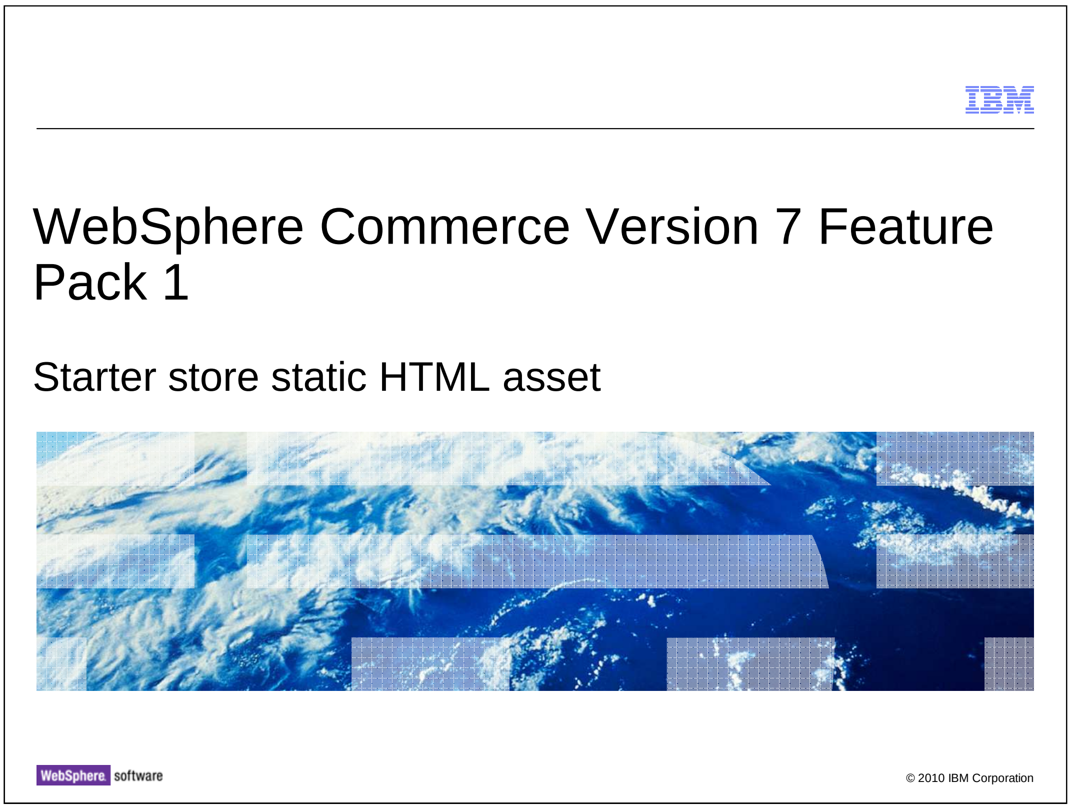


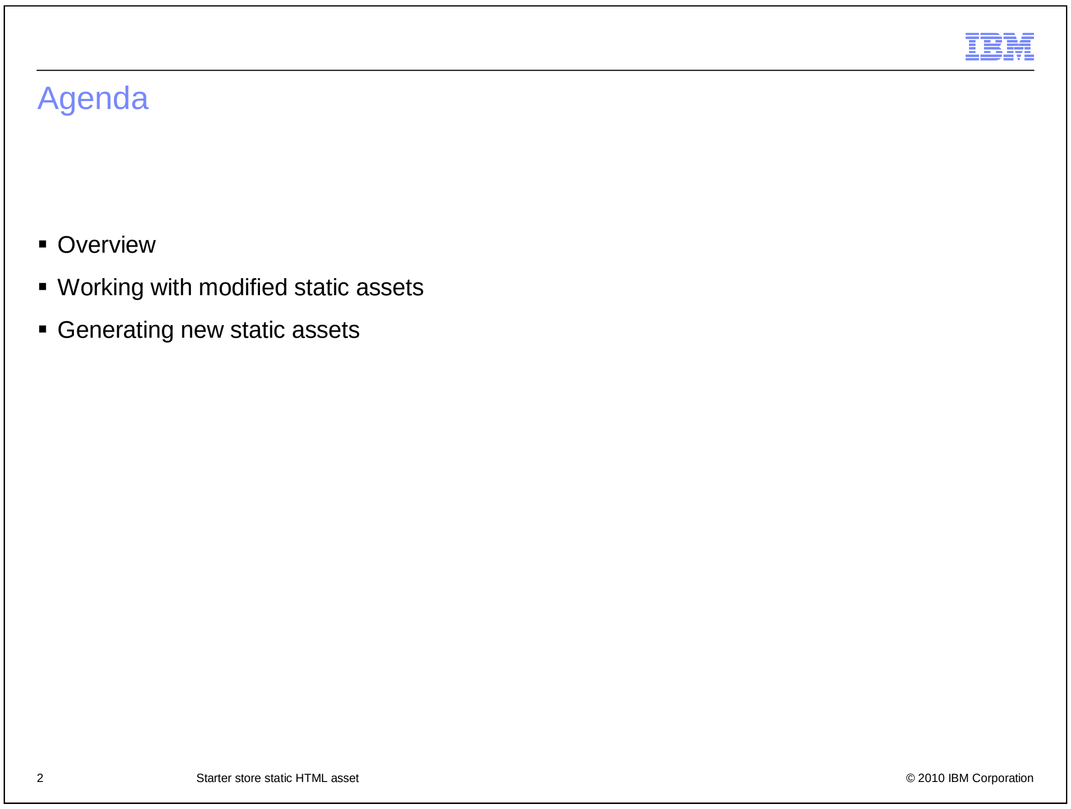
WebSphere Commerce Version 7 Feature Pack 1

Starter store static HTML asset

WebSphere software

© 2010 IBM Corporation

This presentation will go over the components of the starter store static HTML asset and how to use them. It will also cover how to create a set of static HTML assets for your own store or modified starter store.

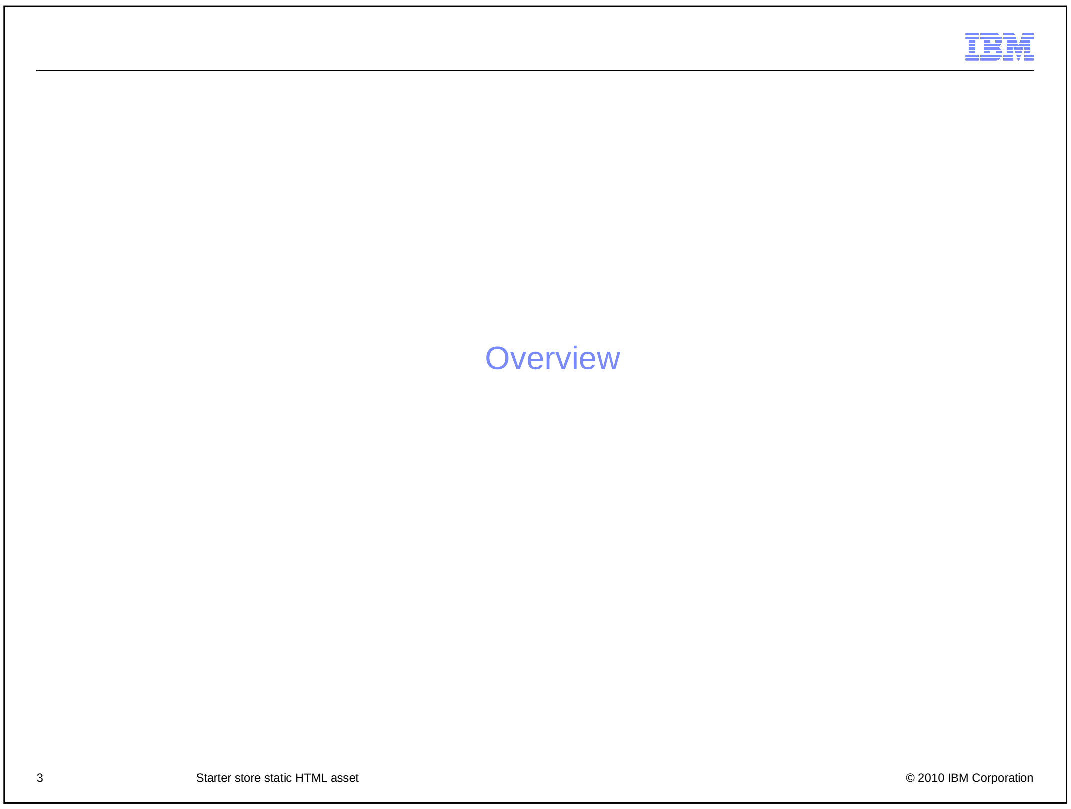


Agenda

- Overview
- Working with modified static assets
- Generating new static assets

© 2010 IBM Corporation

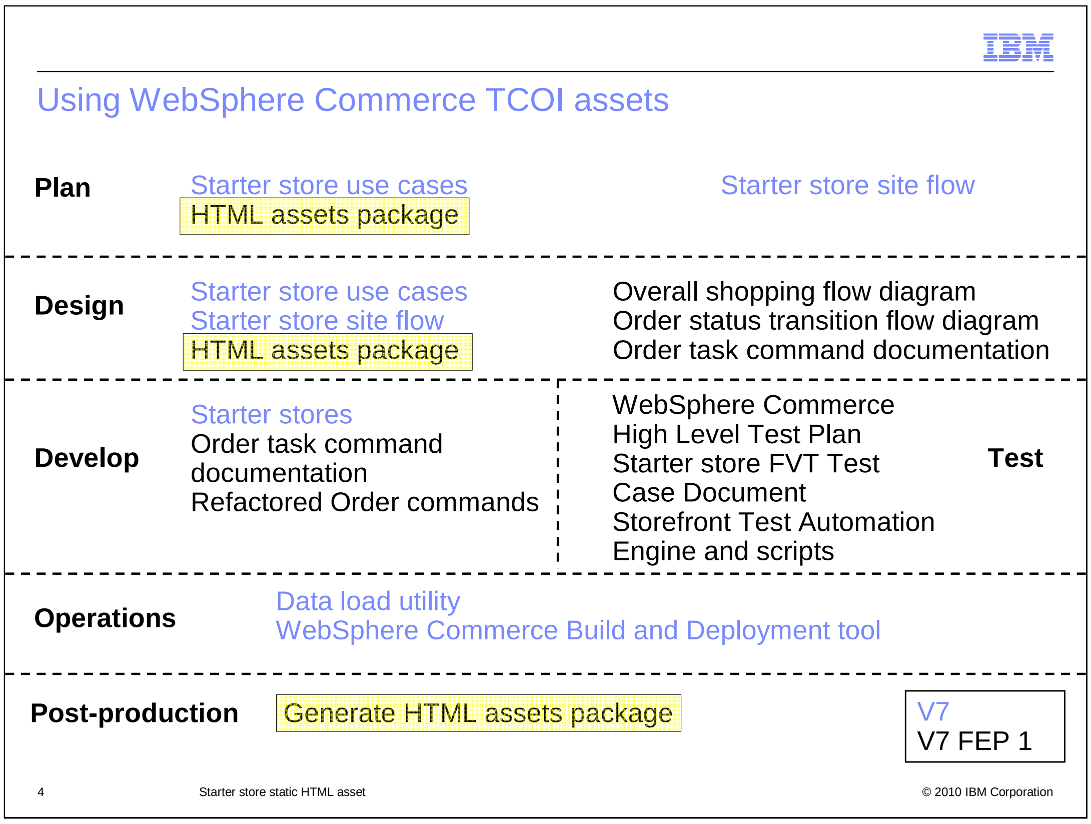
The agenda for this presentation is listed above.



Overview

© 2010 IBM Corporation

The first section of this presentation will go over the static HTML asset.



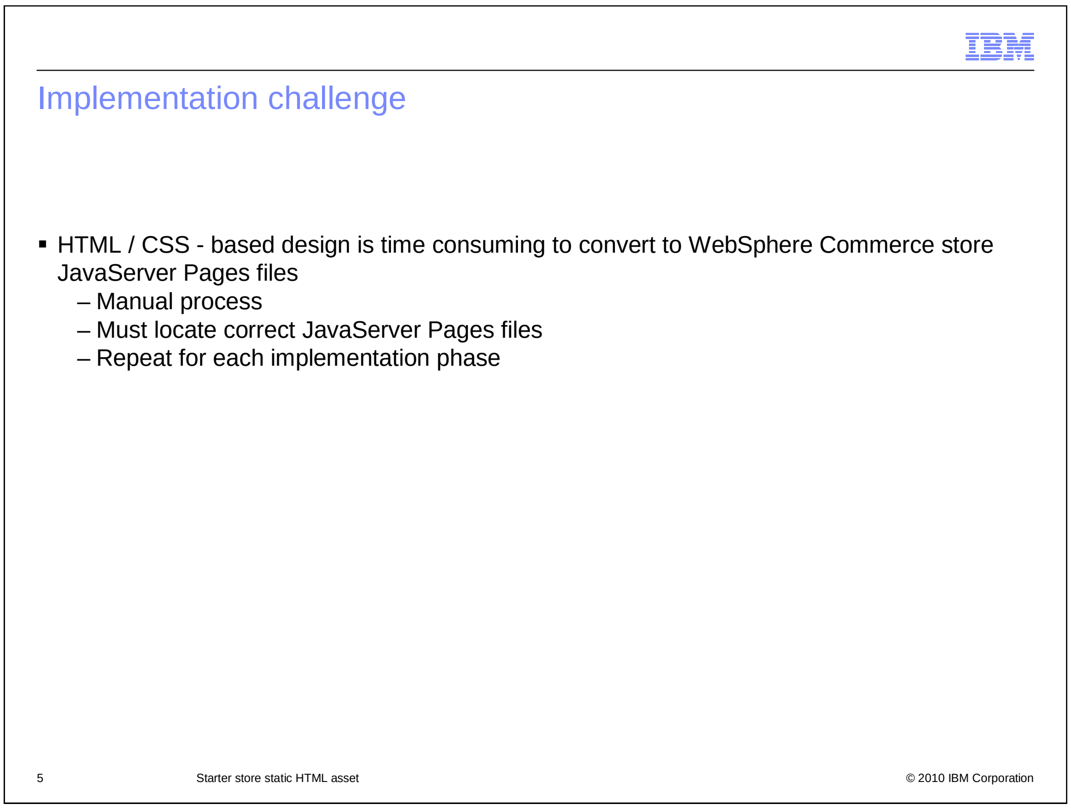
Using WebSphere Commerce TCOI assets

Plan	Starter store use cases HTML assets package	Starter store site flow
Design	Starter store use cases Starter store site flow HTML assets package	Overall shopping flow diagram Order status transition flow diagram Order task command documentation
Develop	Starter stores Order task command documentation Refactored Order commands	WebSphere Commerce High Level Test Plan Starter store FVT Test Case Document Storefront Test Automation Engine and scripts **Test**
Operations	Data load utility WebSphere Commerce Build and Deployment tool	
Post-production	Generate HTML assets package	

V7
V7 FEP 1

4 Starter store static HTML asset © 2010 IBM Corporation

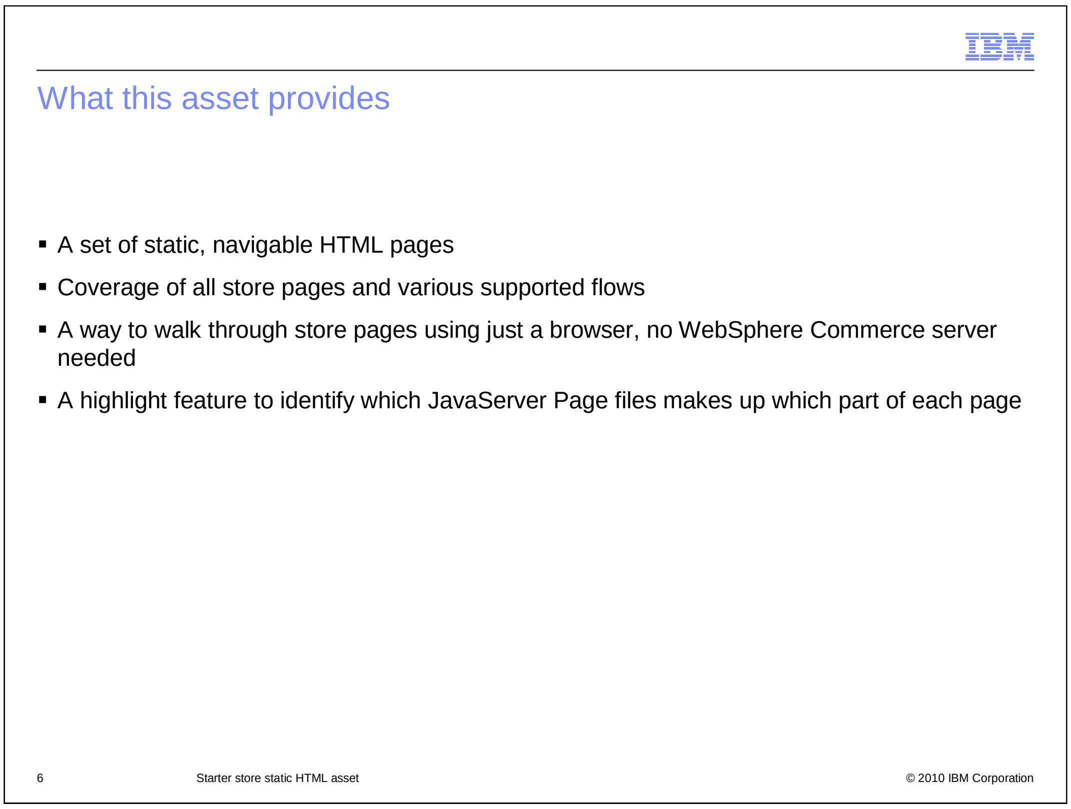
The slide here shows the overall WebSphere Commerce TCOI assets. The mapping here shows which assets to use during each phase of production. The HTML assets package is helpful to Reduce Total Cost of Implementation. The HTML assets package should be used in both the Plan and Design phases and then new assets should be generated in the post-production phase.



Implementation challenge

- HTML / CSS - based design is time consuming to convert to WebSphere Commerce store JavaServer Pages files
  - Manual process
  - Must locate correct JavaServer Pages files
  - Repeat for each implementation phase

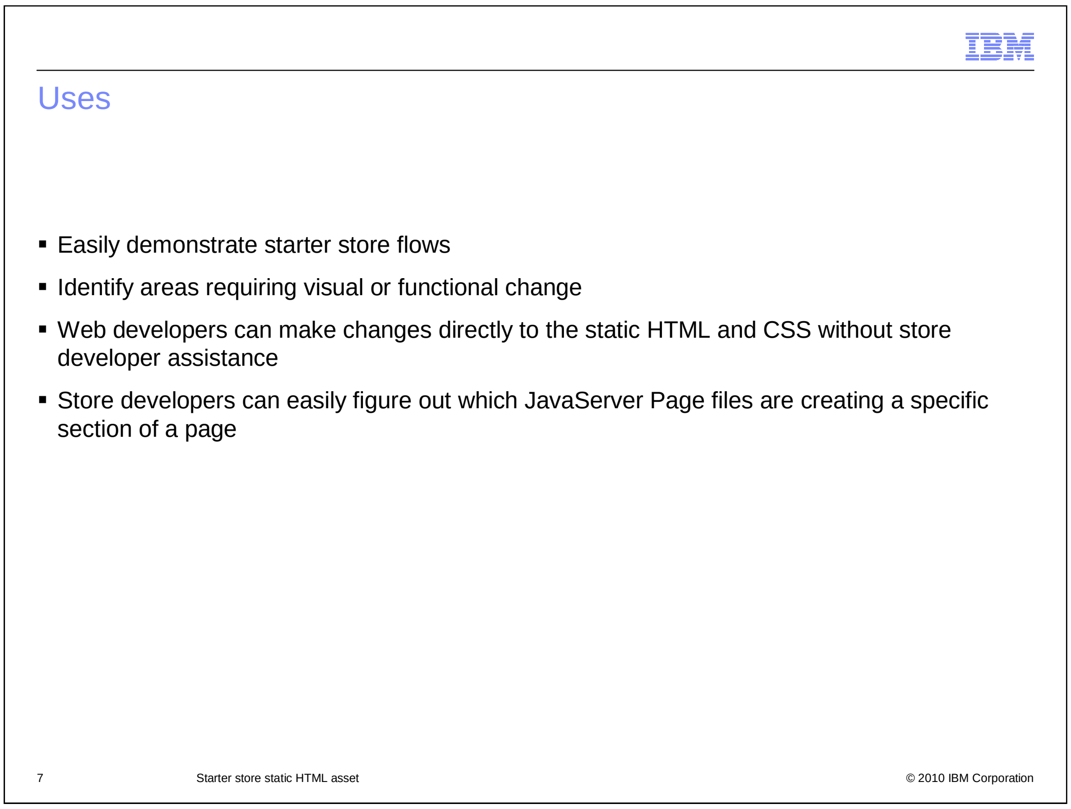
There are often two kinds of developers used when developing stores with WebSphere Commerce. There are Web design team who are familiar with HTML and CSS and design the look-and-feel of their store. Then there are the JavaServer Page developers who know the WebSphere Commerce APIs and server sides technologies to write the pages. The problem that occurs is that typically the person who designs the page might not be the expert to create the code for the page. Since WebSphere Commerce starter stores are implemented using JavaServer Pages files the implementation phase of a typical project involves translating HTML/CSS-based design into the appropriate WebSphere Commerce JavaServer Page Files. This can be a difficult manual process because it requires that server-side developers locate changes to the HTML and then update the corresponding sections of JavaServer Page files.



What this asset provides

- A set of static, navigable HTML pages
- Coverage of all store pages and various supported flows
- A way to walk through store pages using just a browser, no WebSphere Commerce server needed
- A highlight feature to identify which JavaServer Page files makes up which part of each page

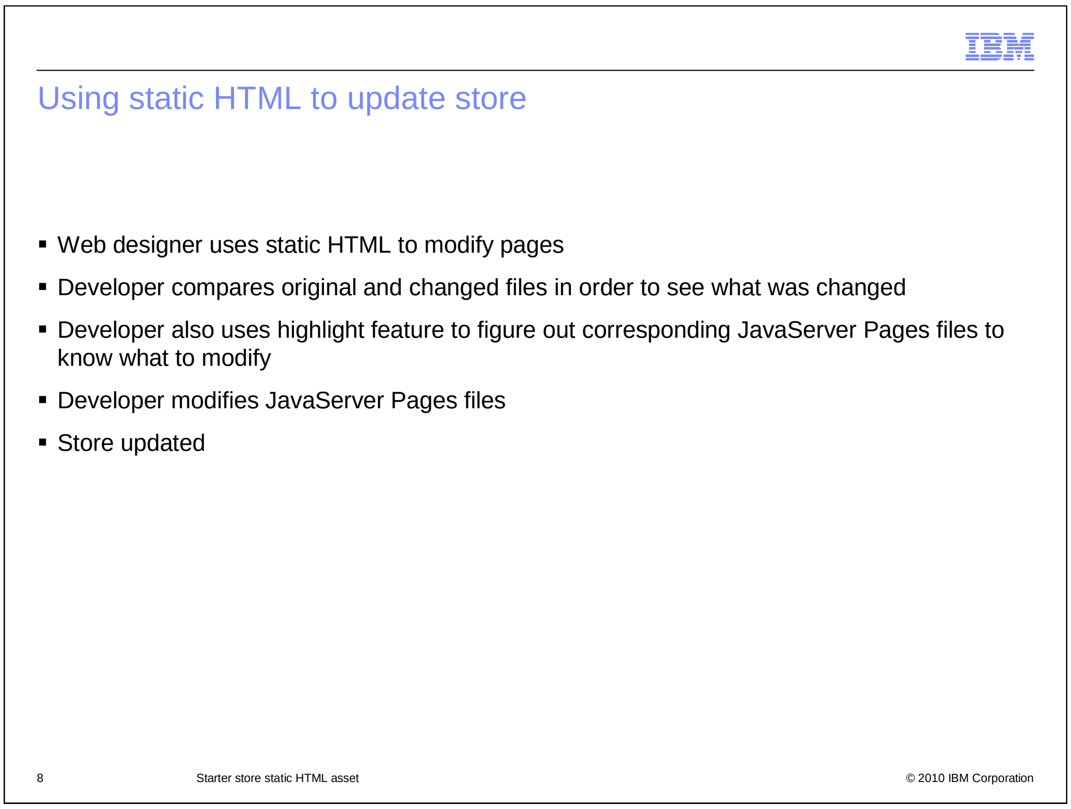
WebSphere Commerce will provide a set of static, navigable HTML pages for both Madisons and Elite starter stores. At no point do any of these pages point to the WebSphere Commerce server. This static asset will showcase all the store pages and their different supported flows. For example, there are different pages to demonstrate an empty shopping cart, a shopping cart page with more than 20 items and pagination invoked, and so on. The asset runs itself by opening the HTML files in a browser without the need of a development environment running on a WebSphere Commerce server. While viewing a static page you can display the names of the JavaServer Page files that make up the current page by using the highlight feature. The highlight feature will mark off each JavaServer Page file and provide its name in the top-left corner of the area.


Uses

- Easily demonstrate starter store flows
- Identify areas requiring visual or functional change
- Web developers can make changes directly to the static HTML and CSS without store developer assistance
- Store developers can easily figure out which JavaServer Page files are creating a specific section of a page

7 Starter store static HTML asset © 2010 IBM Corporation

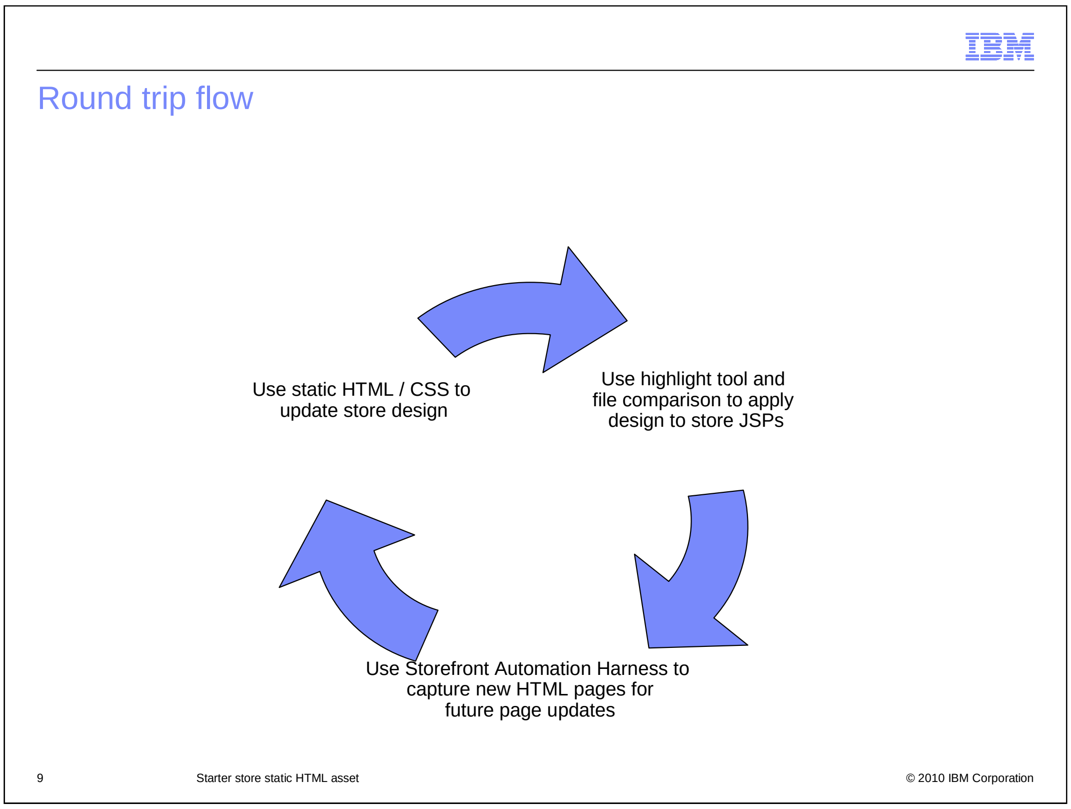
The static HTML pages demonstrate starter store flows for you to prepare for the implementation of your WebSphere Commerce project. You can identify the store appearance or functionality that you want to change and highlight feature “gaps”. You can use these static HTML pages to easily modify the HTML and CSS without the use of a development environment and later incorporate the changes into JavaServer Page files that make up a production store page. You can determine which JavaServer Page files are used to dynamically render a particular section in the store using the highlight feature.



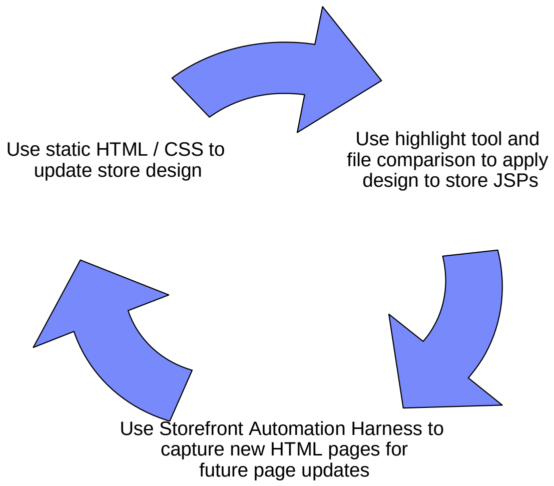
Using static HTML to update store

- Web designer uses static HTML to modify pages
- Developer compares original and changed files in order to see what was changed
- Developer also uses highlight feature to figure out corresponding JavaServer Pages files to know what to modify
- Developer modifies JavaServer Pages files
- Store updated

Here is a high level overview of the steps that you take to make full advantage of the static HTML assets. A Web designer will use the static HTML to modify the pages and design the store. A store developer then picks up the changes and does some caparisons on the HTML pages to see what needs to be changed. The highlight feature can also be used to help display what the corresponding JavaServer Page files are that need to be updated. After figuring out what files need to be updated, the store developer makes the changes and updates the store.



Round trip flow


Use static HTML / CSS to update store design
Use highlight tool and file comparison to apply design to store JSPs
Use Storefront Automation Harness to capture new HTML pages for future page updates

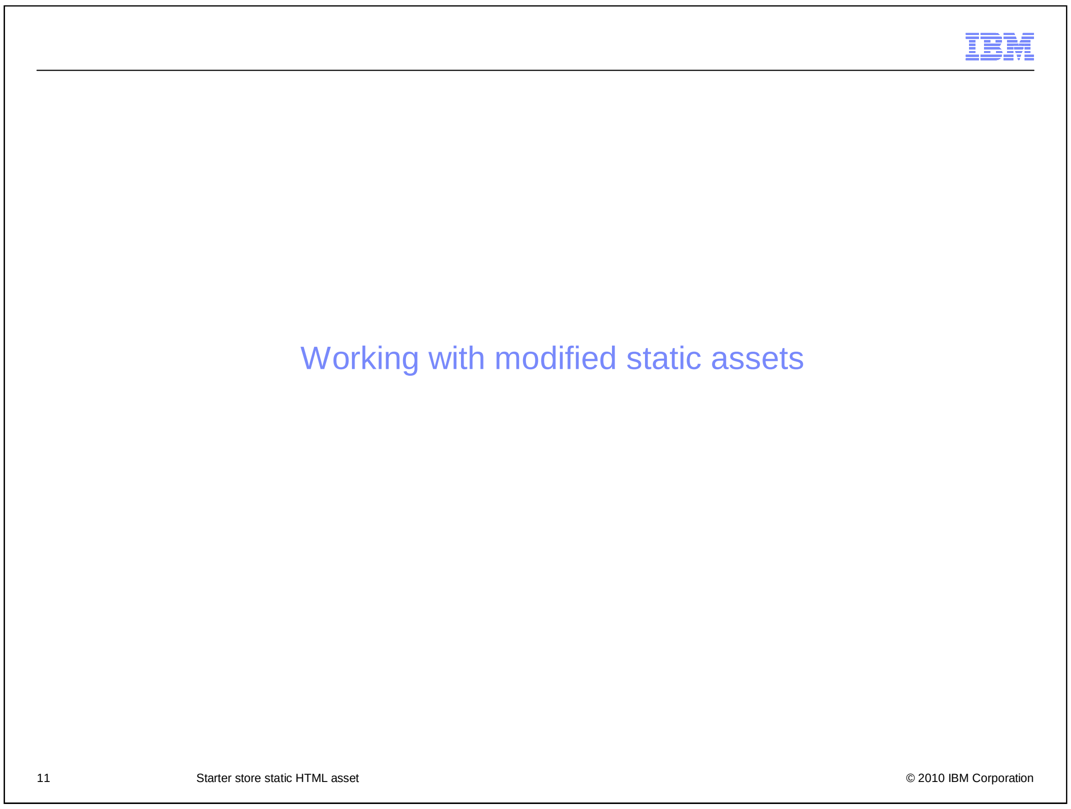
9 Starter store static HTML asset © 2010 IBM Corporation

The ideal situation is to modify the store making modifications only to the CSS and images, but this is not realistic as most will have to change some HTML. So the process is to use these static HTML pages. If you want to do store development based on Madisons or Elite store you can take the static HTML pages and review the scenarios and decide which ones you want to edit. You then can give the HTML / CSS package to the creative team for editing being sure that they do not remove the page identifier comments. The Web design team will style the pages the way they like and then they can give it to the WebSphere Commerce developer. If you are the developer, your job is to translate all the changes back into the real store. The CSS and images will transfer, but not the changes made to the HTML. To update the changes made in the HTML you can look at the changed static HTML pages. You can use a comparison tool to compare the original and changed files in order to be able to see what sections of the page the designer changed. You can also use the highlight feature that is part of the static HTML pages. This tool marks up the page with boarders and gives you the JavaServer Page files names. The developer then updates the store JavaServer Pages files accordingly. After this is complete you will want to capture new static HTML pages. You will use the Storefront Test Automation Engine to capture the new HTML pages for future updates.

Package components

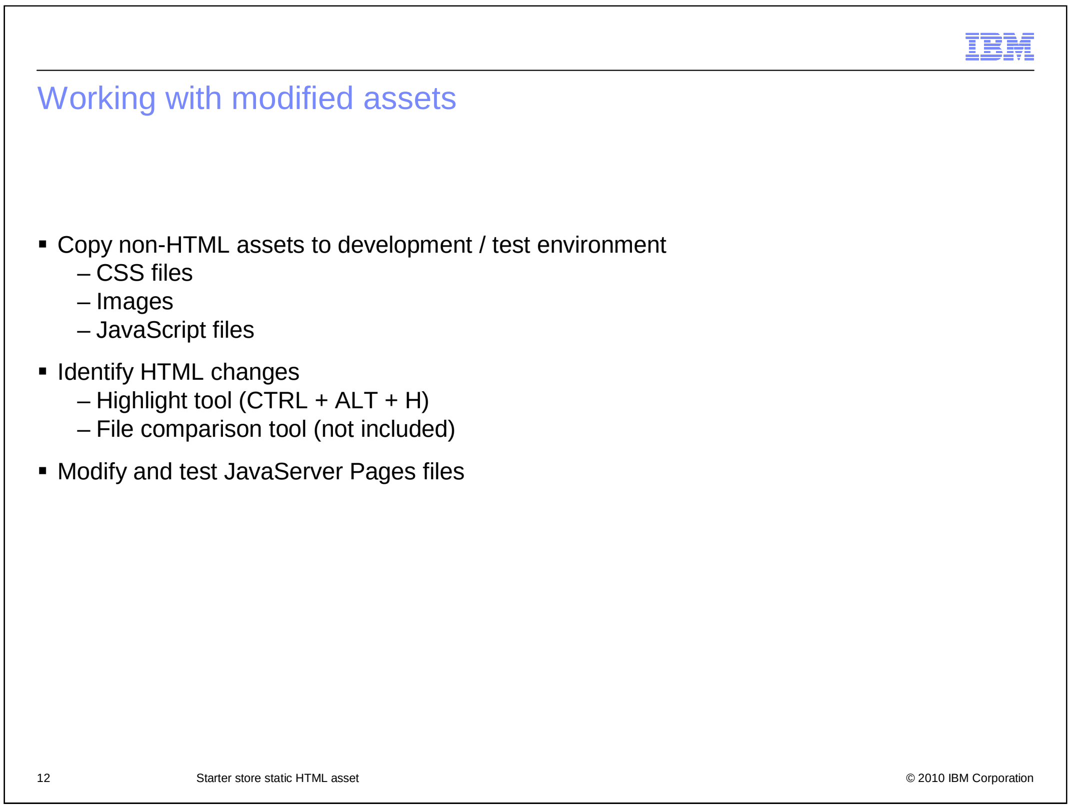
- A starter page, index.html, to link to each package
- Various static packages for Madisons and Elite that showcase different flows
  - Guest shopper
  - Registered shopper
- CSS, image, and JavaScript files
- Dojo framework
- Supporting tools used for generating navigable static HTML assets

That static HTML package contains various components to help you. First it contains an index HTML page to get started. This page provides links to all the different “packages” for the Madisons and Elite starter stores and a description of each. You can navigate the static HTML pages within a “package” using the dynamic links on the store pages. Static HTML pages for Madisons and Elite starter stores are divided into “packages” that represent a series of different flows through the store. For example, one “package” contains the pages seen during a guest checkout flow whereas another contains the pages seen during a checkout flow performed by a registered customer using Quick Checkout. The static HTML package also contains the CSS, images, and JavaScript files referenced by the static HTML pages. The Dojo framework is required for the static stores to behave like they do when running on a WebSphere Commerce server is also contained in the package. Finally, the bin directory contains the tools required to generate your own set of static pages.



Working with modified static assets

The next section talks about how to work with the modified static assets.

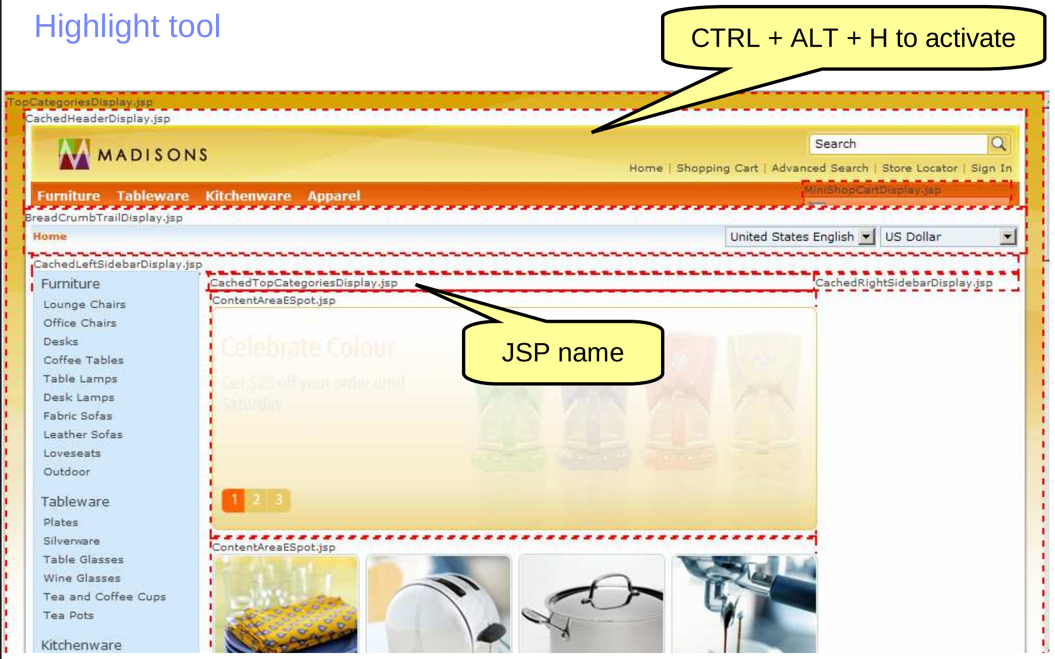


Working with modified assets

- Copy non-HTML assets to development / test environment
  - CSS files
  - Images
  - JavaScript files
- Identify HTML changes
  - Highlight tool (CTRL + ALT + H)
  - File comparison tool (not included)
- Modify and test JavaServer Pages files

When getting the files from the Web design team you should copy the non-HTML assets to your development/test environment. Be sure to copy the updated CSS files, images, and JavaScript files. You should then open the static HTML pages and use the highlight tool (CTRL – ALT – H). This will help you identify which JavaServer Page files to edit. You can also use a comparison tool to help identify the changes you need to make. Finally, you can modify and test out your JavaServer Pages files.

Highlight tool

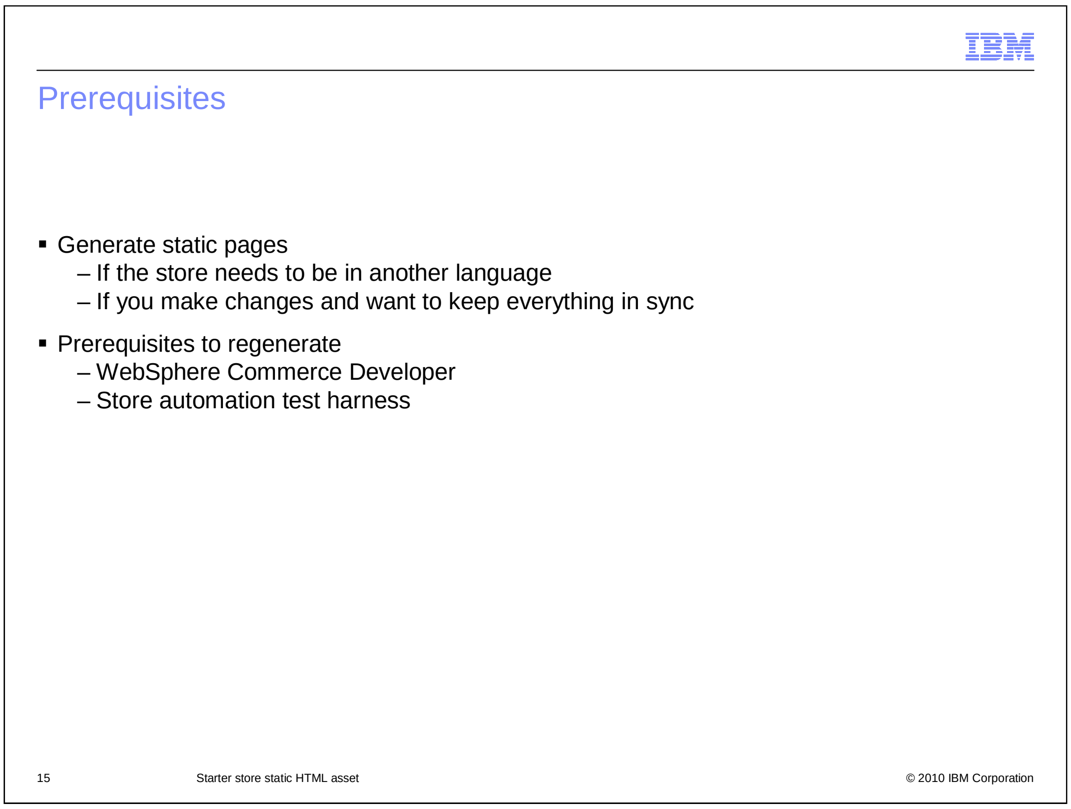


You can use the highlight feature on the static page using CTRL-ALT-H and it will mark up the page with boarders and give you the JavaServer Page files names.

Generating new static assets

14 Starter store static HTML asset © 2010 IBM Corporation

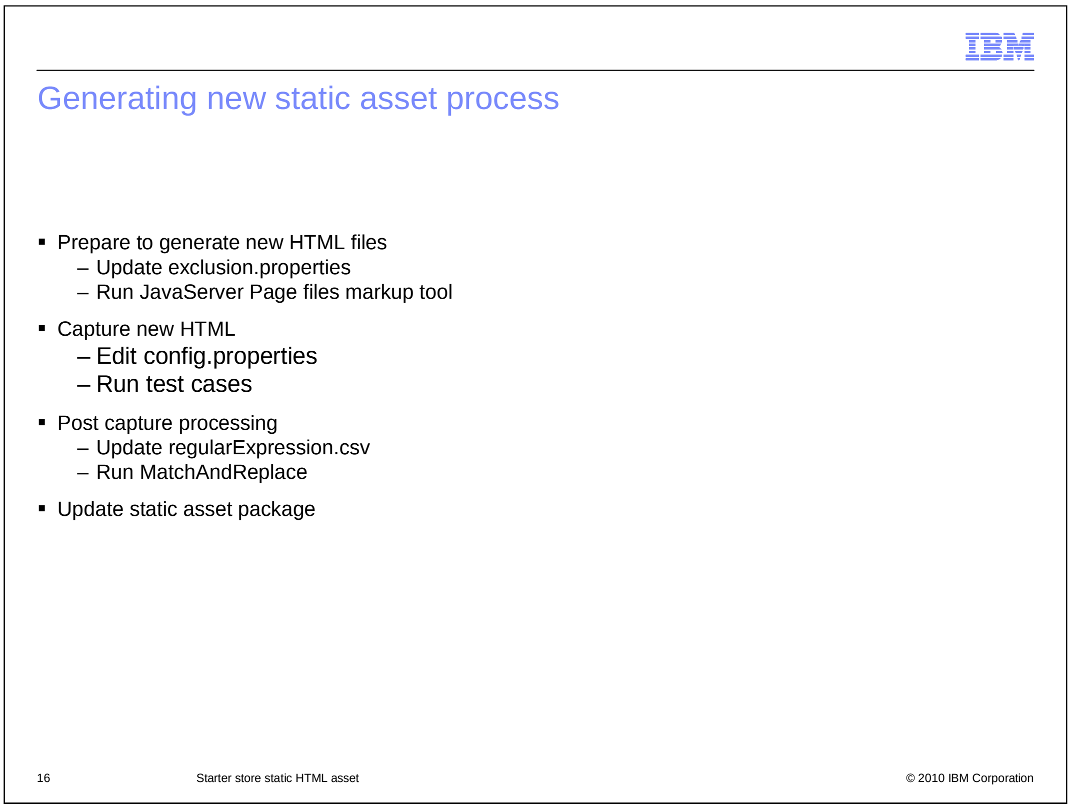
The next section will cover how to generate new static assets.



Prerequisites

- Generate static pages
  - If the store needs to be in another language
  - If you make changes and want to keep everything in sync
- Prerequisites to regenerate
  - WebSphere Commerce Developer
  - Store automation test harness

There are two main reasons to generate new static pages. First the static HTML pages are only in English. If the store needs to be in another language you will need to generate new static pages. If you make changes to the store and want to keep everything in sync you will need to regenerate the static pages. In order to regenerate the static pages you will need WebSphere Commerce Developer. You will also need the Storefront Test Automation Engine. This is another asset in WebSphere Commerce V7 Feature Pack 1 that is used for automated test script for the store. HTML is captured while running the test cases in this script

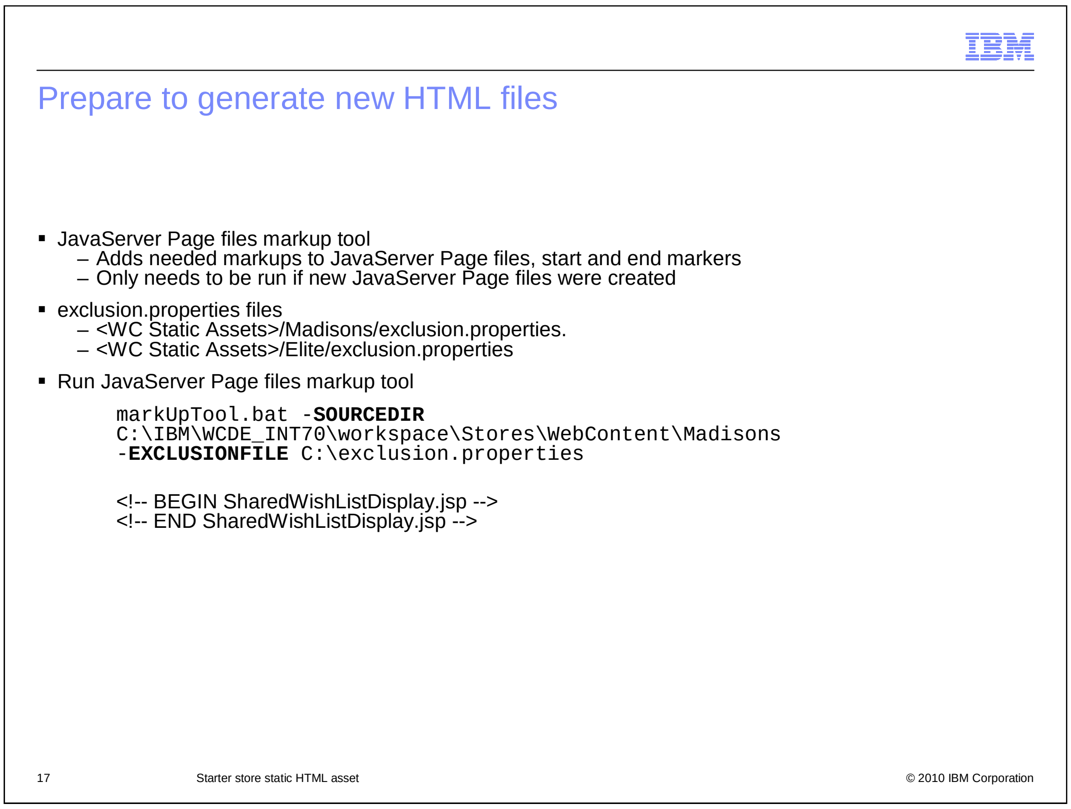


Generating new static asset process

- Prepare to generate new HTML files
  - Update exclusion.properties
  - Run JavaServer Page files markup tool
- Capture new HTML
  - Edit config.properties
  - Run test cases
- Post capture processing
  - Update regularExpression.csv
  - Run MatchAndReplace
- Update static asset package

16 Starter store static HTML asset © 2010 IBM Corporation

Listed here is the process of generating the new static assets. You will first run the JavaServer Page files markup tool, then run test cases to capture the HTML pages, run the MatchAndReplace tool, and finally update your static asset package. The next few slides will cover each of these steps in more detail.



Prepare to generate new HTML files

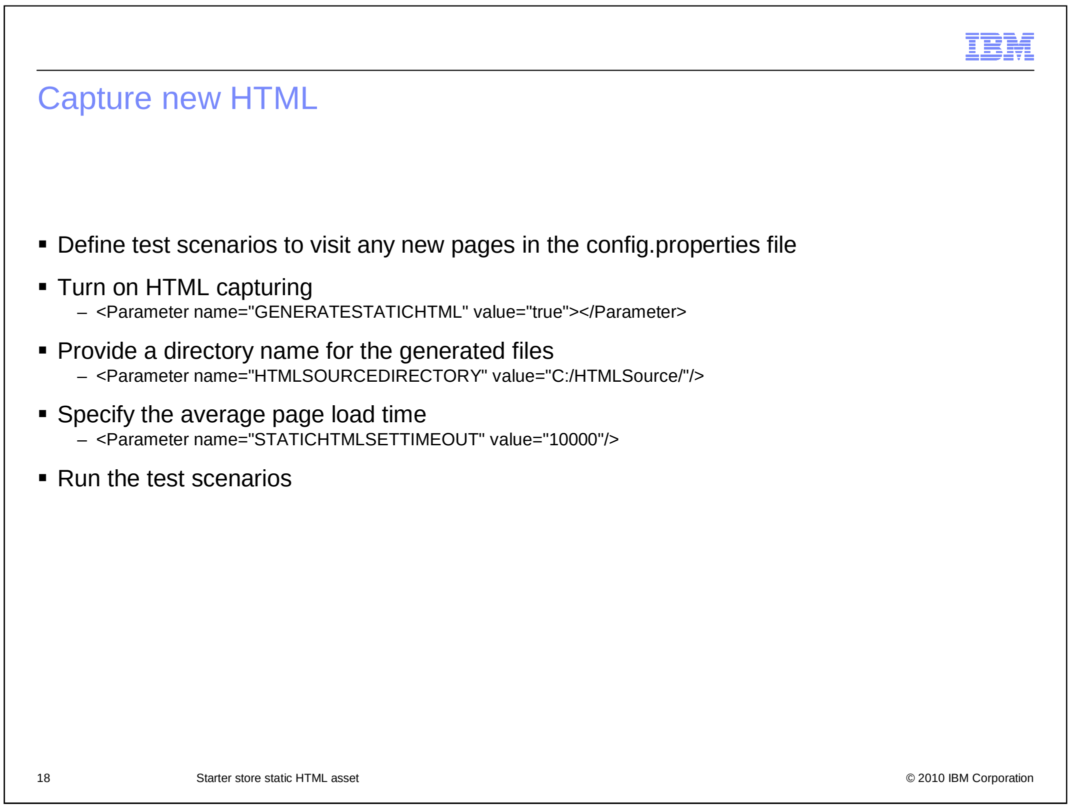
- JavaServer Page files markup tool
  - Adds needed markups to JavaServer Page files, start and end markers
  - Only needs to be run if new JavaServer Page files were created
- exclusion.properties files
  - <WC Static Assets>/Madisons/exclusion.properties.
  - <WC Static Assets>/Elite/exclusion.properties
- Run JavaServer Page files markup tool

```
markUpTool.bat -SOURCEDIR
C:\IBM\WCDE_INT70\workspace\Stores\WebContent\Madisons
-EXCLUSIONFILE C:\exclusion.properties
```

```
<!-- BEGIN SharedWishListDisplay.jsp -->
<!-- END SharedWishListDisplay.jsp -->
```

17 Starter store static HTML asset © 2010 IBM Corporation

In order to generate the static HTML pages and use features like highlight, the JavaServer Page files need to have markers inserted into them. This JavaServer Page files markup tool step is only needed if you created any new JavaServer Pages files. Existing JavaServer Pages files in the Madison and Elite stores already contain the necessary markups. The markup tool accepts a directory as input. It goes into the directory and all subdirectories to find all the JavaServer Page files. In each of the JavaServer Page files, it inserts the name of the corresponding file name as beginning and closing comments. If some files, such as JavaServer Page fragments, do not need to be marked up they can be specified in an exclusion list. You can see examples in the exclusion.properties files that exist for each of the starter stores. A log file is generated capturing the list of modified files and the errors if any.



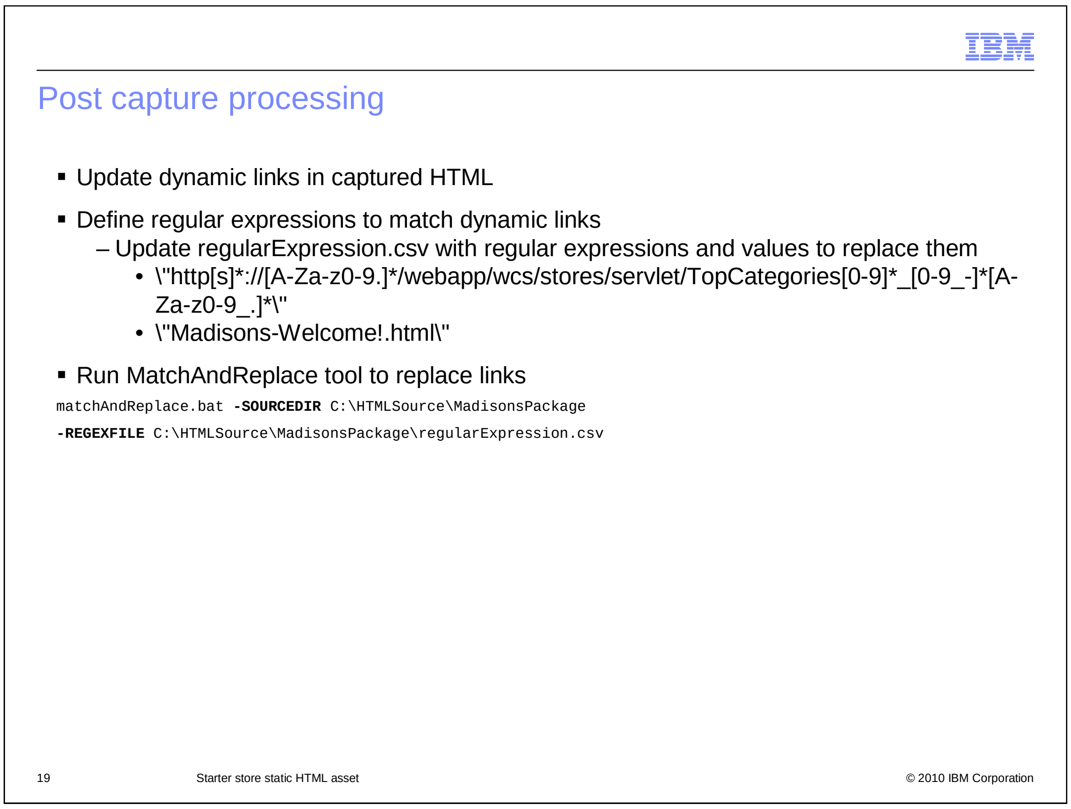
Capture new HTML

- Define test scenarios to visit any new pages in the config.properties file
- Turn on HTML capturing
  - <Parameter name="GENERATESTATICHTML" value="true"></Parameter>
- Provide a directory name for the generated files
  - <Parameter name="HTMLSOURCEDIRECTORY" value="C:/HTMLSource/"/>
- Specify the average page load time
  - <Parameter name="STATICHTMLSETTIMEOUT" value="10000"/>
- Run the test scenarios

18 Starter store static HTML asset © 2010 IBM Corporation

When test cases are run through the test harness, the static HTML pages can be captured. If you add new pages to your store then you will need to define new test scenarios for these pages so that they are captured. With the help of the Selenium software testing framework for Web applications, you can generate the required HTML source pages by running your new test scripts. To use a test script to generate HTML you need to edit some configuration values in the config.properties file. The config.properties file for the Madisons store is located in the Madisons-Tests project folder. For Elite, it is in Elite-Tests project folder. Madisons-Tests and Elite-Tests are the projects that come with the Storefront Test Automation Engine. You will need to set the values for the “GENERATESTATICHTML”, “HTMLSOURCEDIRECTORY” and “STATICHTMLSETTIMEOUT“ environment parameters in the data file of that test case. You then need to call the method activateStaticHTMLCapture to pass on these parameters from the test case to the Selenium main class. The value of the flag “GENERATESTATICHTML” needs to be set to “true” in the data file to generate the HTML source page. The static HTML is placed in the location given by the value of the “HTMLSOURCEDIRECTORY”. The value of the “STATICHTMLSETTIMEOUT” should be updated based on the network traffic and the time taken to load a page completely. The tool will get the value of “STATICHTMLSETTIMEOUT “and waits for that many milliseconds before capturing the HTML source of the page. The STATICHTMLSETTIMEOUT value should not need to be changed.

When the Selenium engine runs the automated test case, HTML source is captured when the pages are loaded and the source files are saved in the given directory. The images and the CSS files for the corresponding store need to be manually copied to the directory where the HTML files are saved. The image and the CSS path in the HTML files are modified to match the current directory before saving the HTML source of the pages.



Post capture processing

- Update dynamic links in captured HTML
- Define regular expressions to match dynamic links
  - Update regularExpression.csv with regular expressions and values to replace them
    - \"http[s]*://[A-Za-z0-9.]*/webapp/wcs/stores/servlet/TopCategories[0-9]*_[0-9_-]*[A-Za-z0-9_.]*\"
    - \"Madisons-Welcome!.html\"
- Run MatchAndReplace tool to replace links

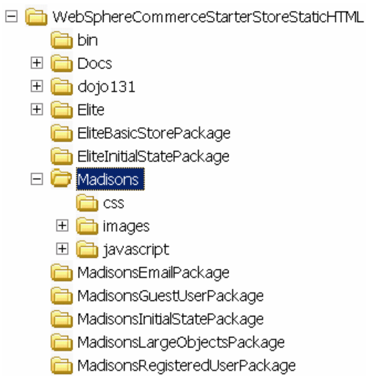
```
matchAndReplace.bat -SOURCEDIR C:\HTMLSource\MadisonsPackage
-REGEXFILE C:\HTMLSource\MadisonsPackage\regularExpression.csv
```

19 Starter store static HTML asset © 2010 IBM Corporation

Once all the required source pages are created you will use the “MatchAndReplace” utility function. You need to make sure all links, paths to images and CSS files will point to the files in your static HTML asset folder. The MatchAndReplace utility function will take the HTML source directory as input and replaces the navigation links to match your requirements. The links are matched and replaced based on the regular expressions. You specify the replacing value provided in a CSV (Coma Separated Value) file. This tool will help you create the set of pages which can be navigated statically. All the static store pages are modified to include a div element and a highlight button to highlight the corresponding JavaServer Page files snippets in the Static HTML page. This highlight section will help you to identify which store JavaServer Page files to modify based on the Web designer modifications to the styles for that page.

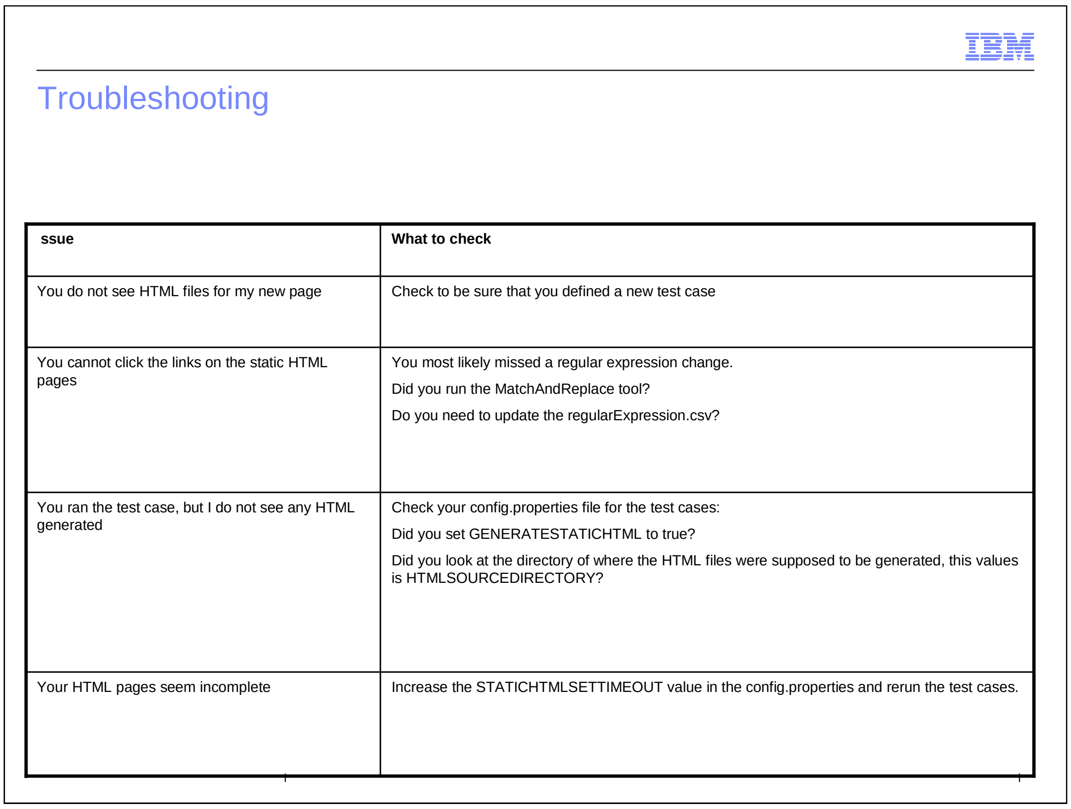
Creating a new static asset package

- Copy CSS, image and JavaScript directories into named store directory under asset root directory
- Copy updated HTML files into a new or existing package directory
- Update index.html as needed
- Test your new static pages



```
⊟ WebSphereCommerceStarterStoreStaticHTML
     bin
  ⊞  Docs
  ⊞  dojo131
  ⊞  Elite
     EliteBasicStorePackage
     EliteInitialStatePackage
  ⊟  Madisons
        css
     ⊞  images
     ⊞  javascript
     MadisonsEmailPackage
     MadisonsGuestUserPackage
     MadisonsInitialStatePackage
     MadisonsLargeObjectsPackage
     MadisonsRegisteredUserPackage
```

If you create new Elite or Madisons static HTML pages you can replace the existing static HTML files in the existing directory structure. You will need to add any new CSS files, images or JavaScript files and possibly update the index.html file. You can also go ahead and create a new static asset package. You will want to do this if you create your own store. You will need to create a new directory for the named store. You need to copy the HTML files into the package directory. Then copy your CSS files, images, and JavaScript directories into the new store directory. The index file is a convenience file that points to the home page of all the packages, you might want to add the new store links into the index.html for easy access into the new store's packages.



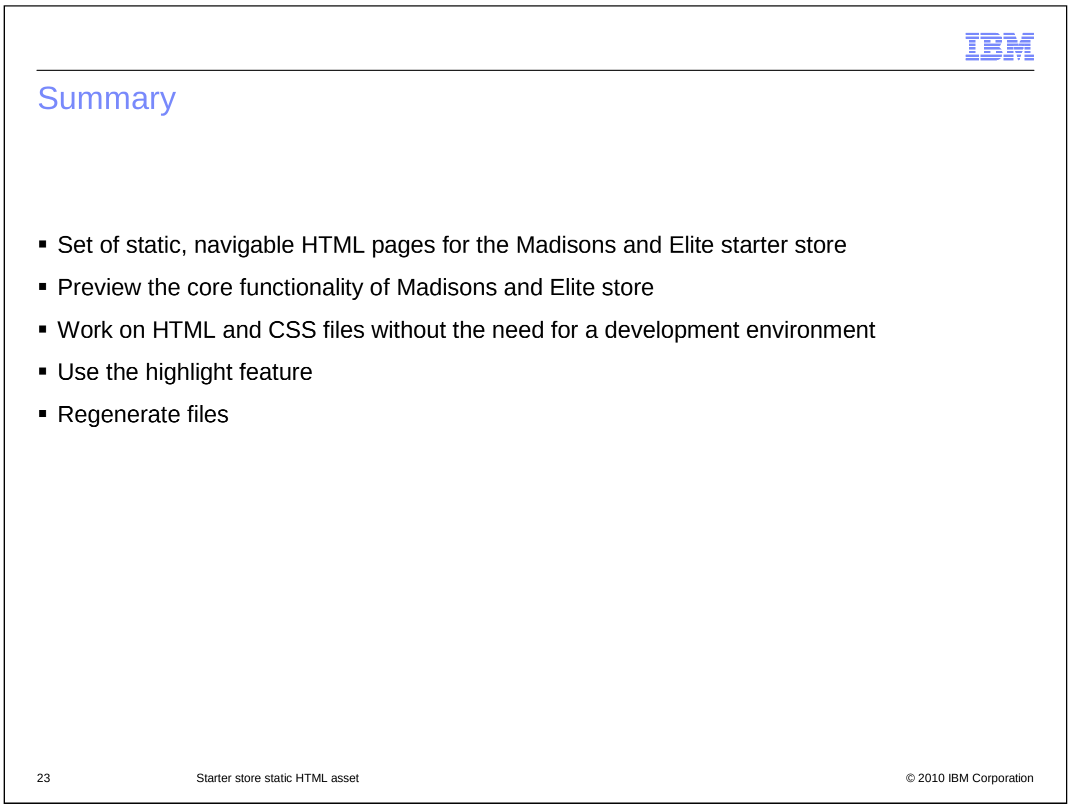
Troubleshooting

ssue	What to check
You do not see HTML files for my new page	Check to be sure that you defined a new test case
You cannot click the links on the static HTML pages	You most likely missed a regular expression change. Did you run the MatchAndReplace tool? Do you need to update the regularExpression.csv?
You ran the test case, but I do not see any HTML generated	Check your config.properties file for the test cases: Did you set GENERATESTATICHTML to true? Did you look at the directory of where the HTML files were supposed to be generated, this values is HTMLSOURCEDIRECTORY?
Your HTML pages seem incomplete	Increase the STATICHTMLSETTIMEOUT value in the config.properties and rerun the test cases.

Here are some common issues that you can experience and some helpful tips for what to check. If you do not see HTML for a new page you have created, be sure to create a test case for your new page, then rerun your test cases. If you open your static HTML page and the links of the page are not working, you most likely missed a regular expression change. Update the regularExpression.csv file and rerun the Match and Replace tool. If you ran the test cases, but do not see any html generated be sure to check your config.properties file for your test cases. You need to make sure that the GENERATESTATICHTML flag is set to true. Also, check where you set the HTML source directory. If you see the issue that your HTML pages seem incomplete, then you might need to increase the STATICHTMLSETTIMEOUT value in the config.properties file and rerun your test cases. Increasing he timeout value gives the page more time to fully render before the HTML is captured. The default value should be fine but if you made your pages bigger, and you can see that the rendered pages are incomplete, updating the timeout value should solve this issue.

Summary and references

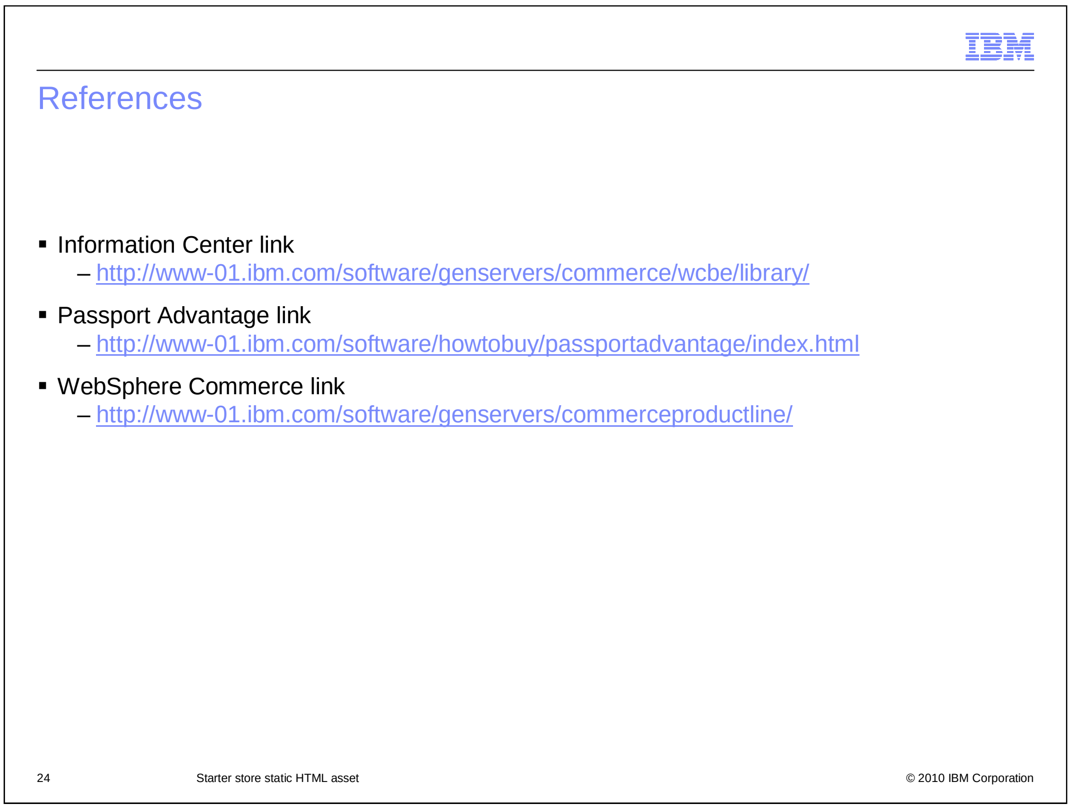
The last section covers the summary and some useful references.



Summary

- Set of static, navigable HTML pages for the Madisons and Elite starter store
- Preview the core functionality of Madisons and Elite store
- Work on HTML and CSS files without the need for a development environment
- Use the highlight feature
- Regenerate files

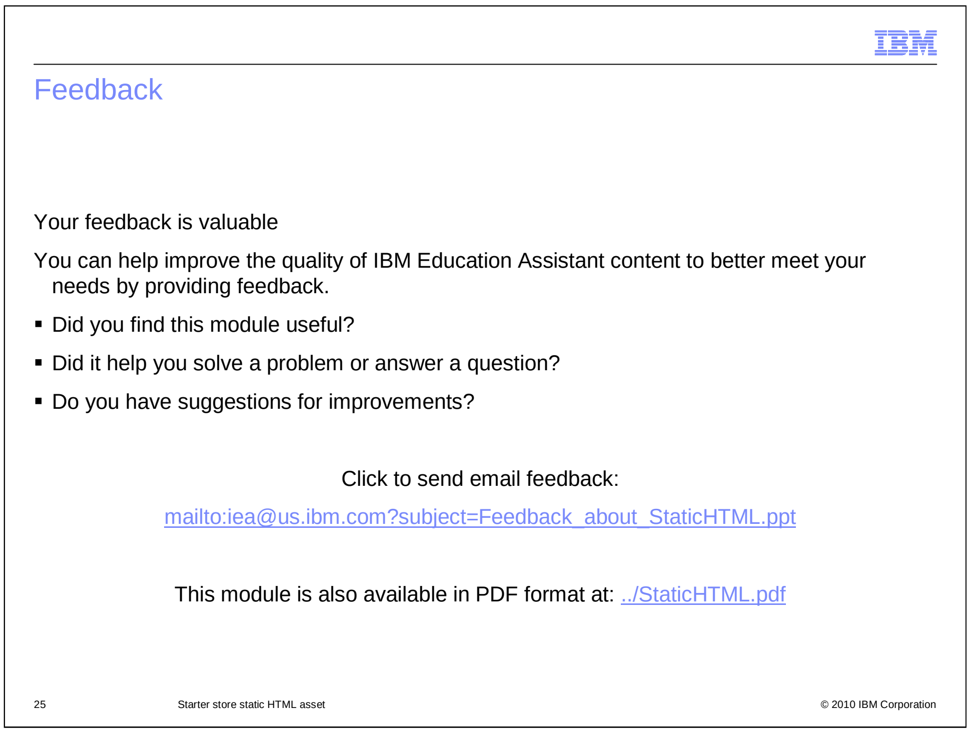
The HTML assets include a set of static, navigable HTML pages for the Madisons and Elite starter stores. You can preview the core functionality of Madisons and Elite store and then work on HTML and CSS files without the need for a development environment running on a WebSphere Commerce server. You can also use the highlight feature to determine which JavaServer Page files dynamically render the different areas, such as the sidebars, of specific store pages. Finally, you can regenerate the HTML files after you have updated the store.



References

- Information Center link
  - http://www-01.ibm.com/software/genservers/commerce/wcbe/library/
- Passport Advantage link
  - http://www-01.ibm.com/software/howtobuy/passportadvantage/index.html
- WebSphere Commerce link
  - http://www-01.ibm.com/software/genservers/commerceproductline/

Here are some useful links.



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_StaticHTML.ppt

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

25 Starter store static HTML asset © 2010 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, Passport Advantage, 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

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

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

© Copyright International Business Machines Corporation 2010. All rights reserved.

© 2010 IBM Corporation