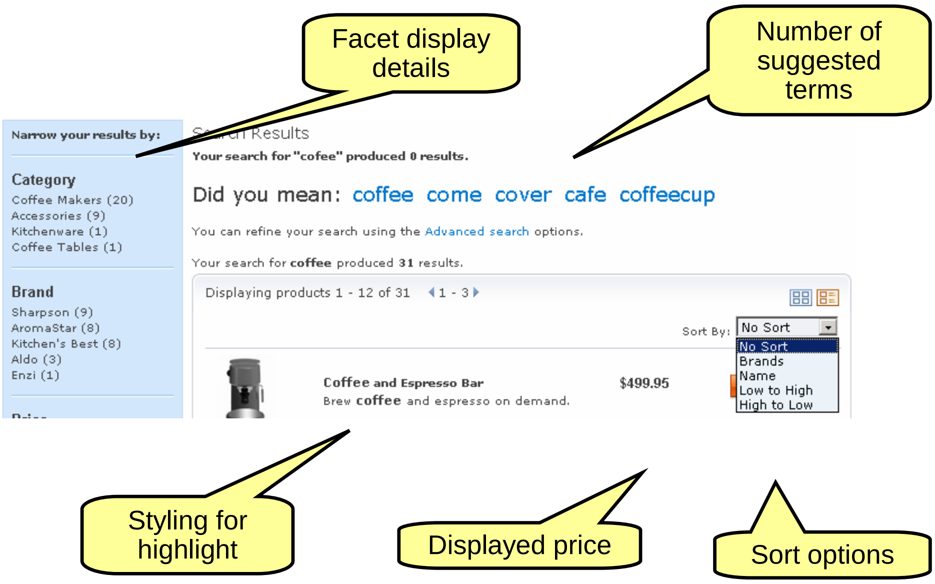
Section

Configuration

15 Search administration and configuration © 2011 IBM Corporation

This section covers configuration of WebSphere Commerce Search.

Search profile – wc-search.xml



Narrow your results by:

Category
Coffee Makers (20)
Accessories (9)
Kitchenware (1)
Coffee Tables (1)

Brand
Sharpson (9)
AromaStar (8)
Kitchen's Best (8)
Aldo (3)
Enzi (1)

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 31 results.

Displaying products 1 - 12 of 31 1 - 3

Sort By: No Sort

No Sort
Brands
Name
Low to High
High to Low

Coffee and Espresso Bar
Brew coffee and espresso on demand.
$499.95

Facet display details

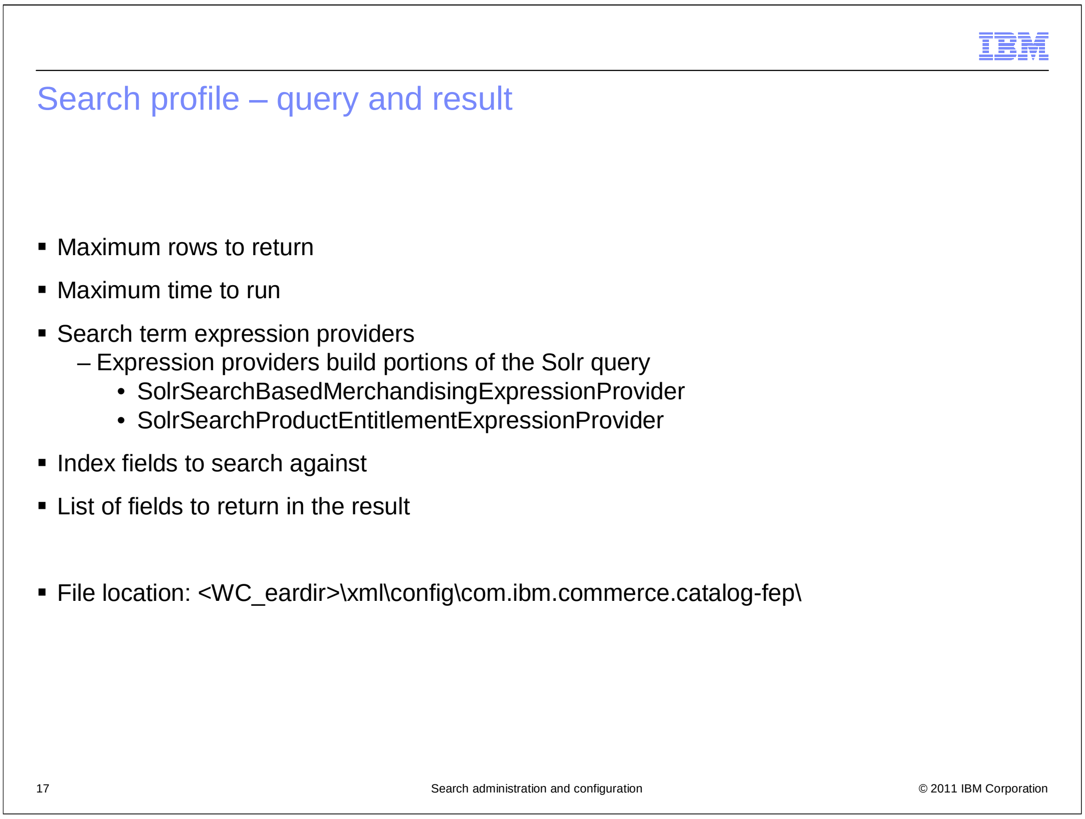
Number of suggested terms

Styling for highlight

Displayed price

Sort options

The search profile controls many aspects of the search behavior in the storefront. You can define multiple search profiles and use them on different pages within your store. This slide shows some examples of display options you can control with the search profile. There are three options for displaying price. The first is to display only the computed price. This means the price is always calculated by the WebSphere Commerce server and never comes from the search index. Price facets and advanced search by price are not used in the store. The second option is to use indexed prices. This is the default setting. The third option is to use computed prices for display, but still allow price facets and advanced search based on indexed prices.



Search profile – query and result

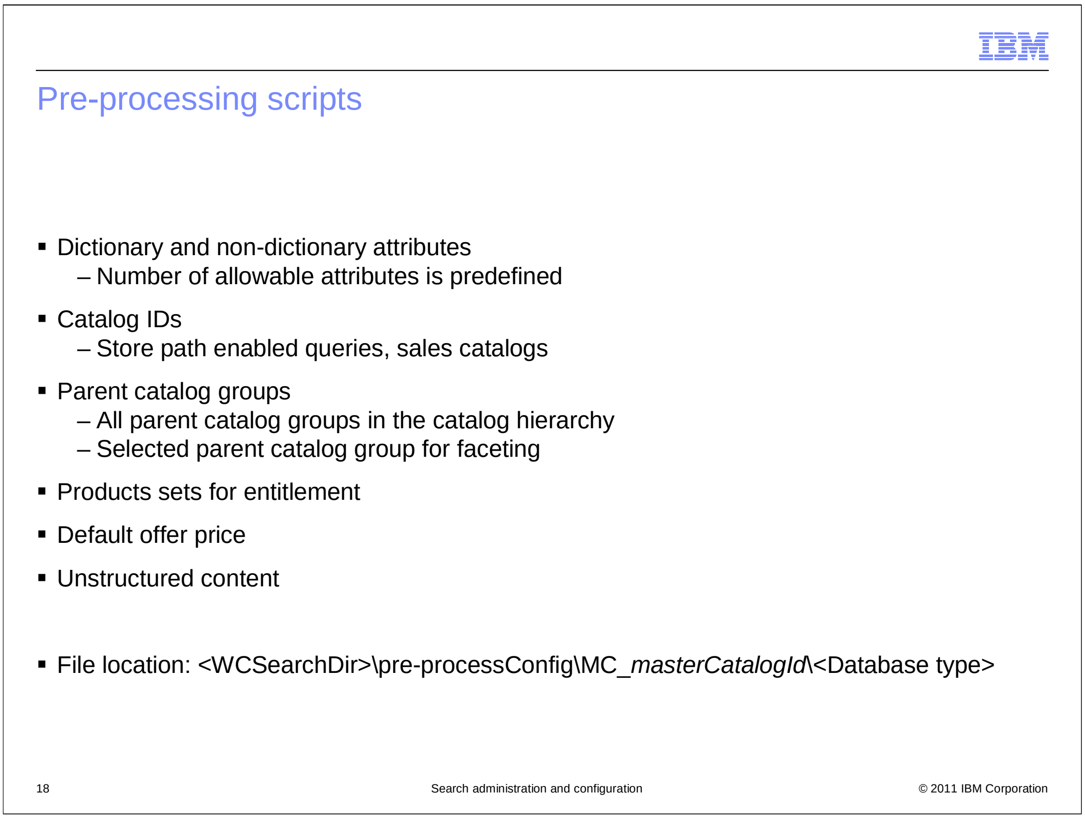
- Maximum rows to return
- Maximum time to run
- Search term expression providers
  - Expression providers build portions of the Solr query
    - SolrSearchBasedMerchandisingExpressionProvider
    - SolrSearchProductEntitlementExpressionProvider
- Index fields to search against
- List of fields to return in the result
- File location: <WC_eardir>\xml\config\com.ibm.commerce.catalog-fep\

17 Search administration and configuration © 2011 IBM Corporation

The search profile also allows you to configure the search query and results. You can set the maximum number of search results to return and the maximum length of time a query can run. As discussed in the search overview and architecture presentation, the search profile is also where you define the expression builders that convert the XPath request from the storefront into a Solr search expression. Each expression provider class listed in the search profile is run by the search service façade and contributes a portion of the overall query. You can configure the search profile to call your own expression providers to customize the Solr search expression or generate a query for another search engine.

The search profile can also specify which fields to run the query against and which fields to return in the result set.

The location of the search profile is shown on the slide. You should create a custom version of this file for your changes rather than edit the file directory. The steps for changing search profile properties can be found in the WebSphere Commerce Information Center.



Pre-processing scripts

- Dictionary and non-dictionary attributes
  - Number of allowable attributes is predefined
- Catalog IDs
  - Store path enabled queries, sales catalogs
- Parent catalog groups
  - All parent catalog groups in the catalog hierarchy
  - Selected parent catalog group for faceting
- Products sets for entitlement
- Default offer price
- Unstructured content
- File location: <WCSearchDir>\pre-processConfig\MC_*masterCatalogId*\<Database type>

18 Search administration and configuration © 2011 IBM Corporation

The preprocessing scripts convert data in the WebSphere Commerce database into a form that can be more efficiently imported into the Solr index. This slide summarizes the types of WebSphere Commerce data that are pre-processed. For both dictionary and non-dictionary product attributes, each attribute is stored in a separate field in the index. A number of attribute fields are predefined and if these are filled, more can be allocated.

Catalog data is pre-processed so that each catalog entry is mapped to all the catalogs it belongs to. This information is used to support store path enabled queries and sales catalogs. The hierarchy for each catalog is flattened to allow search-based navigation and category faceting.

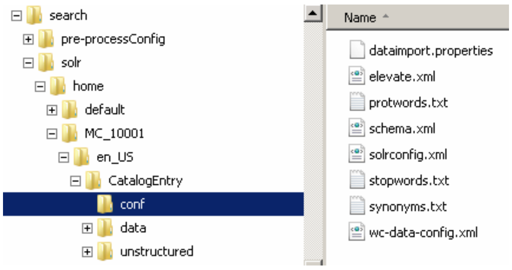
The product sets a catalog entry that its default offer price also belong part of the pre-processing scripts. These scripts also prepare the information in the attachment table for the unstructured content index.

You can extend the pre-processing scripts to prepare your custom tables for indexing. Not all custom tables need to be pre-processed. If your custom table has a simple one-to-one mapping to the CATENTRY table, you do not need to pre-process the data. It can be indexed directly from the custom table.

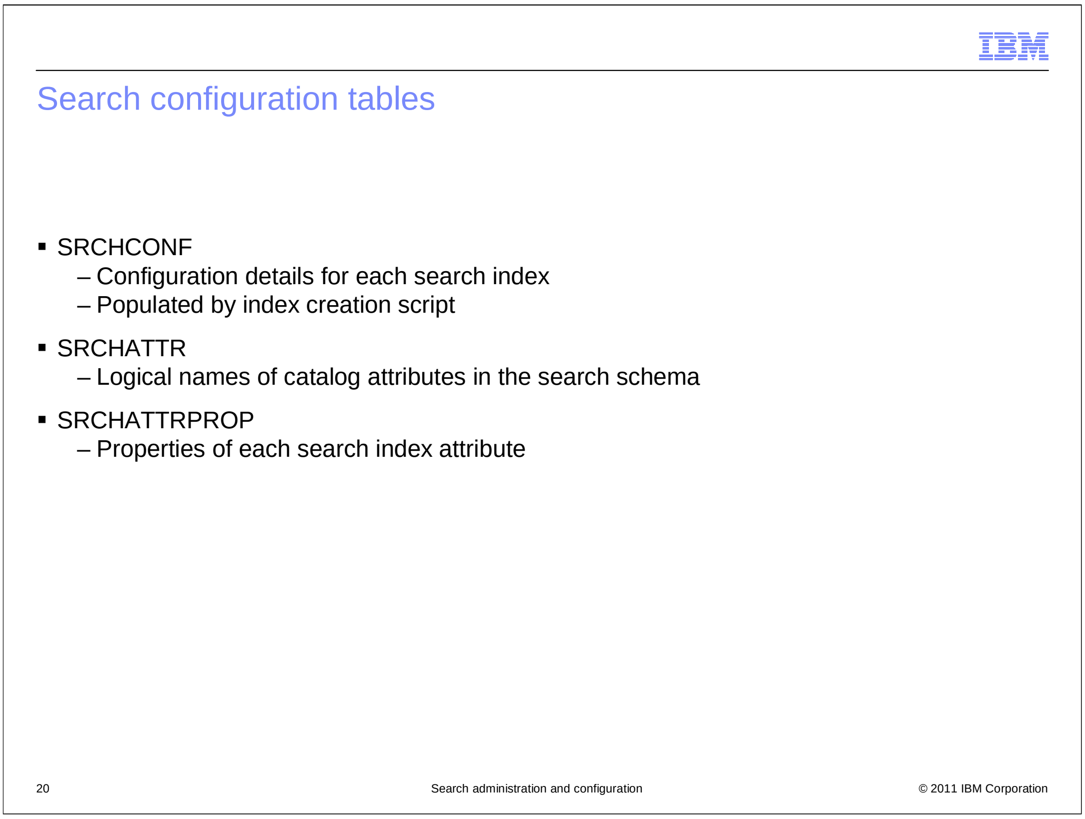
The location of the pre-processing scripts is shown on the slide.

Index configuration

- 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 Data Import Handler



The conf directory for each Solr core contains the configuration files for the index. The schema.xml file defines the index schema. You customize this file to add new fields to the index. 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. If you are adding fields to the index, you need to customize this file to load your new data.



Search configuration tables

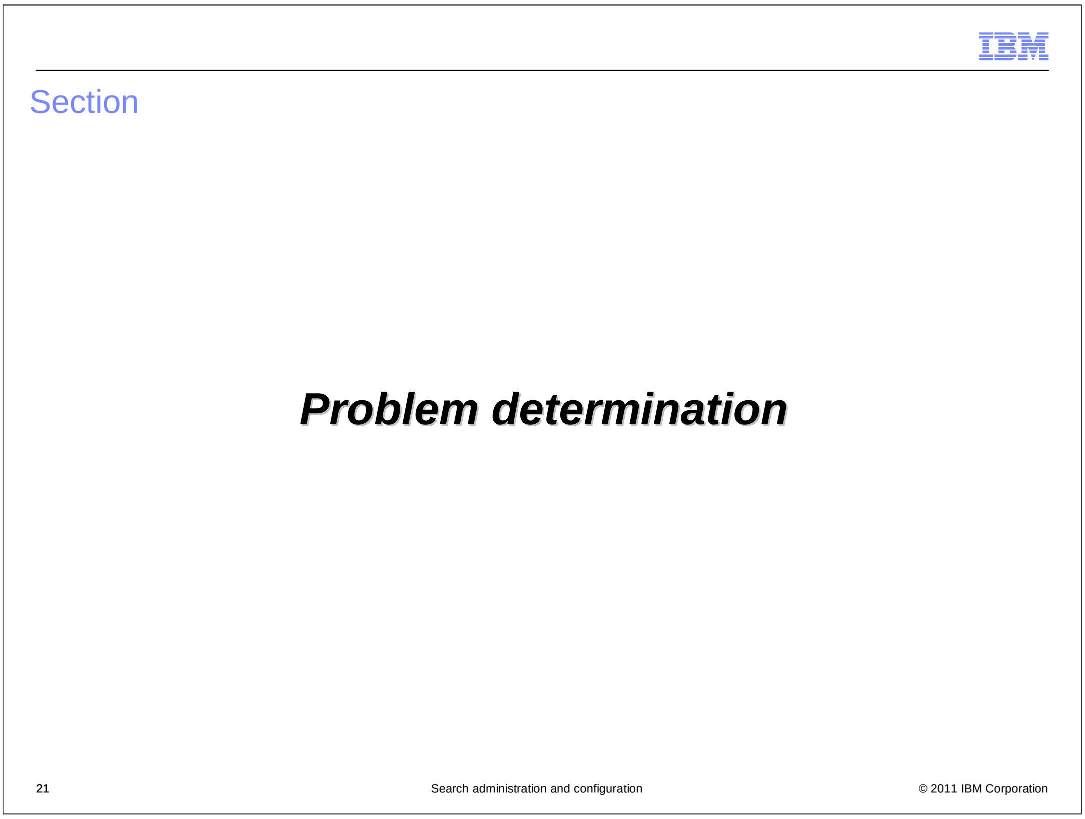
- SRCHCONF
  - – Configuration details for each search index
  - – Populated by index creation script
- SRCHATTR
  - – Logical names of catalog attributes in the search schema
- SRCHATTRPROP
  - – Properties of each search index attribute

20 Search administration and configuration © 2011 IBM Corporation

The WebSphere Commerce search solution introduces three new tables. The first table, search configuration, is populated by the index creation script. It contains general configuration details for each search index.

The second table, search attributes, contains the logical name for each catalog attribute in the search schema. There can be multiple fields in the index, such as tokenized and non-tokenized, that represent a single logical field.

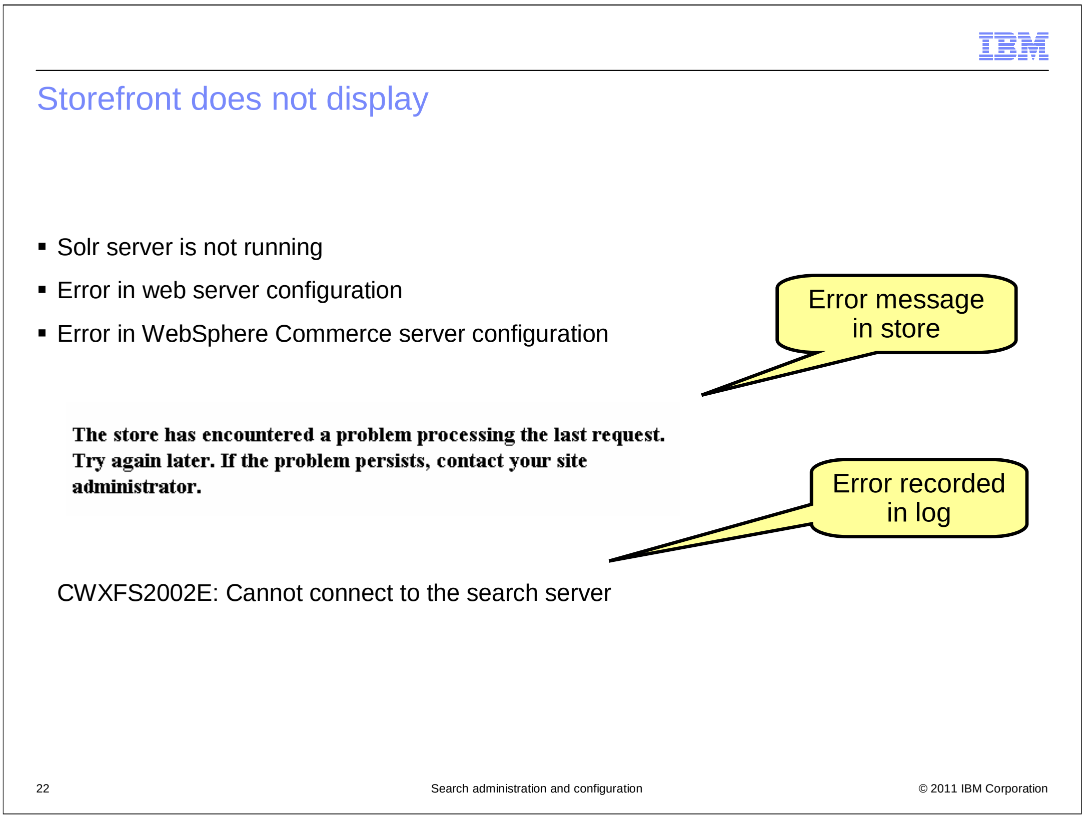
The third table, search attribute properties, contains the usage information for each index field. The usage type for a field can be one or more of display, sort, facet, rank, search, filter, and catalogFilter. Each row in the table represents one usage type for a field. The search attribute and search attribute properties tables are updated automatically when you mark an attribute dictionary attribute as searchable in Management Center.



Section

Problem determination

This section covers some problem determination tips.



Storefront does not display

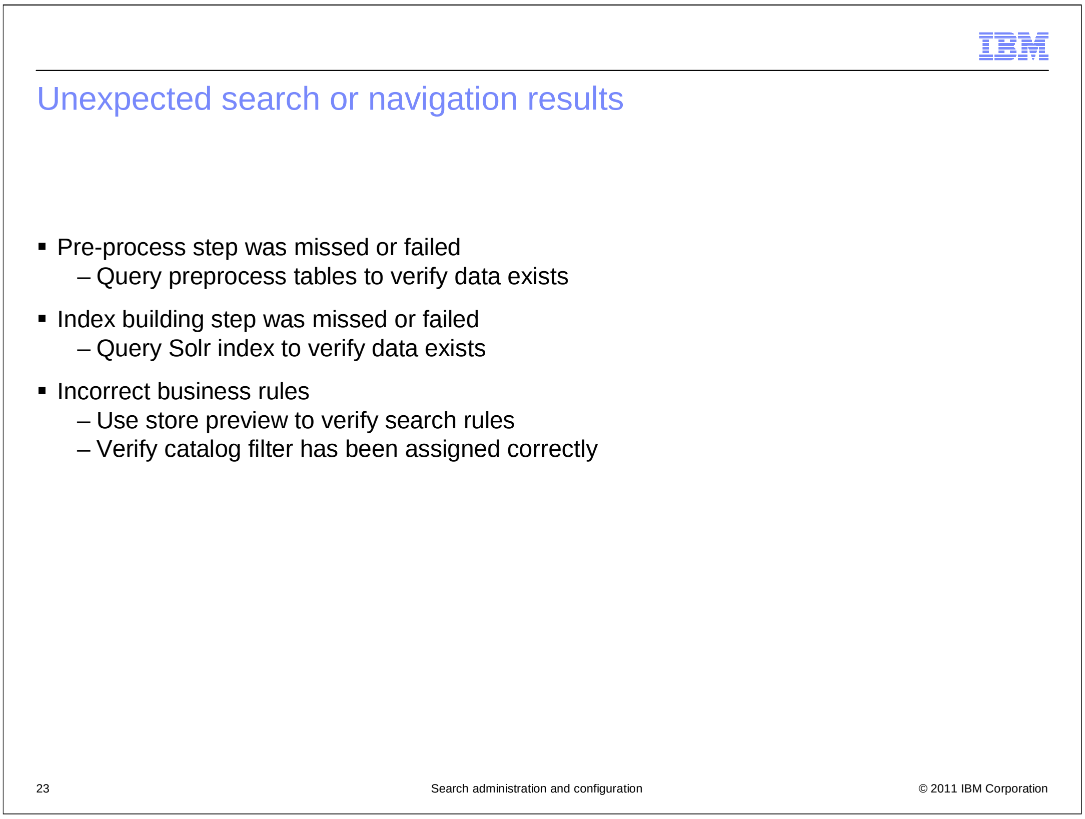
- Solr server is not running
- Error in web server configuration
- Error in WebSphere Commerce server configuration

The store has encountered a problem processing the last request. Try again later. If the problem persists, contact your site administrator.

CWXFS2002E: Cannot connect to the search server

Error message in store
Error recorded in log

Mistakes in configuring WebSphere Commerce search can lead to errors that prevent your storefront from displaying correctly. Problems with the web server or WebSphere Commerce server configuration files are most likely to occur when search is initially configured. This error can also occur if the search server is not running.



Unexpected search or navigation results

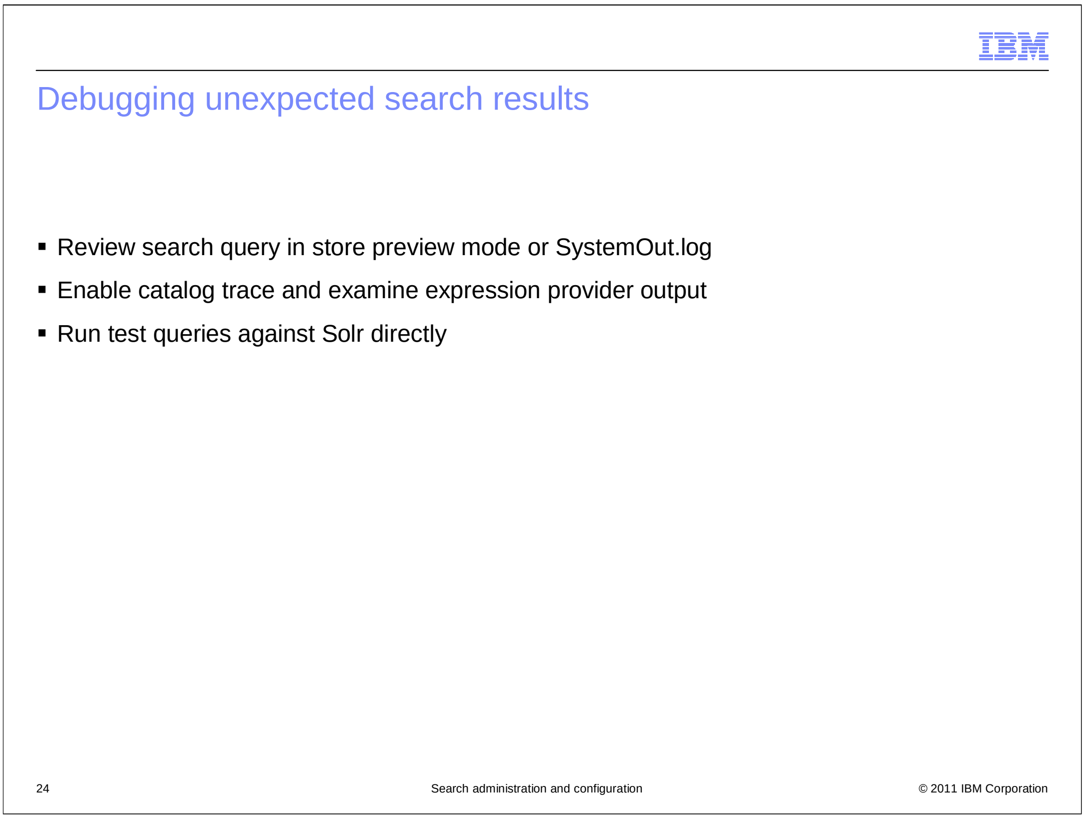
- Pre-process step was missed or failed
  - Query preprocess tables to verify data exists
- Index building step was missed or failed
  - Query Solr index to verify data exists
- Incorrect business rules
  - Use store preview to verify search rules
  - Verify catalog filter has been assigned correctly

23 Search administration and configuration © 2011 IBM Corporation

The most common class of problem is unexpected search results. These problems can be divided into two main root causes.

The first is that data is missing from the index. If search results are missing either products or product updates, a likely cause is the index is out of sync with the WebSphere Commerce database. Querying the Solr index directly allows you to verify whether the data is present. If not, there might have been a problem during pre-processing or index building. The pre-processing tables can be queried to verify the state of the data being loaded into the index. Store preview can be used to check whether a delta update is awaiting processing or a manual indexing step is required.

The second main cause of unexpected search results is incorrect business rules. You can use store preview or the log files to verify which search rules are being applied to adjust the search results. The rule itself might be incorrect or the start date, end date or priority might be wrong. Catalog filters handle inclusion and exclusion of products from the search and navigation results. Verify that your filter rules are correctly assigned and that the filter has been assigned to the contract. If custom expression providers are part of your solution you might have an incorrect Solr fragment being generated or the expression provider might not be correctly configured in the search profile.



Debugging unexpected search results

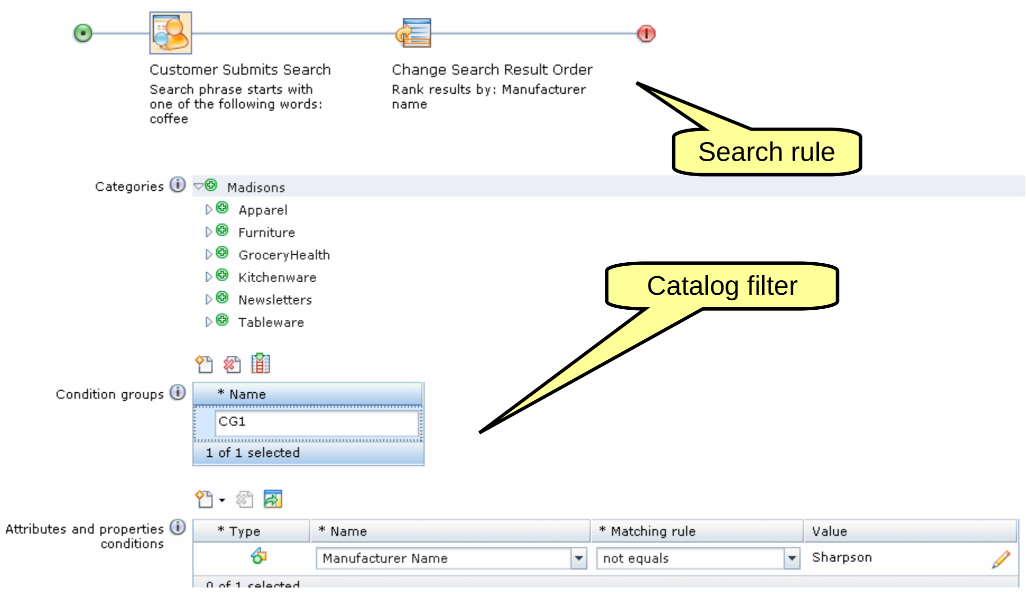
- Review search query in store preview mode or SystemOut.log
- Enable catalog trace and examine expression provider output
- Run test queries against Solr directly

24 Search administration and configuration © 2011 IBM Corporation

The first step in debugging unexpected query results is to examine the query that was run and determine whether it is correct. All queries are logged in SystemOut.log. If you have access to Management Center, queries can also be viewed in store preview mode as can any search rules that contributed to the query. If you find a portion of the query that is incorrect, you can enable the catalog trace and check the input to and output from the expression provider that generates that portion of the query. In situations where the query looks correct, the problem might be with the data contained in the index. You can run test queries against Solr directly to examine the contents of the index. If both the query and the index are correct there might be a problem in the storefront code, in your caching solution or in the business object mediator that creates the CatalogNavigationView noun.

The rest of this presentation shows you how to debug search queries and run test queries against the index.

Business rules applied to a search



Customer Submits Search
Search phrase starts with one of the following words: coffee

Change Search Result Order
Rank results by: Manufacturer name

Search rule

Categories: Madisons
- Apparel
- Furniture
- GroceryHealth
- Kitchenware
- Newsletters
- Tableware

Catalog filter

Condition groups

* Name
CG1

1 of 1 selected

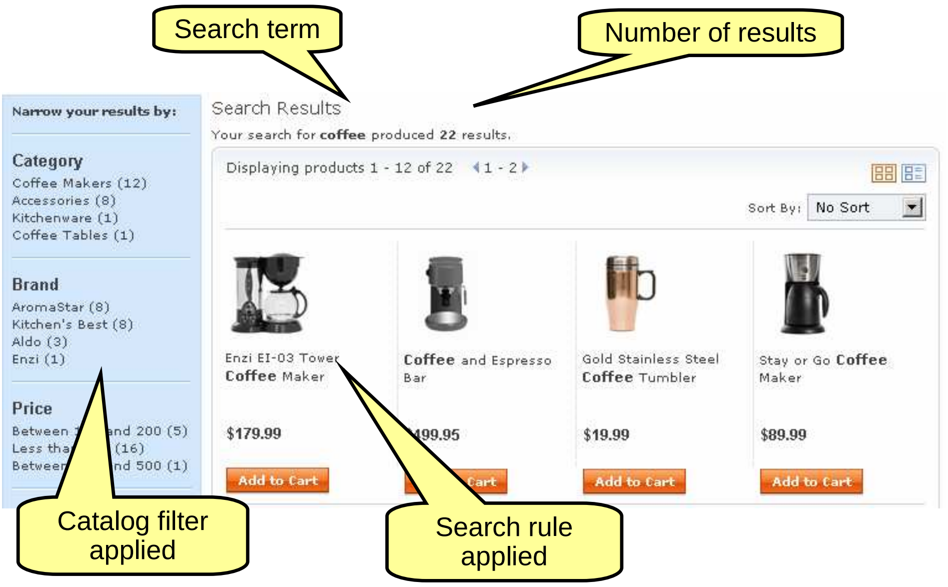
Attributes and properties conditions

* Type	* Name	* Matching rule	Value
	Manufacturer Name	not equals	Sharpson

0 of 1 selected

This slide displays the business rules that are used for demonstration purposes in the upcoming slides. There is one search rule that adjusts the search results ranking by manufacturer name. There is also one catalog filter rule that excludes a specific manufacturer name from the results.

Search results



Narrow your results by:

Category
Coffee Makers (12)
Accessories (8)
Kitchenware (1)
Coffee Tables (1)

Brand
AromaStar (8)
Kitchen's Best (8)
Aldo (3)
Enzi (1)

Price
Between 1 and 200 (5)
Less tha (16)
Betweer nd 500 (1)

Search Results

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

Displaying products 1 - 12 of 22 ◂ 1 - 2 ▸

Sort By: No Sort

Enzi EI-03 Tower **Coffee** Maker	**Coffee** and Espresso Bar	Gold Stainless Steel **Coffee** Tumbler	Stay or Go **Coffee** Maker
$179.99	99.95	$19.99	$89.99
Add to Cart	Cart	Add to Cart	Add to Cart

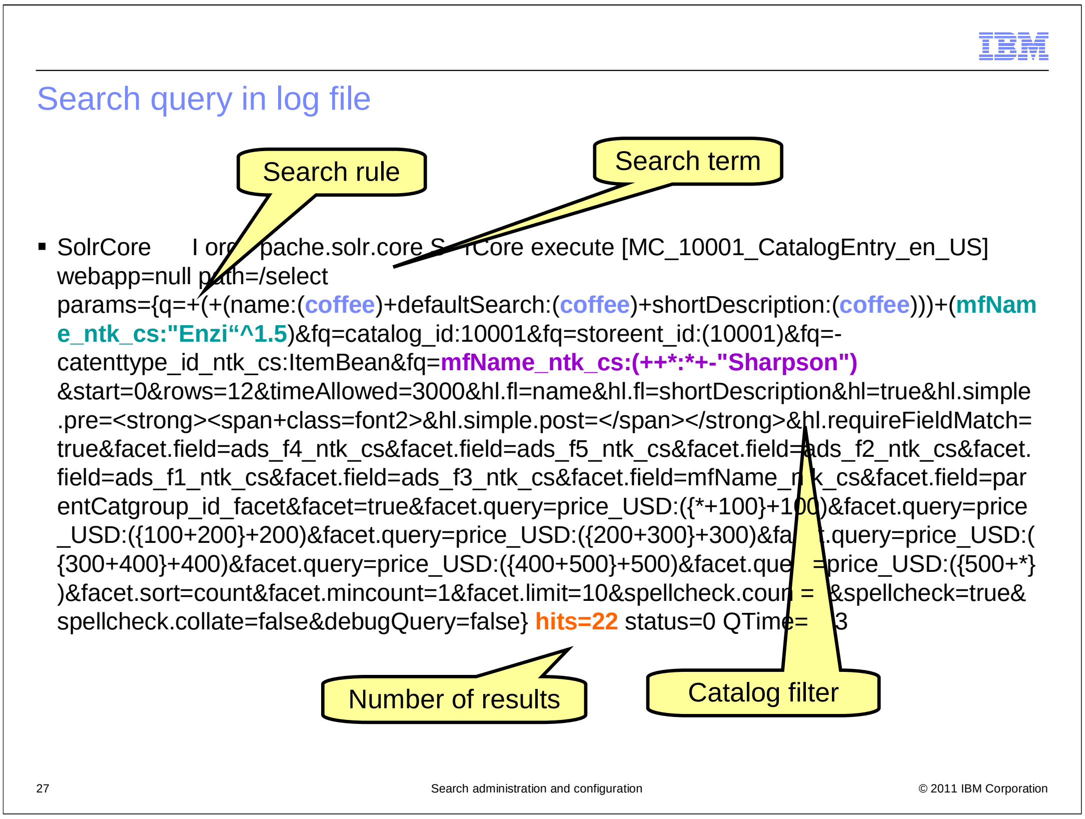
Search term

Number of results

Catalog filter applied

Search rule applied

This slide shows the results of searching for the term ‘coffee’ with these business rules in place. The search rule ranks Enzi products higher in the search results and the catalog filter removes all results from the Sharpson manufacturer.



Search query in log file

Search rule

Search term

- SolrCore I org.apache.solr.core.SolrCore execute [MC_10001_CatalogEntry_en_US] webapp=null path=/select params={q=+(+(name:(**coffee**)+defaultSearch:(**coffee**)+shortDescription:(**coffee**)))+(**mfName_ntk_cs:"Enzi“^1.5**)&fq=catalog_id:10001&fq=storeent_id:(10001)&fq=-catenttype_id_ntk_cs:ItemBean&fq=**mfName_ntk_cs:(++*:*+-"Sharpson")**&start=0&rows=12&timeAllowed=3000&hl.fl=name&hl.fl=shortDescription&hl=true&hl.simple.pre=<strong><span+class=font2>&hl.simple.post=</span></strong>&hl.requireFieldMatch=true&facet.field=ads_f4_ntk_cs&facet.field=ads_f5_ntk_cs&facet.field=ads_f2_ntk_cs&facet.field=ads_f1_ntk_cs&facet.field=ads_f3_ntk_cs&facet.field=mfName_ntk_cs&facet.field=parentCatgroup_id_facet&facet=true&facet.query=price_USD:({*+100}+100)&facet.query=price_USD:({100+200}+200)&facet.query=price_USD:({200+300}+300)&facet.query=price_USD:({300+400}+400)&facet.query=price_USD:({400+500}+500)&facet.query=price_USD:({500+*})&facet.sort=count&facet.mincount=1&facet.limit=10&spellcheck.count=&spellcheck=true&spellcheck.collate=false&debugQuery=false} **hits=22** status=0 QTime=3

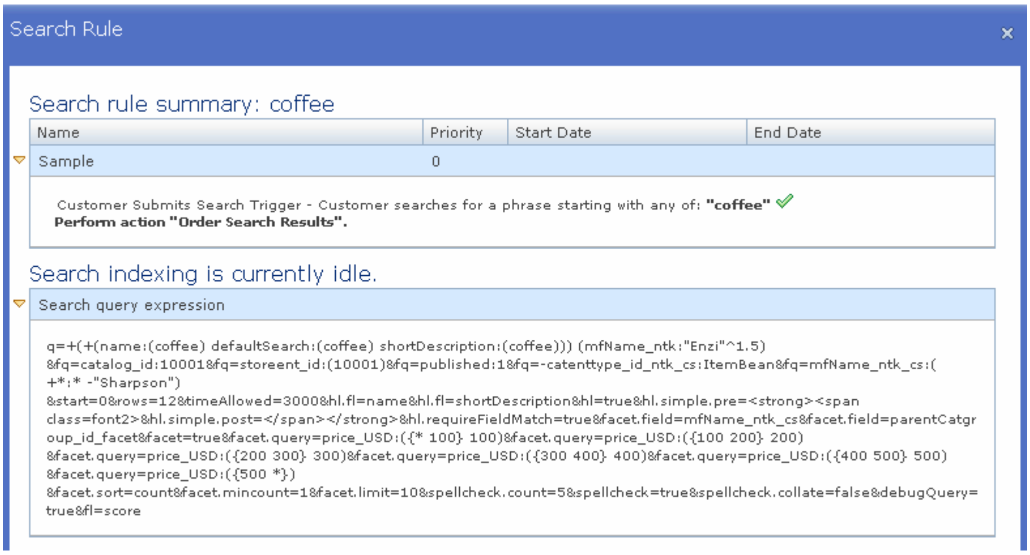
Number of results

Catalog filter

27 Search administration and configuration © 2011 IBM Corporation

This slide shows a highlighted version of a search term from the SystemOut log file. Escape characters normally seen in the log file have been replaced for readability. The search term entered by the shopper is shown in blue. The search profile defines which index fields are searched. In this example, the name and short description fields are searched. The search rule is shown in green. It boosts results from Enzi so they are ranked higher in the result set. This portion of the query is injected by the expression builder for marketing rules. The catalog filter is shown in purple. It excludes results from the Sharpson manufacturer. The number of search result hits is also returned in the log file. This is highlighted in orange at the bottom of the slide.

Search rule and query in store preview mode



Search Rule

Search rule summary: coffee

Name	Priority	Start Date	End Date
Sample	0		

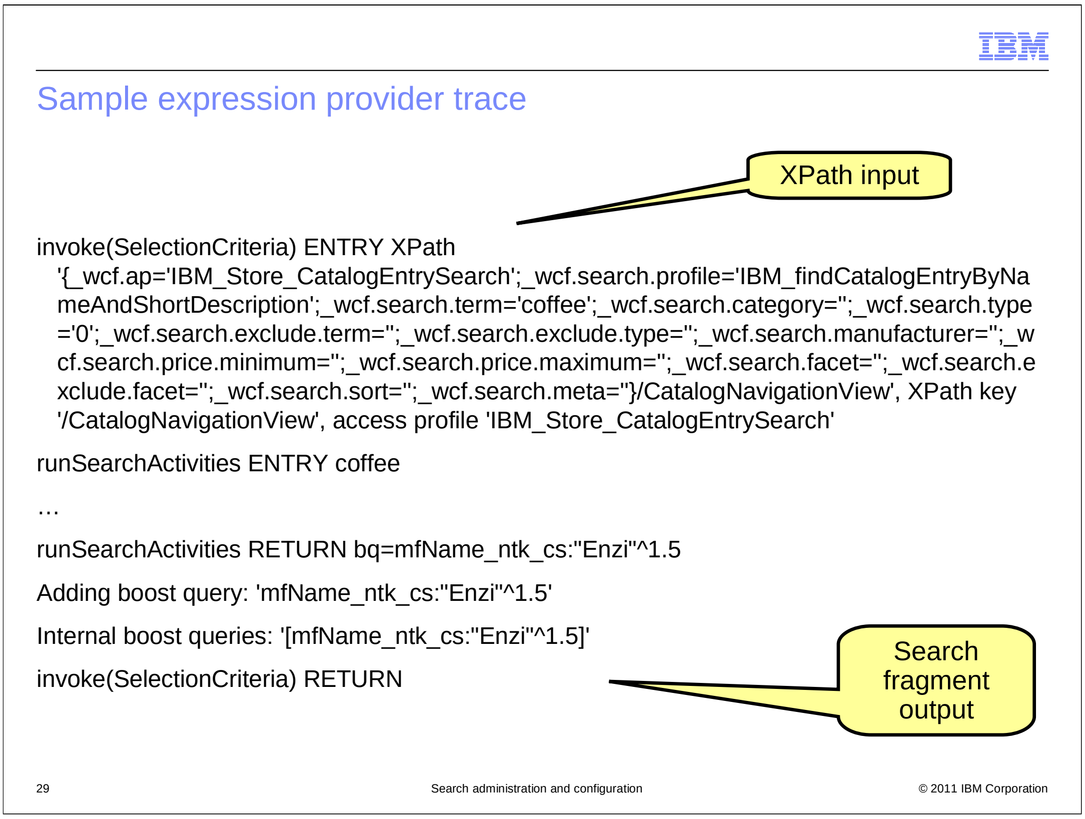
Customer Submits Search Trigger - Customer searches for a phrase starting with any of: **"coffee"** ✓
Perform action "Order Search Results".

Search indexing is currently idle.

Search query expression

```
q=+(+(name:(coffee) defaultSearch:(coffee) shortDescription:(coffee))) (mfName_ntk:"Enzi"^1.5)
&fq=catalog_id:10001&fq=storeent_id:(10001)&fq=published:1&fq=-catenttype_id_ntk_cs:ItemBean&fq=mfName_ntk_cs:(
+*:* -"Sharpson")
&start=0&rows=12&timeAllowed=3000&hl.fl=name&hl.fl=shortDescription&hl=true&hl.simple.pre=<strong><span
class=font2>&hl.simple.post=</span></strong>&hl.requireFieldMatch=true&facet.field=mfName_ntk_cs&facet.field=parentCatgr
oup_id_facet&facet=true&facet.query=price_USD:({* 100} 100)&facet.query=price_USD:({100 200} 200)
&facet.query=price_USD:({200 300} 300)&facet.query=price_USD:({300 400} 400)&facet.query=price_USD:({400 500} 500)
&facet.query=price_USD:({500 *})
&facet.sort=count&facet.mincount=1&facet.limit=10&spellcheck.count=5&spellcheck=true&spellcheck.collate=false&debugQuery=
true&fl=score
```

The same query is shown here in the search rule viewer available in store preview mode. In addition to the query, you can also see exactly which search rules are applied.



Sample expression provider trace

XPath input

```
invoke(SelectionCriteria) ENTRY XPath
  '{_wcf.ap='IBM_Store_CatalogEntrySearch';_wcf.search.profile='IBM_findCatalogEntryByNa
  meAndShortDescription';_wcf.search.term='coffee';_wcf.search.category='';_wcf.search.type
  ='0';_wcf.search.exclude.term='';_wcf.search.exclude.type='';_wcf.search.manufacturer='';_w
  cf.search.price.minimum='';_wcf.search.price.maximum='';_wcf.search.facet='';_wcf.search.e
  xclude.facet='';_wcf.search.sort='';_wcf.search.meta=''}/CatalogNavigationView', XPath key
  '/CatalogNavigationView', access profile 'IBM_Store_CatalogEntrySearch'

runSearchActivities ENTRY coffee

…

runSearchActivities RETURN bq=mfName_ntk_cs:"Enzi"^1.5

Adding boost query: 'mfName_ntk_cs:"Enzi"^1.5'

Internal boost queries: '[mfName_ntk_cs:"Enzi"^1.5]'

invoke(SelectionCriteria) RETURN
```

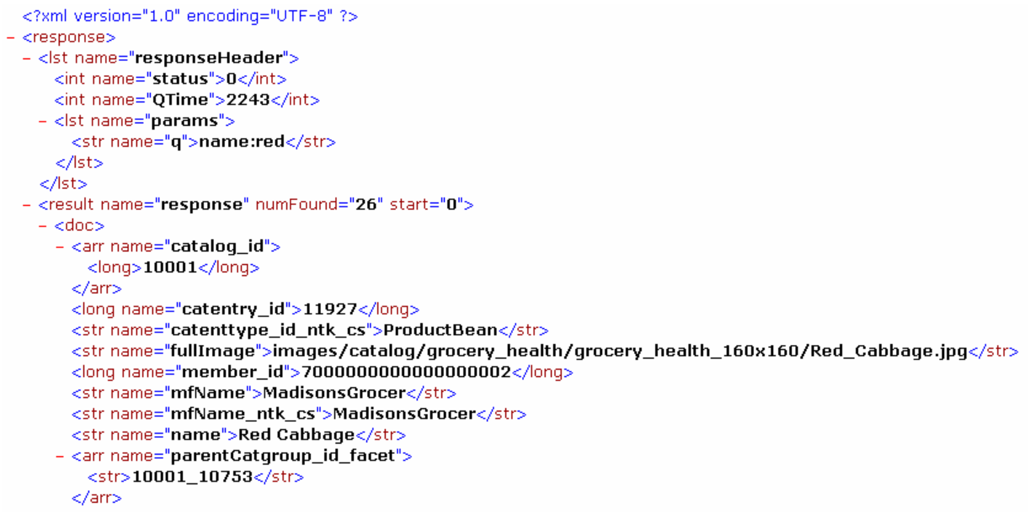
Search fragment output

29 Search administration and configuration © 2011 IBM Corporation

On this slide, you see the input and output for the SolrSearchBasedMerchandisingExpressionProvider. This data can be viewed by enabling the catalog trace component.

Run a query in a browser window

http://localhost/solr/MC_10001_CatalogEntry_en_US/select?q=name:red

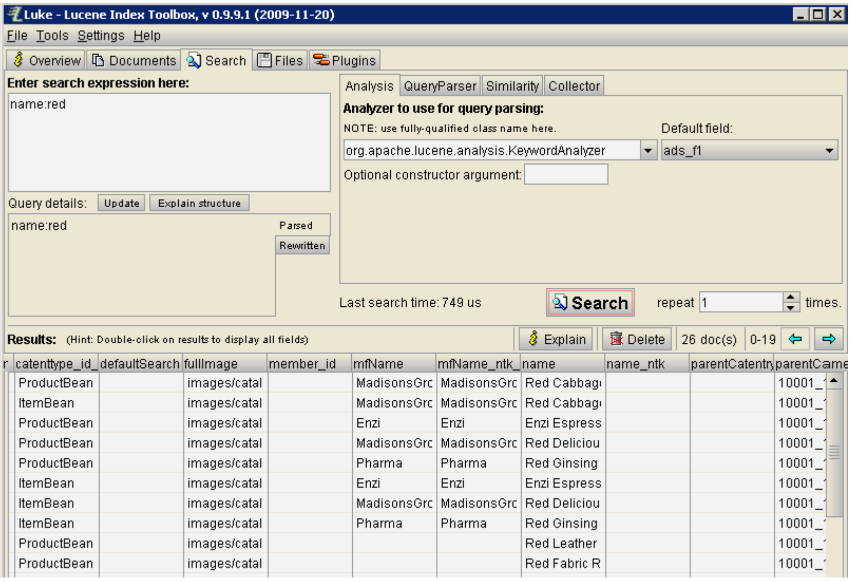


```
  <?xml version="1.0" encoding="UTF-8" ?>
- <response>
  - <lst name="responseHeader">
      <int name="status">0</int>
      <int name="QTime">2243</int>
    - <lst name="params">
        <str name="q">name:red</str>
      </lst>
    </lst>
  - <result name="response" numFound="26" start="0">
    - <doc>
      - <arr name="catalog_id">
          <long>10001</long>
        </arr>
        <long name="catentry_id">11927</long>
        <str name="catenttype_id_ntk_cs">ProductBean</str>
        <str name="fullImage">images/catalog/grocery_health/grocery_health_160x160/Red_Cabbage.jpg</str>
        <long name="member_id">7000000000000000002</long>
        <str name="mfName">MadisonsGrocer</str>
        <str name="mfName_ntk_cs">MadisonsGrocer</str>
        <str name="name">Red Cabbage</str>
      - <arr name="parentCatgroup_id_facet">
          <str>10001_10753</str>
        </arr>
```

30 Search administration and configuration © 2011 IBM Corporation

Running a query directly against Solr can help to determine whether the problem is in the index itself or in the result processing or display. You can enter a Solr query into a browser window and the Solr response object is displayed as the result.

Run a query in Luke*



Luke - Lucene Index Toolbox, v 0.9.9.1 (2009-11-20)
File Tools Settings Help
Overview Documents Search Files Plugins
Analysis QueryParser Similarity Collector
Update
Explain structure
Parsed
Rewritten
Search
Explain
Delete

Enter search expression here:

name:red

Query details:

name:red

Analyzer to use for query parsing:

NOTE: use fully-qualified class name here.

org.apache.lucene.analysis.KeywordAnalyzer

Default field: ads_f1

Optional constructor argument:

Last search time: 749 us

repeat 1 times.

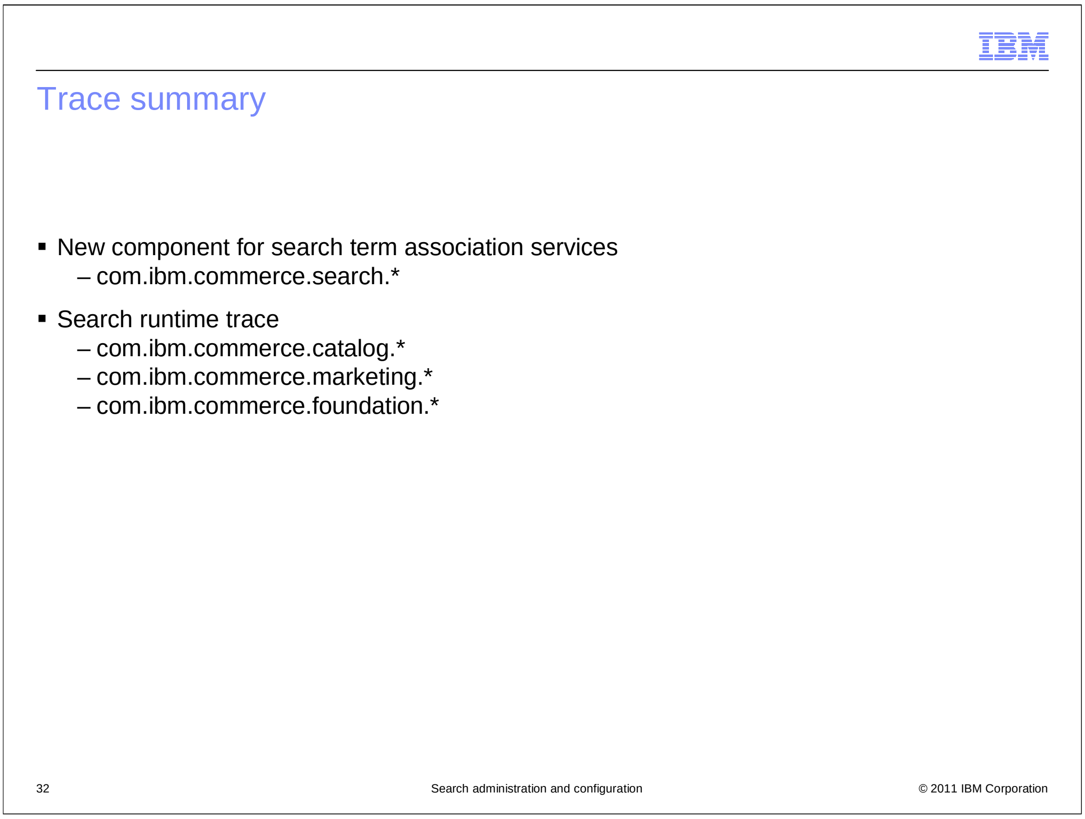
Results: (Hint: Double-click on results to display all fields)

26 doc(s) 0-19

r	catenttype_id_	defaultSearch	fullImage	member_id	mfName	mfName_ntk_	name	name_ntk	parentCatentry	parentCame
	ProductBean		images/catal		MadisonsGrc	MadisonsGrc	Red Cabbag			10001_
	ItemBean		images/catal		MadisonsGrc	MadisonsGrc	Red Cabbag			10001_
	ProductBean		images/catal		Enzi	Enzi	Enzi Espress			10001_
	ProductBean		images/catal		MadisonsGrc	MadisonsGrc	Red Deliciou			10001_
	ProductBean		images/catal		Pharma	Pharma	Red Ginsing			10001_
	ItemBean		images/catal		Enzi	Enzi	Enzi Espress			10001_
	ItemBean		images/catal		MadisonsGrc	MadisonsGrc	Red Deliciou			10001_
	ItemBean		images/catal		Pharma	Pharma	Red Ginsing			10001_
	ProductBean		images/catal				Red Leather			10001_
	ProductBean		images/catal				Red Fabric R			10001_

*Luke is not shipped with Feature Pack 2

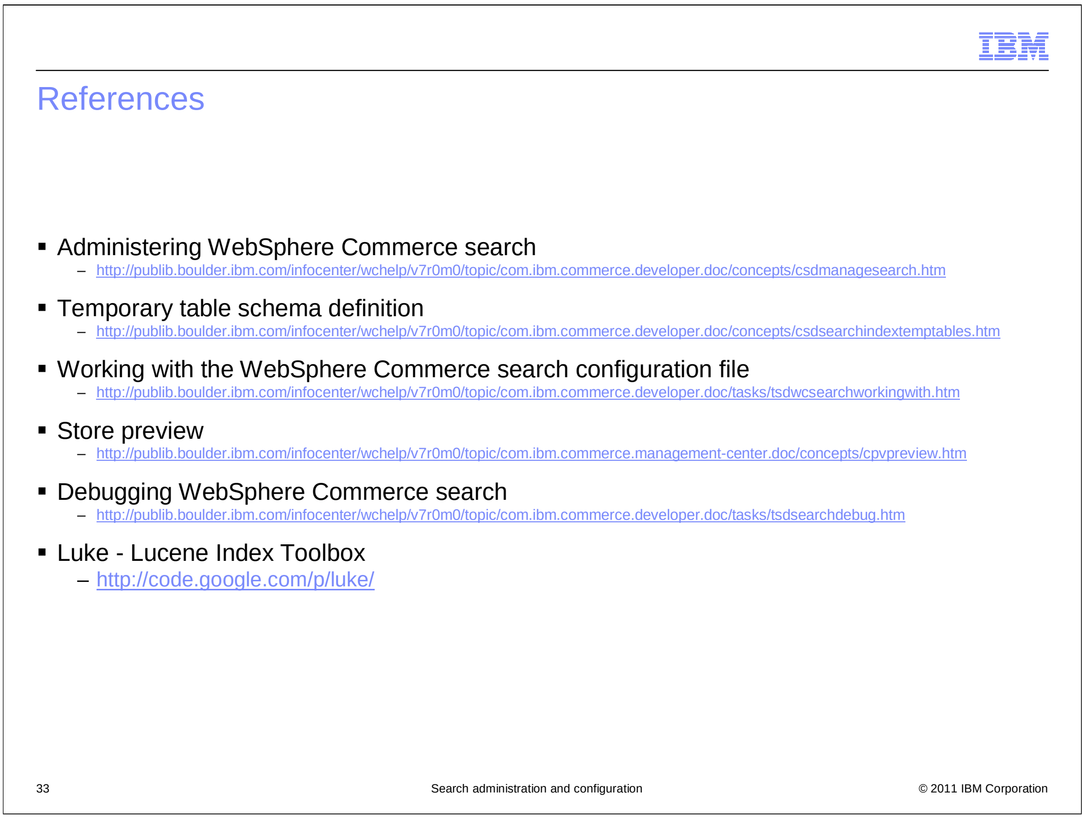
To get a deeper look into the index, you can install the Lucene index tool called Luke. Luke is not shipped as part of WebSphere Commerce feature pack 2 and must be separately downloaded and installed. This screen capture shows the results of the query on the previous slide as seen in Luke.



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.*

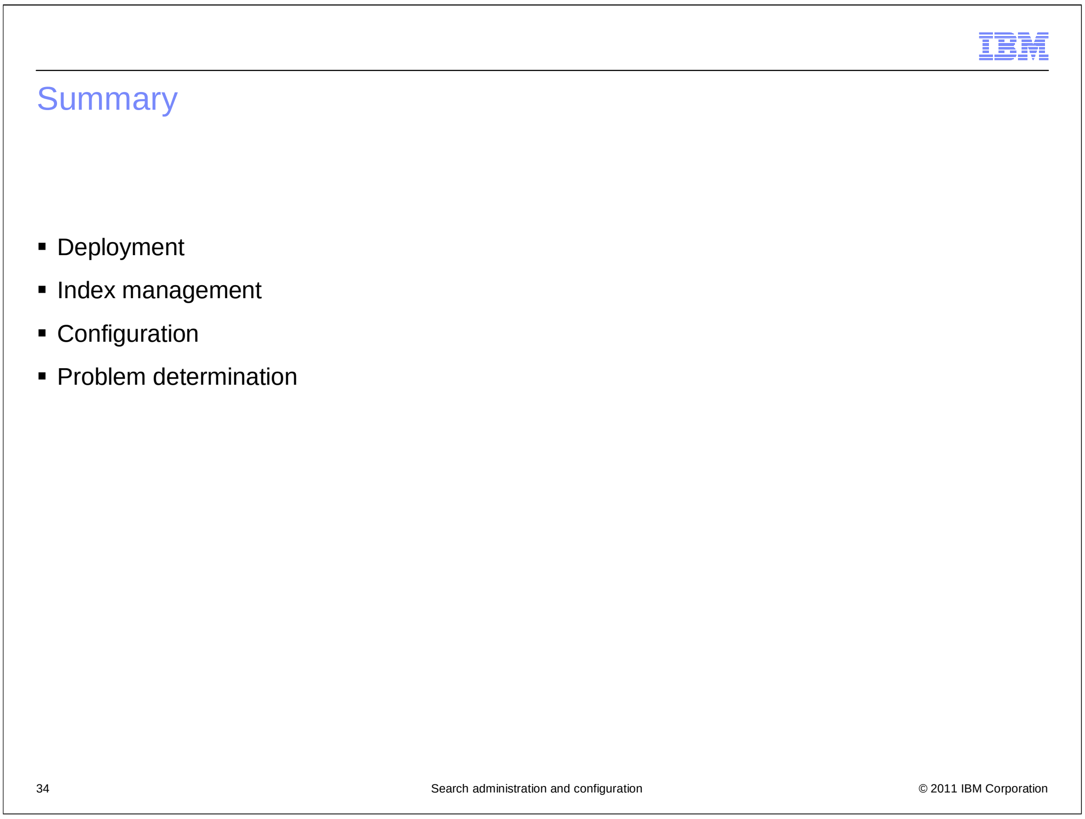
The WebSphere Commerce search solution adds one new trace component for the services dealing with search term association management. The majority of the search runtime trace statements can be viewed by enabling the existing catalog, marketing and foundation trace components.



References

- Administering WebSphere Commerce search
  - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/concepts/csdmanagesearch.htm
- Temporary table schema definition
  - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/concepts/csdsearchindextemptables.htm
- Working with the WebSphere Commerce search configuration file
  - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/tasks/tsdwcsearchworkingwith.htm
- Store preview
  - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.management-center.doc/concepts/cpvpreview.htm
- Debugging WebSphere Commerce search
  - http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/topic/com.ibm.commerce.developer.doc/tasks/tsdsearchdebug.htm
- Luke - Lucene Index Toolbox
  - http://code.google.com/p/luke/

This slide contains some useful references.

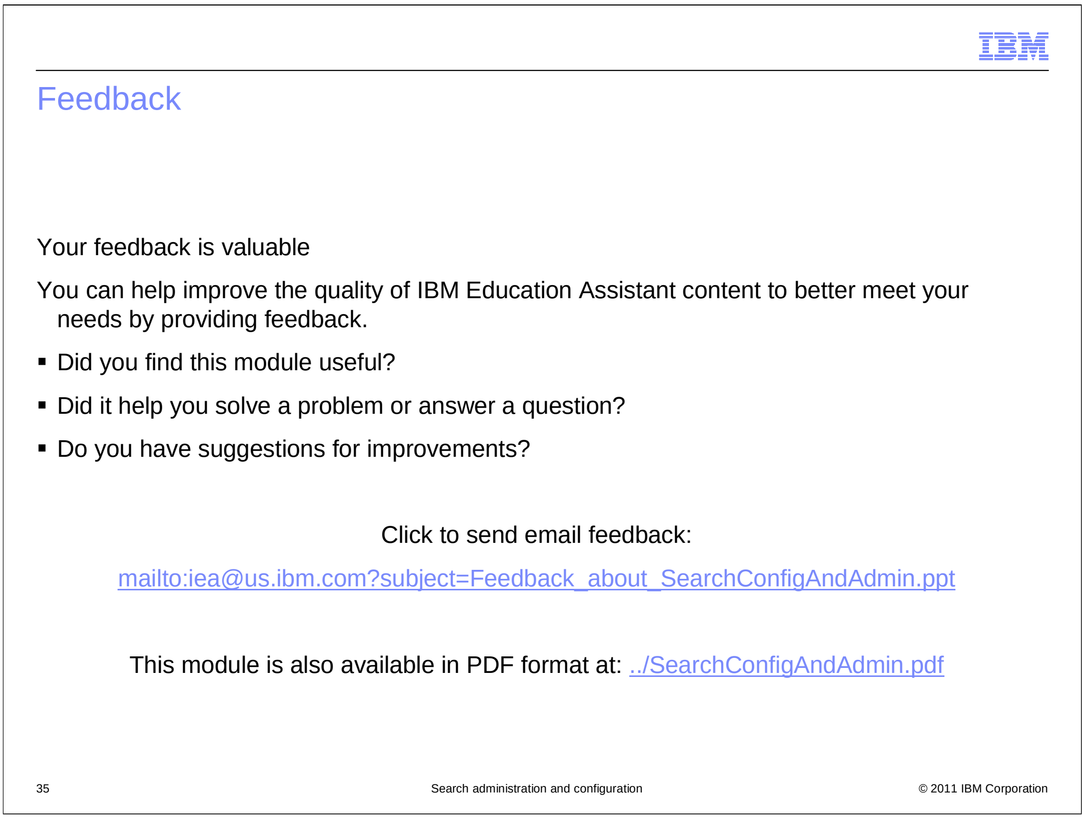


Summary

- Deployment
- Index management
- Configuration
- Problem determination

34 Search administration and configuration

The presentation began with an overview of the steps for deploying WebSphere Commerce Search. Following deployment was an overview of index management. The next section introduced the configuration options and the presentation concluded with some problem determination tips.



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?

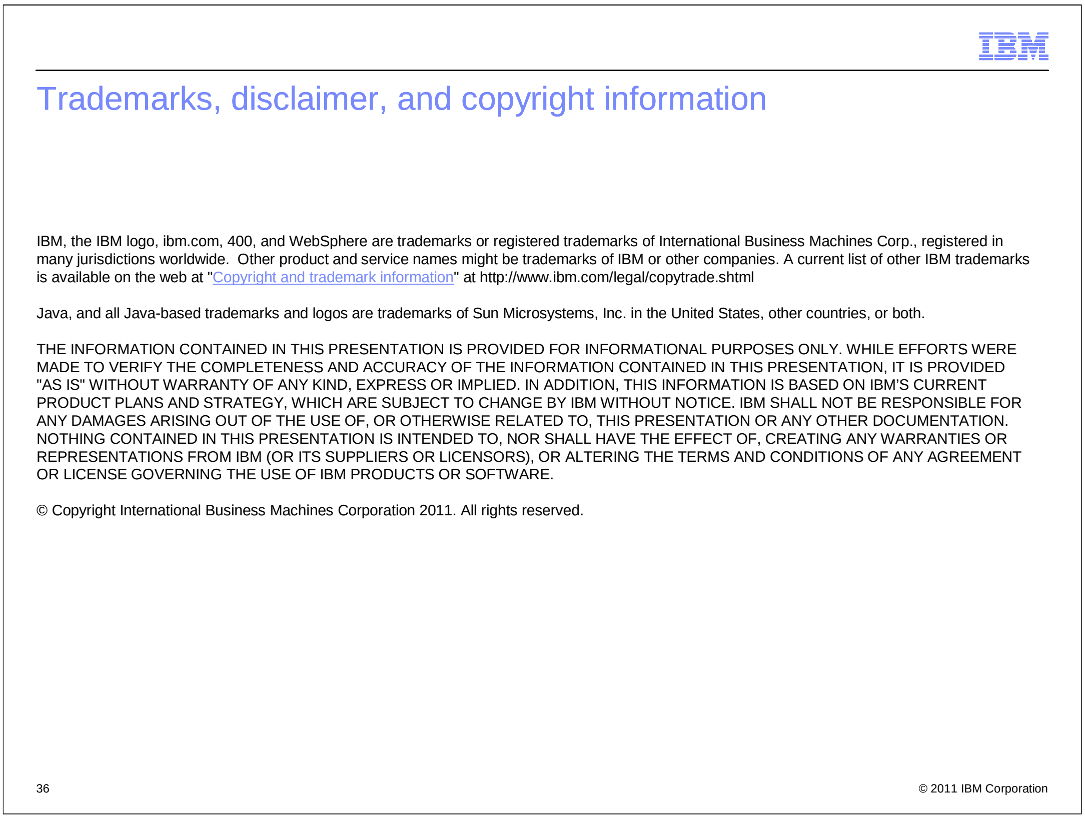
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_SearchConfigAndAdmin.ppt

This module is also available in PDF format at: ../SearchConfigAndAdmin.pdf

35 Search administration and configuration © 2011 IBM Corporation

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, 400, 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" at http://www.ibm.com/legal/copytrade.shtml

Java, 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.