
Welcome to the WebSphere Commerce Feature Pack 2 WebSphere Commerce Integration with Sitemap Overview presentation.

## Agenda

- Search engine crawling
- Sitemaps
- Configuring WebSphere Commerce sitemap feature
- Creating a sitemap
- Failure problem determination



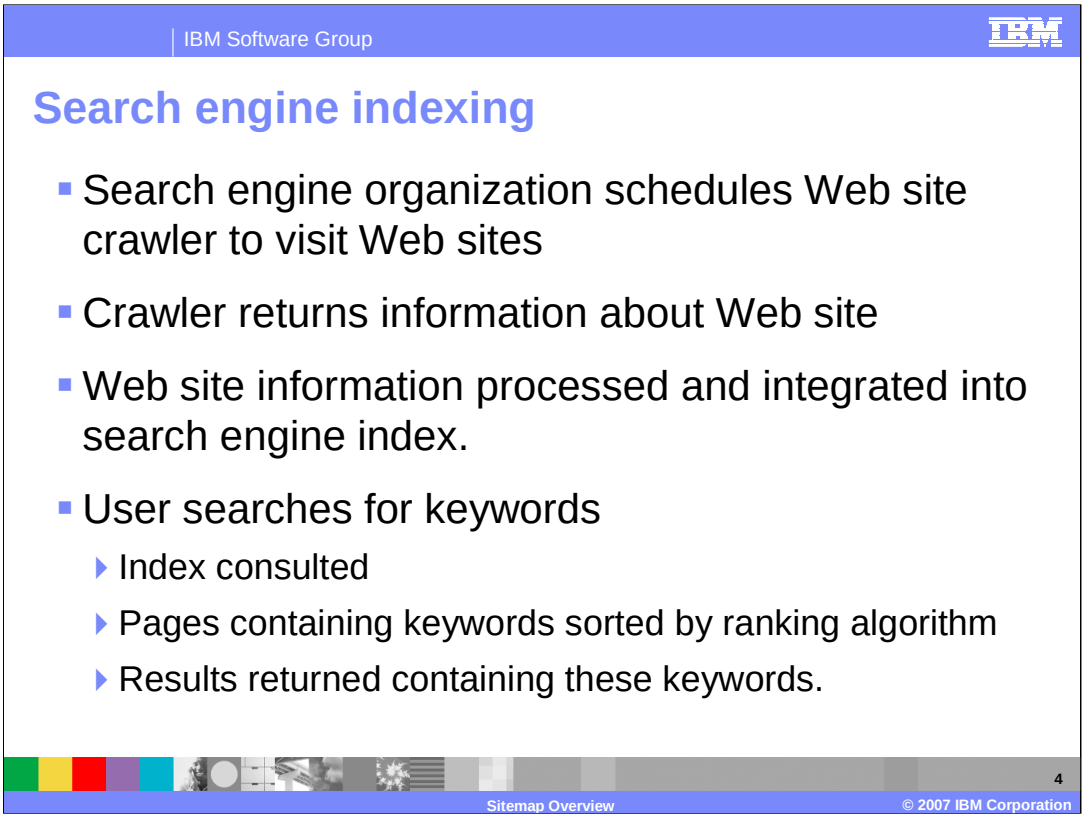

The agenda for this presentation is to discuss search engine crawling, sitemaps, WebSphere Commerce Sitemap feature overview, configuring the sitemap feature, creating a sitemap, and problem determination for operational failures.

## Section

***Search engine crawling***




This section discusses the basics of how search engines crawl the Web.




Web search engines index the contents of the World Wide Web by using a device called a Web site crawler. The crawler returns information about the contents of a Web site. This information is processed and integrated into a search engine database or search engine index. When a user searches for a string of keywords, the search engine uses its index to return search results about which Web pages contain these keywords.

## Crawler operation

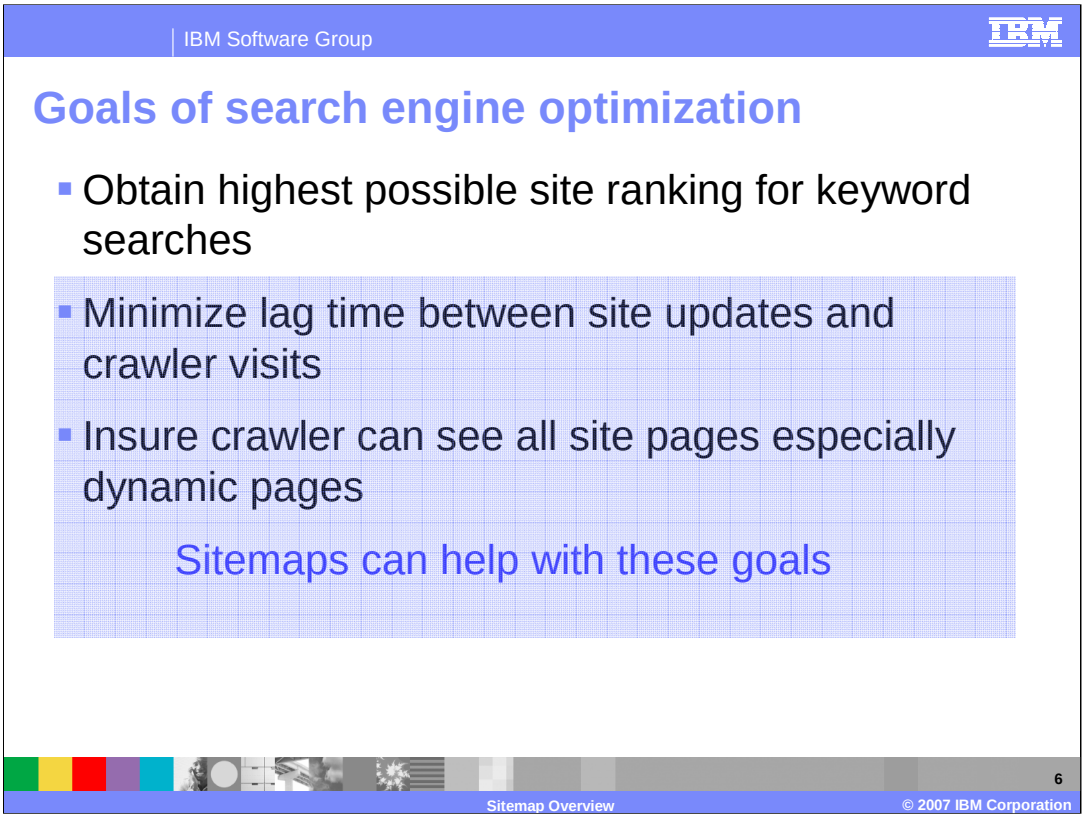
- Web crawler learns of pages from links from
  - Self-referencing links
  - External links
- Search engine ranking ranks sites and pages from
  - Keywords on the page
  - Which and how many sites link to your page
- Links and keywords are learned by parsing each page
- Lag time between site updates and crawler visits




A Web crawler learns of the existence of a Web page by some other page reference. The page is referenced either from a different site, or self referenced from the same site. A self reference is one which does not contain a full domain name URL.

The search engine has an algorithm to rank pages for a particular keyword according to which keywords are on a page, and how many times other sites refer to an external page. More external references means a higher page ranking.

Links and keywords are learned only by the crawler reading and parsing every character on the page. Crawling is time and computer resource intensive. Consequently, there is a lag time between the time a site is updated and when the crawler next visits. Without some help, the crawler has no knowledge of when updates occur, and visits when it thinks it should rather than when the Web site administrator thinks it should.

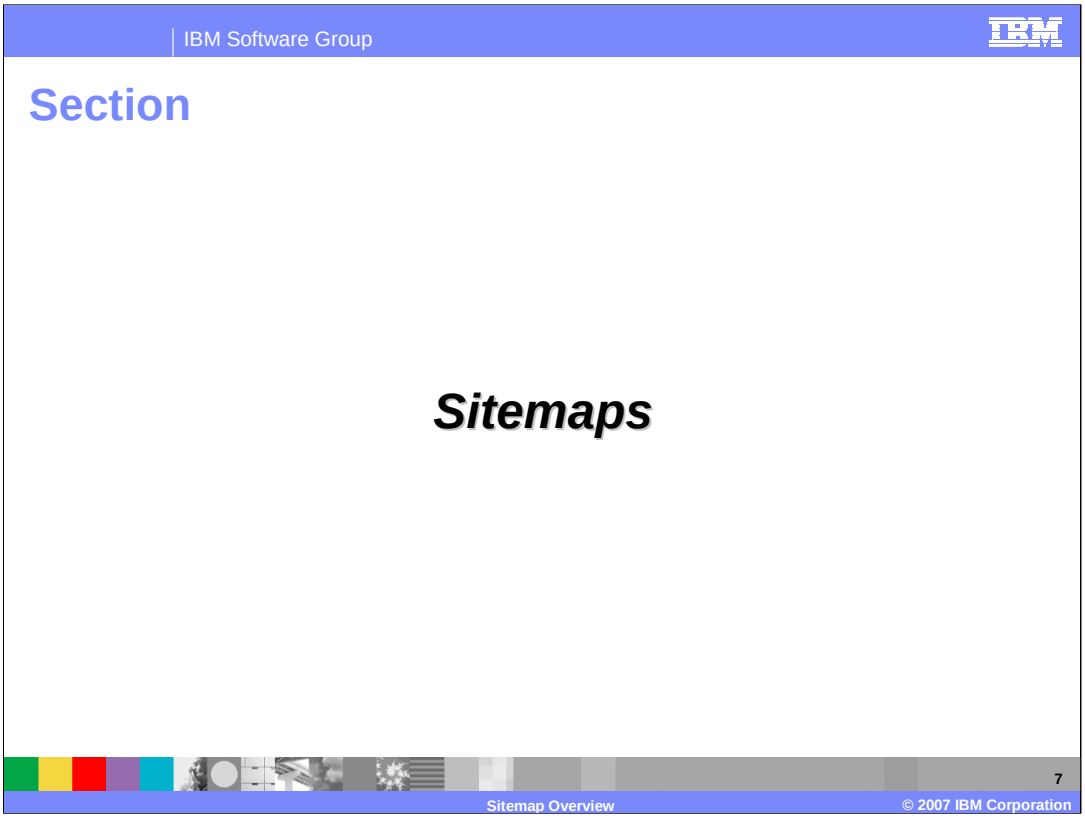



Web site administrators have several goals in order to get the best results for their site from search engines. The administrator can take several steps to optimize search engine results to insure these goals are met. Since search engines return results in page rank sort order, the most important goal is to achieve the highest keyword ranking.

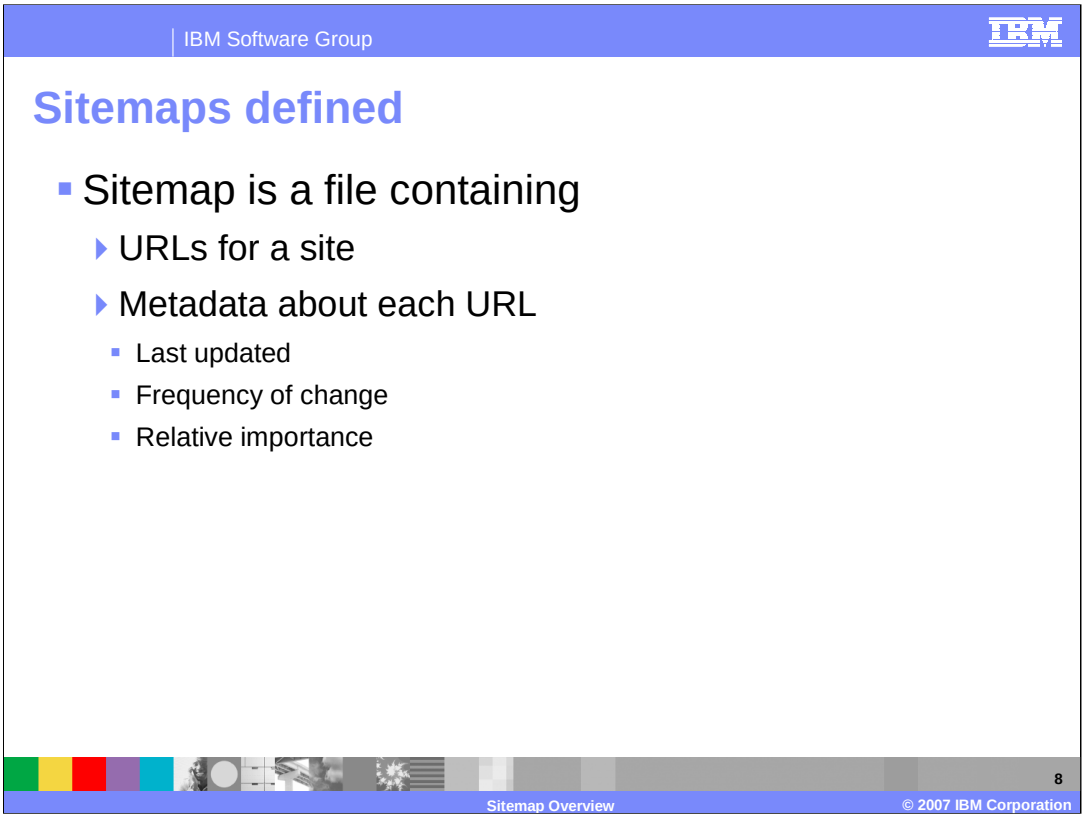
The second most important goal is to have the crawler visit as soon as a page in the site is changed.

The third most important goal is to insure the crawler can see all pages especially those which are dynamically generated pages that are not referenced by any other page.

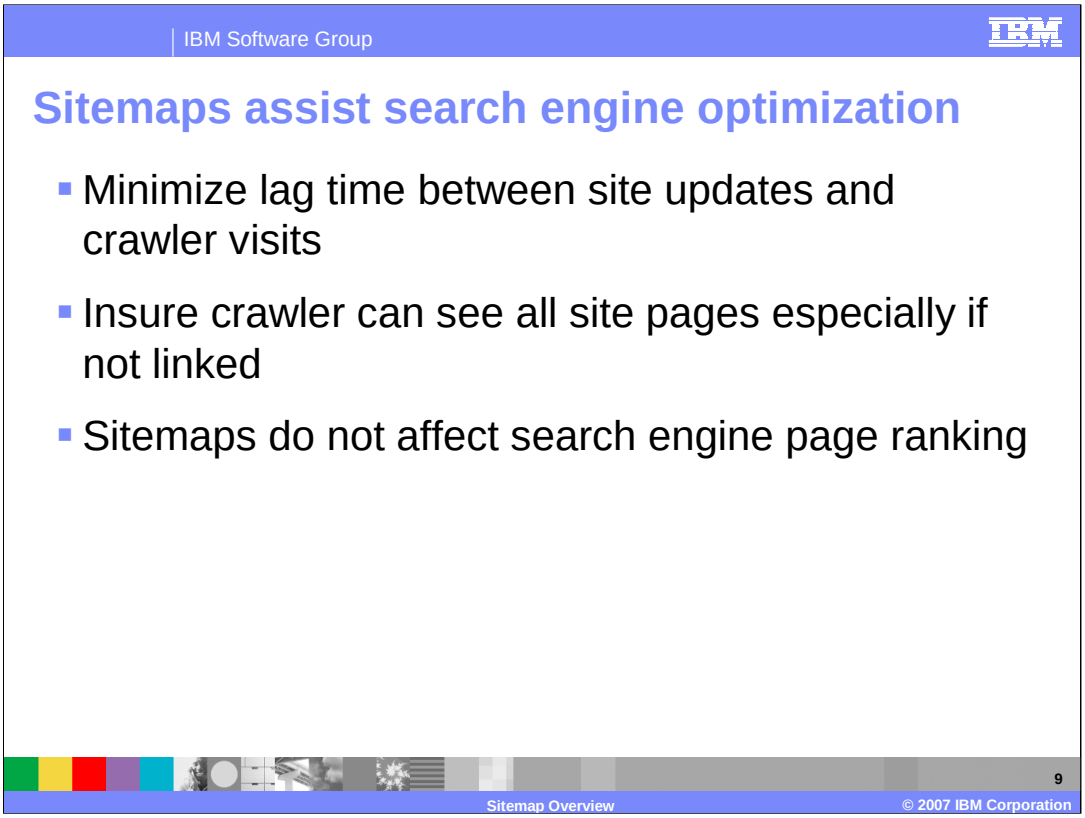
A mechanism called sitemaps can help achieve the second and third goals.




This section discusses the basics of sitemaps.

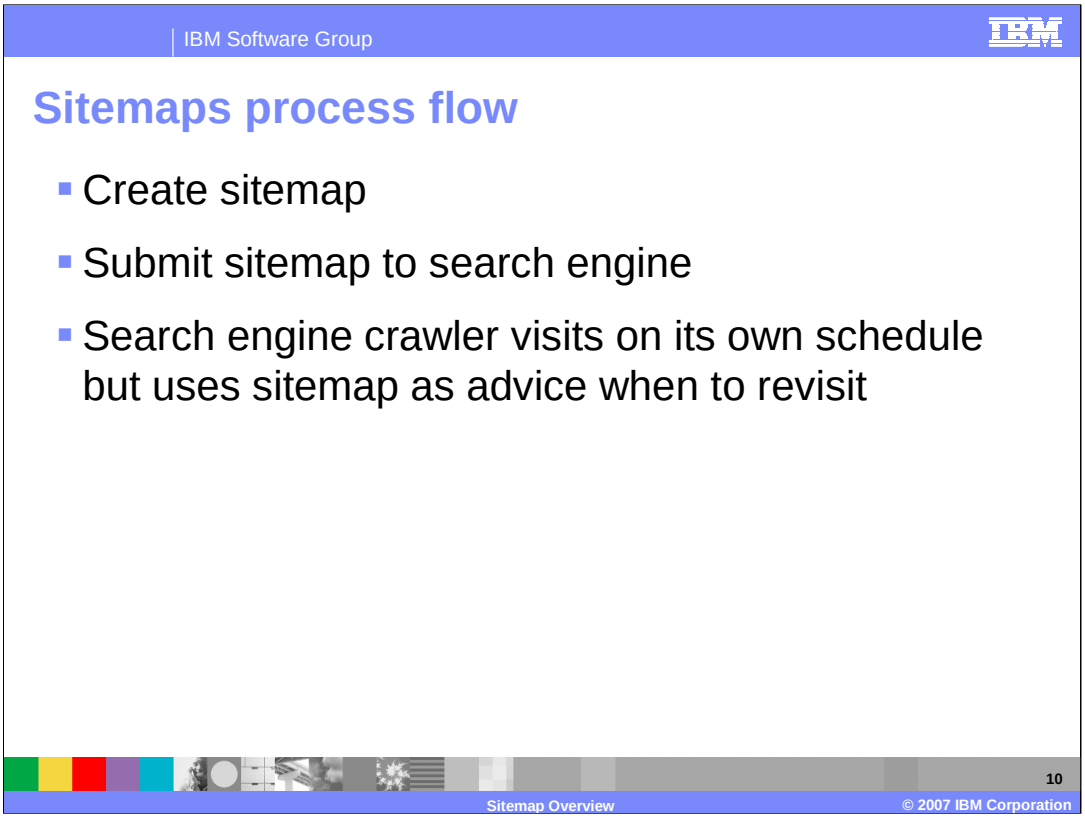



A sitemap is a file that contains URLs for a Web site, and metadata describing each URL. Some examples of metadata are when the page was last updated, how often it is updated, and the relative importance of the page relative to all the other pages on the same site.




Search engine optimization goals are enhanced by use of a sitemap. The Web crawler uses a sitemap as advice about when to revisit a site, and about the existence of dynamically generated pages that have no references to them. An example of a unreferenced page would be a page listing a product that was found by searching a site database of stocked products.

Sitemaps do not affect the rank of a page for a particular keyword.




A Web site administrator first creates a sitemap, and then submits the sitemap to the search engine. The crawler revisits the site on a revised schedule that uses the sitemap information as advice as to when and how often to visit.

## Sitemap protocol

- Sitemap.org open protocol body with support from Google, Microsoft, Yahoo, IBM
- XML file with simple tags
- Each file limited to
  - 10 MB
  - 50,000 URLs
  - UTF-8 encoding
- Each file may be compressed with gzip
- Multiple sitemap files require a sitemap index
  - Required to keep individual files under size or URL limit




The sitemap protocol is defined by sitemap.org, which is an open body of several internet companies. Google, Microsoft, Yahoo and IBM all have endorsed sitemap.org. The protocol specification is documented at sitemap.org. The protocol defines an XML file with a simple set of tags. Each file must be less than 10 Mbytes in size and contain less than 50,000 URLs. The file must be in UTF-8 encoding. Each file may be compressed using gzip compression. In order to keep each file under the size limit, the sitemap can contain multiple files. The case of multiple files requires a sitemap index file which lists each of the individual sitemap files.

# Sitemap file examples

- Simplest sitemap with only required tags

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
  <loc>http://www.chipsgalore.com/wcs/webapp/potatoe</loc>
</url>
</urlset>
```

- More complex example with optional tags

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
 <loc>http://www.chipsgalore.com/wcs/webapp/potatoe</loc>
 <lastmod>2007-04-01</lastmod>
 <changefreq>weekly</changefreq>
 <priority>0.5</priority>
</url>
</urlset>
```




The first sitemap file example shows the minimum URL description listing only the required tags. The second example shows a more complex example showing the use of optional tags.

# Sitemap index example

- Simple sitemap index example

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<sitemap>
  <loc>http://www.chipsgalore.com/sitemap1.xml.gz</loc>
</sitemap>
<sitemap>
 <loc>http://www.chipsgalore.com/sitemap2.xml.gz</loc>
</sitemap>
</sitemapindex>
```



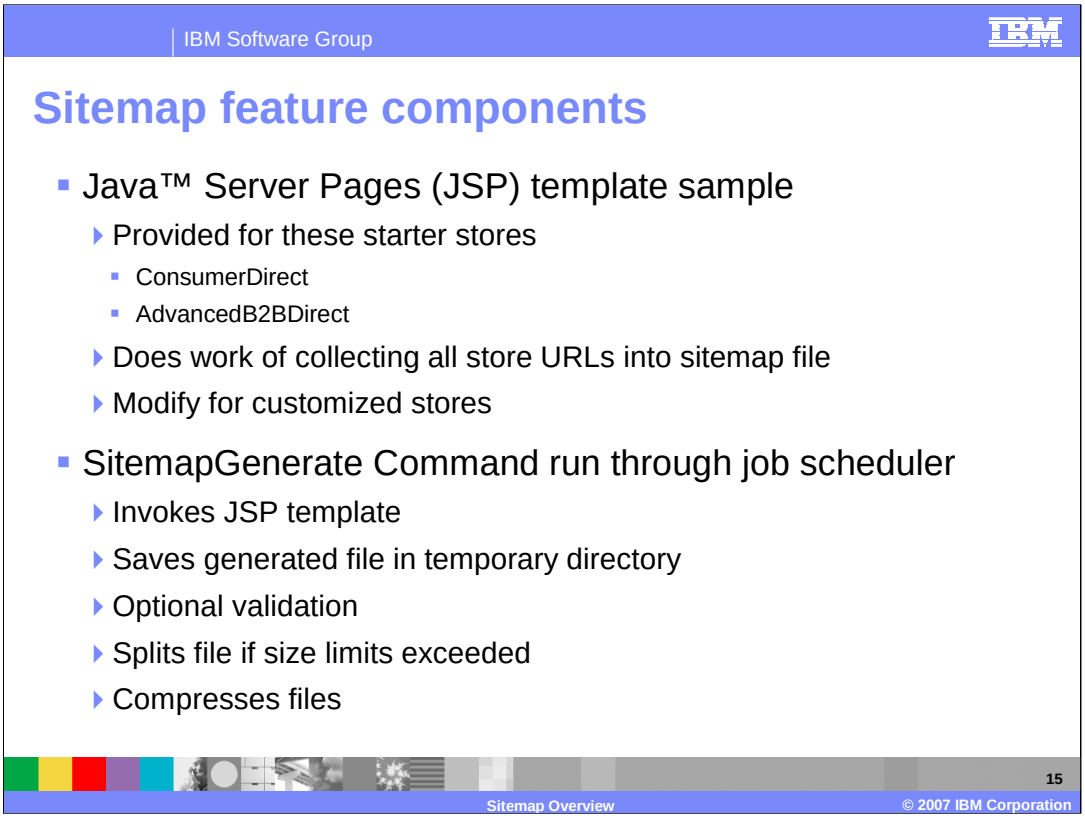

This is an example of a simple sitemap index.

## Section

***Configuring sitemap feature***




This section discusses how to configure the sitemap feature.

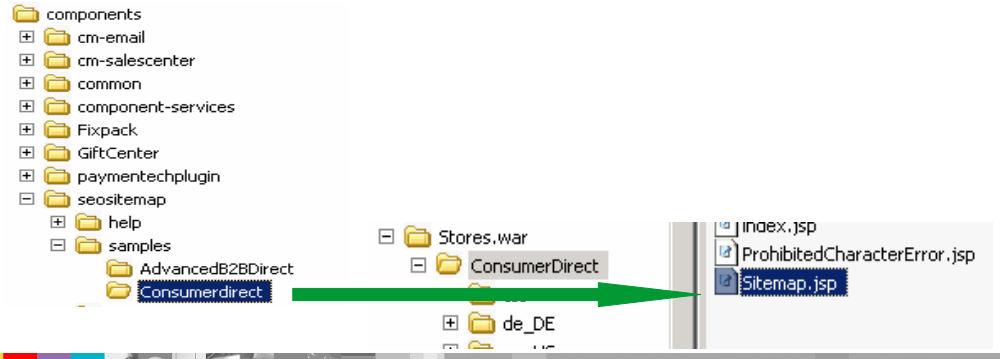



The sitemap feature in WebSphere Commerce feature pack 2 provides a sample JSP template file for the Consumer Direct and Advanced B2B Direct starter stores. This JSP file does the actual searching of the site for page URLs.

The sample JSP file is invoked from the SitemapGenerate command. This command is run through the job scheduler. The SitemapGenerate command takes the resultant JSP file output and saves it, does optional validation, and performs the mechanics of splitting the sitemap file into multiple files if the size limits are exceeded.

# Configuring sitemaps feature

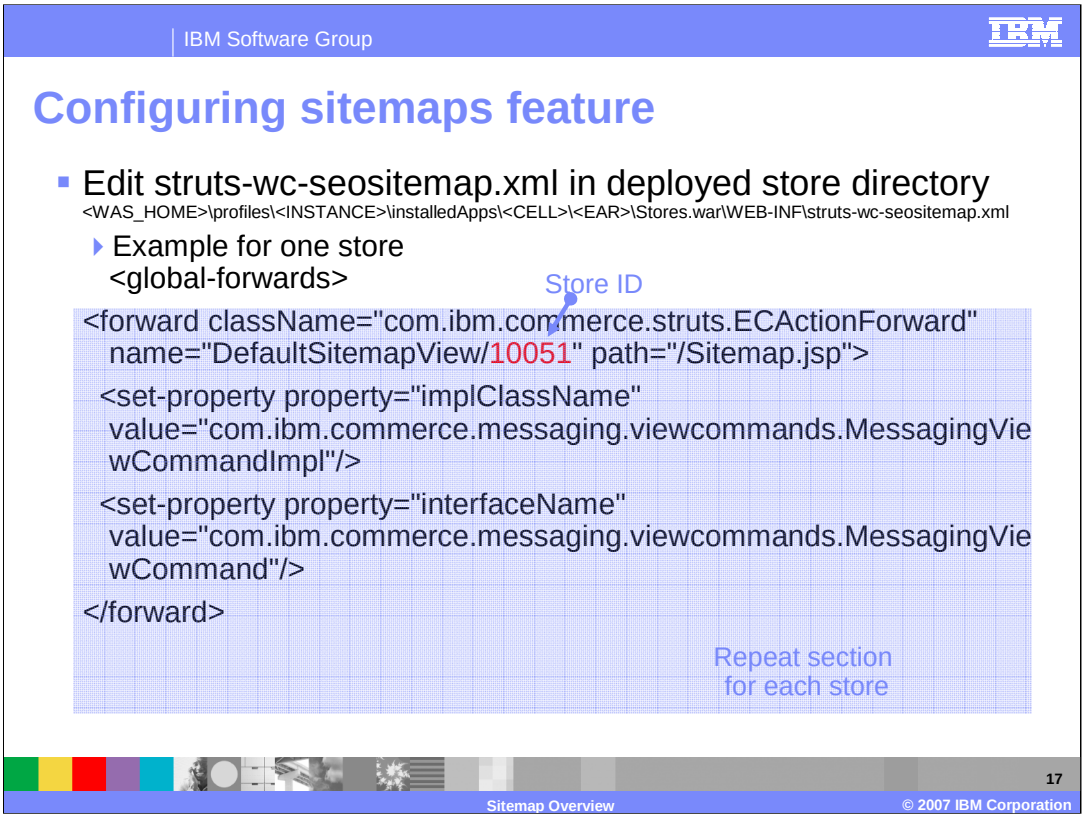
- Enable feature for
  - Server enable for each instance
    config_ant -buildfile <WCS_HOME>/components/common/xml/enableFeature.xml -DinstanceName=<instance> -DfeatureName=seositemap -DdbUserPassword=<dbUserPassword>
  - Toolkit enable
    enableFeature -DfeatureName=seositemap
- Copy sample JSP template to deployed store directory







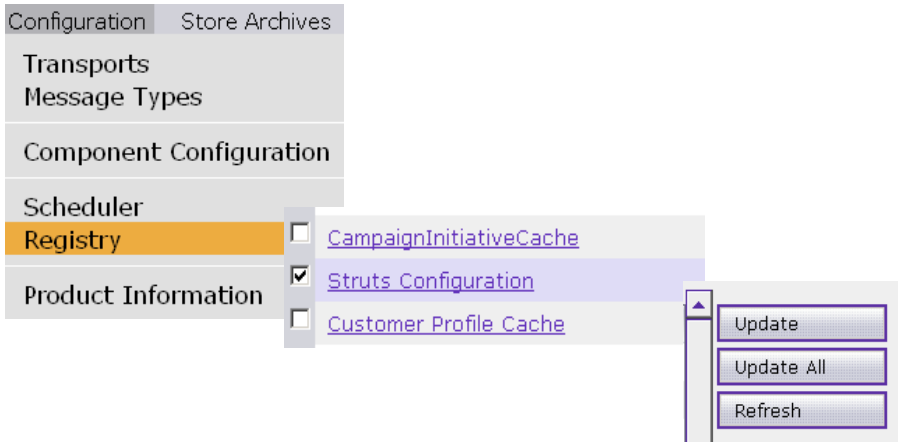
The first step to configuring the sitemap feature is to enable the feature. For the server environment, use the config_ant command. An example is shown in the slide. To enable the sitemap feature for the developer environment, use the enableFeature command. An example is shown in the slide. The next step is to copy the sample JSP template file to the deployed store directory. The sample is found under the components/seositemap directory.




The next step is to edit the struts configuration file in the deployed store directory. The full path to the file is found in the slide. Edit the file with a text editor and find the line which has the field name=DefaultSitemapView. Add the storeId after the field as indicated in the slide. If you have multiple stores to configure, you must copy and repeat the highlighted section for each store.

# Configuring sitemaps feature

- Update struts configuration with WebSphere Commerce administration console







After editing the file, update the struts configuration in the WebSphere Commerce administration console. Select the menu Configuration > Registry, and then select Struts configuration. Select Update and wait a few seconds for completion. You can use the Refresh button to refresh the screen to see when the operation is complete.

This last step completes the configuration step.

## Section

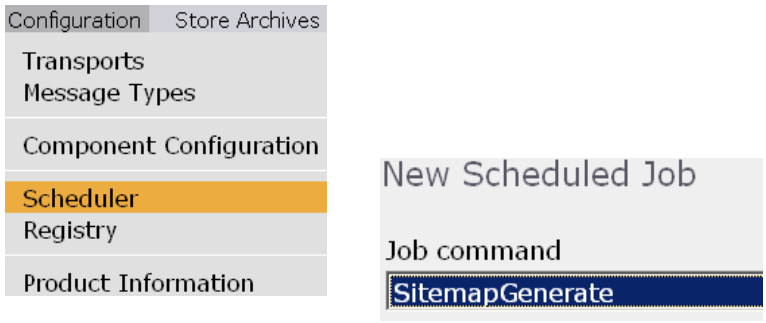
***Creating sitemap files***




This section discusses creating the sitemap files.

# Run creation command

- Run job scheduler in WebSphere Commerce administration console
- Select scheduler job SitemapGenerate
  - Specify necessary parameters
  - Specify appropriate schedule







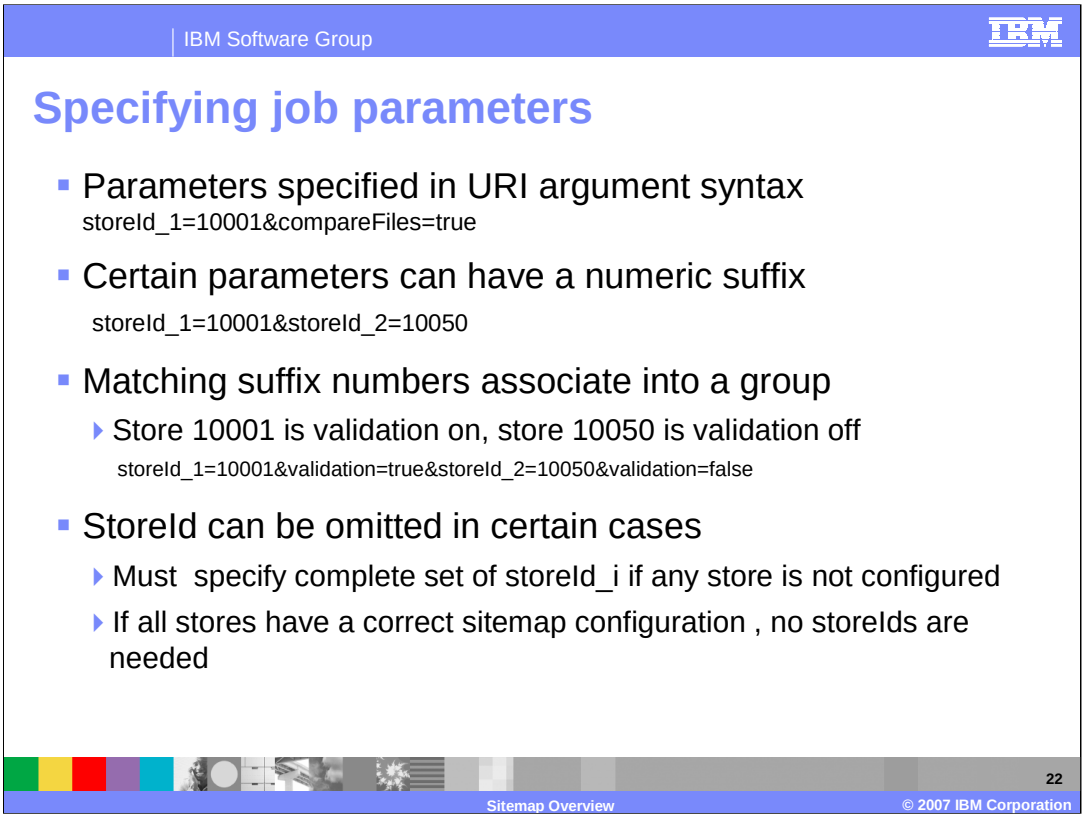
To generate a sitemap for one or more stores, you run a job in the job scheduler. In the WebSphere Commerce administration console, select the menu Configuration > Scheduler. Select New job and specify the SitemapGenerate command. You then specify any necessary parameters and the necessary time schedule information. Information on necessary parameters is discussed in subsequent slides.

# Sitemap parameter summary

| | | |
|---|---|---|
| validation_i | true \| false | Validate against sitemap schema |
| storeId_i | Decimal number | Id found in STORE table |
| catalogId_i | Decimal number | Id found in STORE table |
| siteMapView_i | string | Used if customization specifies different struts view name |
| storeType | string | Type found in STORE table |
| hostname | Valid DNS name | Use for staging site |
| compareFiles | true \| false | Compare to previous sitemaps |
| maxSize | Decimal number | File size limit in Mbytes |
| maxUrlNumber | Decimal number | URL count limit per file |




This table summarizes all the valid parameters for the SitemapGenerate job. Subsequent slides give examples of how to use these parameters.




Job parameters are specified as if they are using URI syntax. That is, each parameter must be separated by an ampersand character.

The first example shows a storeId and the compareFiles parameters separated by an ampersand.

Some of the parameters have the suffix _i which indicates they can have a numeric suffix attached to a sequence number for multiple parameters of the same type. The example in the slide shows how 2 storeId parameters would be specified.

When you use numeric suffixes, the suffix implies a grouping of like numbered parameters with that same numbered store. The example in the slide shows how to group the validation parameter with different stores.

The storeId parameter can be omitted in special cases, but in general, the storeId must be specified. If all the stores on the server are configured for a sitemap, and you want the job to generate for all the stores, then you can omit the storeId as a selector. In other words, if all your stores have their storeID entered into the struts configuration file, then you can omit storeId as a parameter. You then get sitemaps for all your published stores when you run the SitemapGenerate job.

## Sitemap output

- Sitemap output files placed in deployed store in Stores.war directory
- File names have storeId embedded
  - Example single file
    sitemap_10001.xml.gz
  - Example multiple files
    sitemap_10001_1.xml.gz
    sitemap_10001_2.xml.gz
    sitemap_10001_index.xml.gz




After a successful SitemapGenerate job, the resultant sitemap files are placed in the deployed stores directory directly under Stores.war.

The resultant file names have the storeID embedded in the name in order to distinguish which files are associated with which stores.

The slide shows examples of the names that are created.

## Submit sitemap to search engine

- Optional information for general awareness
- Submission is search engine specific
  - Google search content submission
    http://www.google.com/intl/en/submit_content.html
  - Yahoo search content submission
    https://siteexplorer.search.yahoo.com/submit




After the sitemap is generated, the next step is for you to submit the sitemap files to the appropriate search engine. This sitemap submission process is something that is outside the scope of the WebSphere Commerce product and therefore is not something that IBM can help you with. The particular search engine organization would help you with the submission process.

The slide provides some URLs to see examples of what is involved in submitting your sitemap to a search engine.

## Section

***Problem determination***




This section discusses problem determination methods.

## Problem determination

- Logs to gather
  - Runtime – SystemOut.log in instance profile <WAS_HOME>/profiles/<instance>/logs/server1/
  - Feature enable logs in instance logs directory <WC_HOME>/instances/<instance>/logs
- Temporary directories for intermediate results

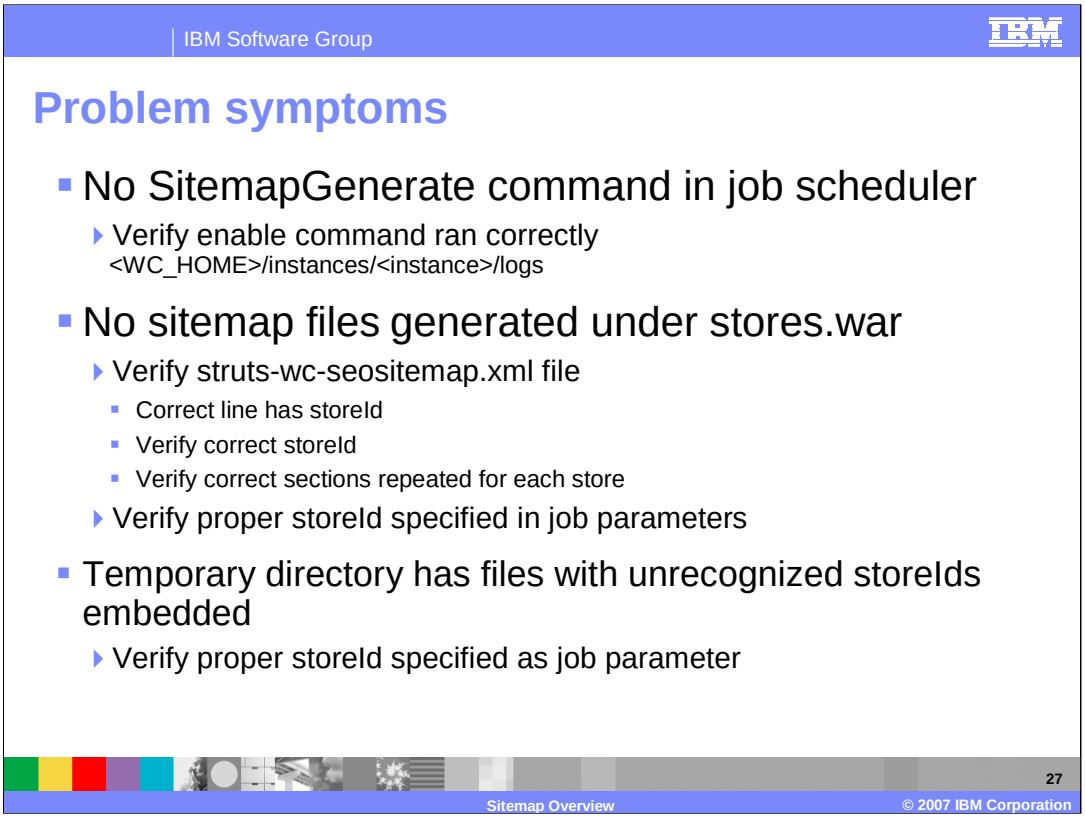
  <WAS_HOME>/profiles/<instance>/temp
- Verify sitemap.jsp files in proper place
- Tracing

  com.ibm.websphere.commerce.seo.*




Problem determination information can be found in the logs directories. The runtime log can be found in the server profile log whose location is given in the slide. The results of the feature enable step are found in the logs in the WebSphere Commerce instances directory. The exact location is given in the slide. You can find intermediate sitemap generation results in the server profile temporary directory whose exact location is given in the slide. One valuable file location is to check that the sitemap.jsp files are in their proper place under the deployed store directory.

A difficult problem can require trace log data. To set the proper trace point information, specify the trace specification as shown in the slide.




These are some problem symptoms and some likely causes you should check.

If the SitemapGenerate command does not show in the job scheduler, the most likely cause is that the enable command was not run. To verify if it was run, check the output logs in the instances logs directory.

If no sitemap files are generated, verify the contents of the struts configuration file. Also verify that the proper storeId was specified as a job parameter.

If the temporary directory has files with unrecognized storeIds embedded in the file names, then the most likely cause is that storeId was not specified as a job parameter.

## Summary

- WebSphere Commerce sitemap integration feature enables site administrators to easily maintain sitemaps for submission to search engines
- The feature implements industry accepted open protocols for creating sitemaps




The summary of the presentation was to show how the sitemap integration feature enables Web site administrators to easily maintain sitemaps for submission to search engines. The feature implements industry open protocols for creating sitemaps.

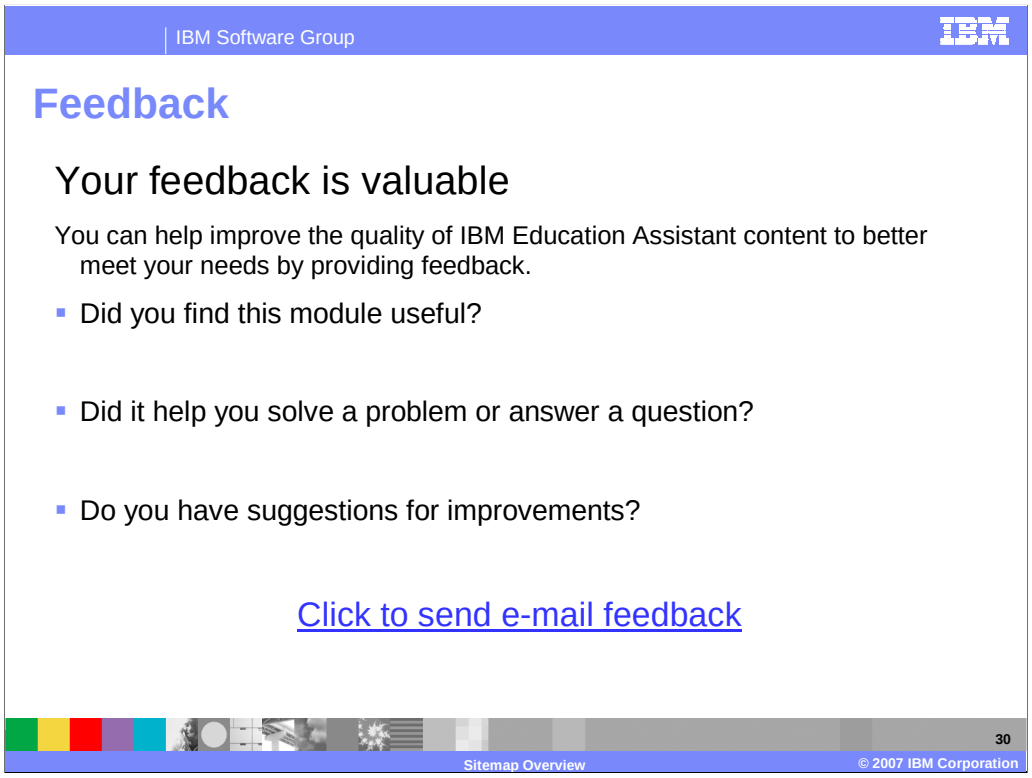
## References

- IBM press release about sitemaps in WebSphere Portal
  http://www-03.ibm.com/press/us/en/pressrelease/21158.wss
- Sitemap protocol
  http://www.sitemaps.org




Here are 2 useful reference articles. A press release about sitemap support in WebSphere Portal can be found at the address in the slide.

The sitemap protocol is defined at the address in the slide.




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

Template Revision: 04/25/2006 11:09 AM

# Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

| | | | | |
|---|---|---|---|---|
| IBM | CICS | IMS | MQSeries | Tivoli |
| IBM(logo) | Cloudscape | Informix | OS/390 | WebSphere |
| e(logo)business | DB2 | iSeries | OS/400 | xSeries |
| AIX | DB2 Universal Database | Lotus | pSeries | zSeries |

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

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.



Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.