IBM Software Group

IBM WebFacing Tool in IBM Development Studio Client for iSeries: Advanced

iSeries Application Development Team: IBM Toronto

WebSphere. software

e business software

August 2003 | Advanced IBM WebFacing

This presentation reviews WebFacing and what it is. Next we introduce Development Studio Client. Then the features to customize a WebFacing application produced from the IBM WebFacing Tool are described.

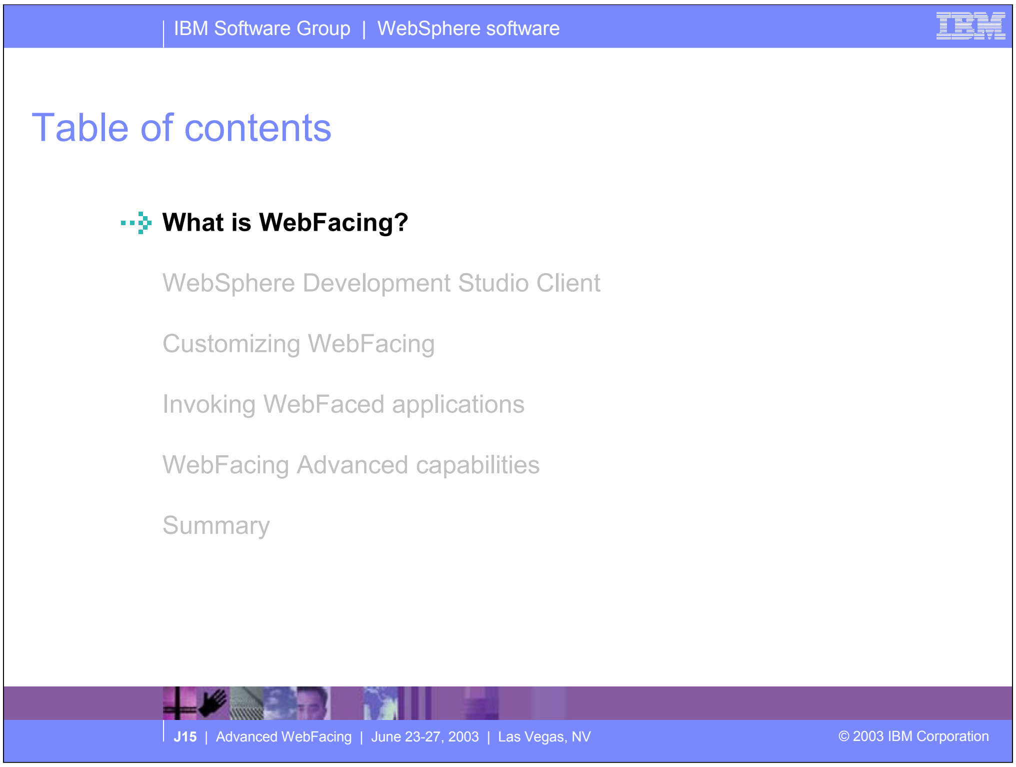
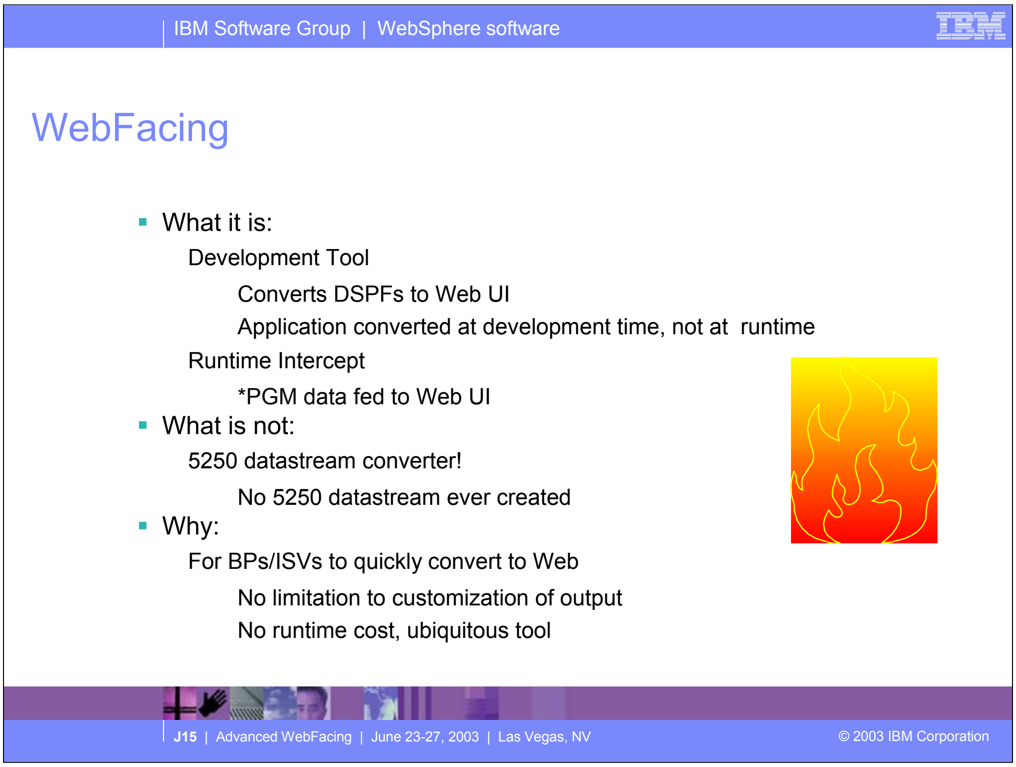


Table of contents

- **What is WebFacing?**
- WebSphere Development Studio Client
- Customizing WebFacing
- Invoking WebFaced applications
- WebFacing Advanced capabilities
- Summary

In the first section of this presentation we review WebFacing and what it is. Next we introduce Development Studio Client. Then customizing a WebFacing application is described.



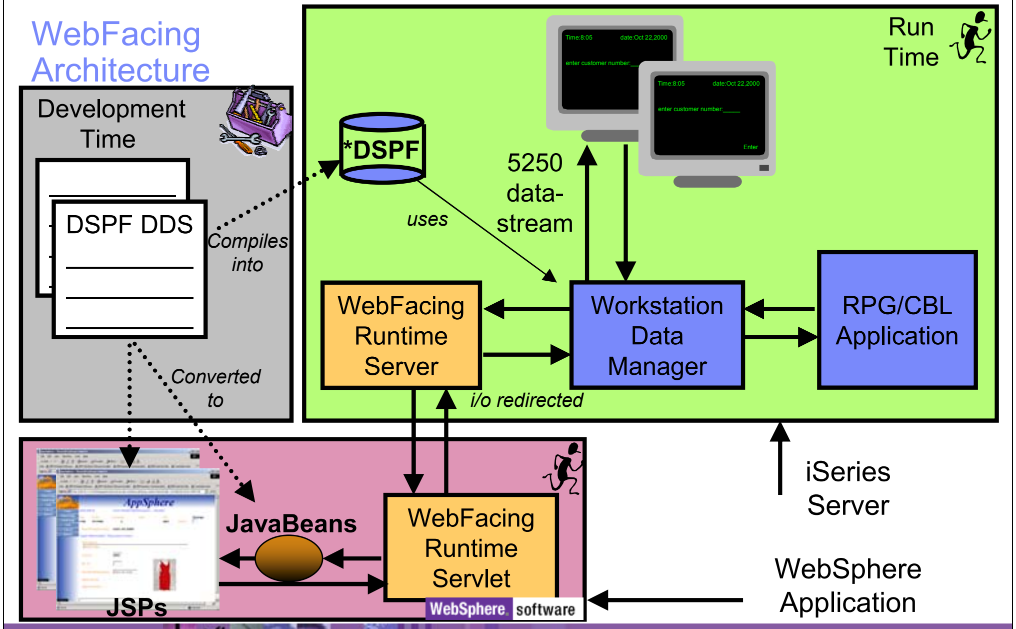
WebFacing

- What it is:
  - Development Tool
    - Converts DSPFs to Web UI
    - Application converted at development time, not at runtime
  - Runtime Intercept
    - *PGM data fed to Web UI
- What is not:
  - 5250 datastream converter!
    - No 5250 datastream ever created
- Why:
  - For BPs/ISVs to quickly convert to Web
    - No limitation to customization of output
    - No runtime cost, ubiquitous tool

There are many options available now for transforming green-screens to Web pages, but WebFacing is unique among them. WebFacing does this conversion at development, taking as input the display file DDS source, and generating JavaServer Pages for the output (for example, Web pages).

The JSPs generated by WebFacing are then deployed to a Web application server, such as WebSphere Application Server, and the application runs as is, but in a Web browser. To enable this to work, with no changes to the underlying application, there is a runtime intercept in OS/400, which detects when an application is running in WebFacing mode. When this happens, the screen data from the application is passed directly to the WebFacing runtime, instead of being used to generate a 5250 datastream. So WebFacing absolutely is not a screen scraper. Indeed, it can't be because there is no 5250 datastream to scrape when running in WebFacing mode. WebFacing is the result of business partners pleading for a conversion option that allows infinite finessing of the generated output, and is cost effective. WebFacing enables this because its output is editable at development time, and the runtime is a free part of OS/400 in V4R5 or higher.

WebFacing Architecture



Development Time

DSPF DDS

Compiles into

*DSPF

Converted to

uses

5250 data-stream

Run Time

WebFacing Runtime Server

Workstation Data Manager

RPG/CBL Application

i/o redirected

iSeries Server

JavaBeans

JSPs

WebFacing Runtime Servlet

WebSphere software

WebSphere Application

First the application is converted. This creates JSPs for each record format as well as Java beans. When the program performs a READ on a record format, control as well as the application data is sent to the Workstation Data Manager.. Because the job was started by the WebFacing Runtime Server, Workstation Data Manager knows this is a WebFacing request and passes the data and control to the WebFacing Runtime Server which runs on the iSeries. Control returns to the Webfacing Runtime Servlet that runs in WebSphere Application Server. The WebFacing Runtime Servlet locates the appropriate JSPs and Java beans. The WebFacing Runtime Servlet tells WebSphere Application Server to return the JSP back to the browser. The JSP is compiled and the resultant HTML is returned to the browser. Note that no 5250 datastream is generated in this flow.

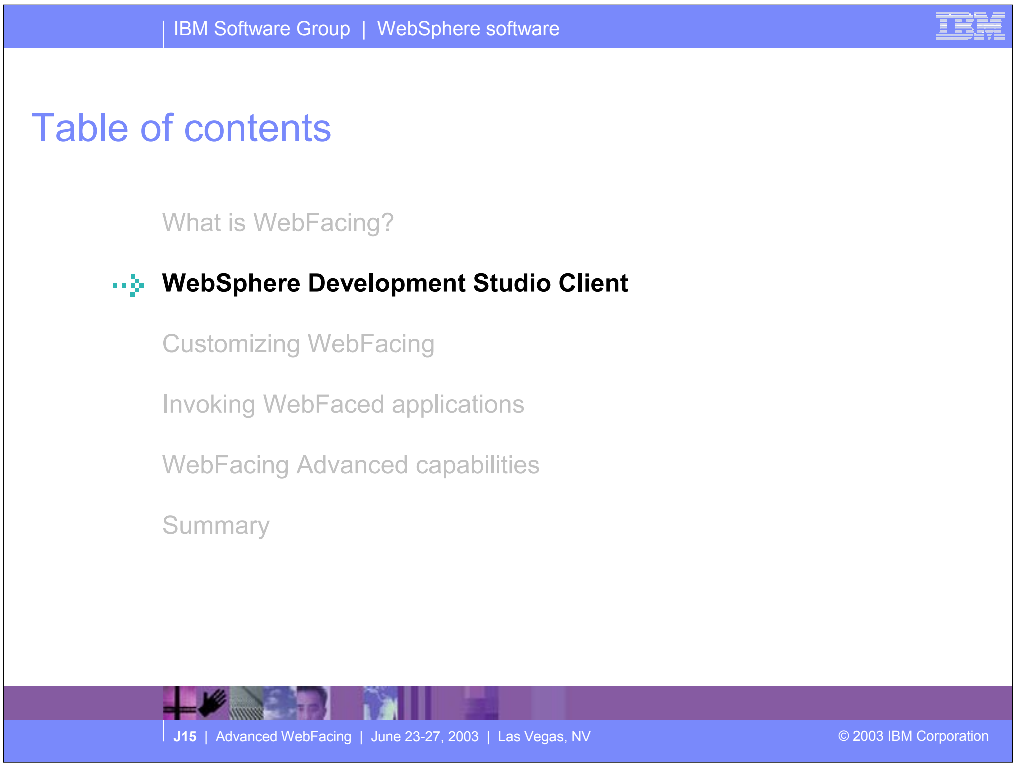


Table of contents

What is WebFacing?

WebSphere Development Studio Client

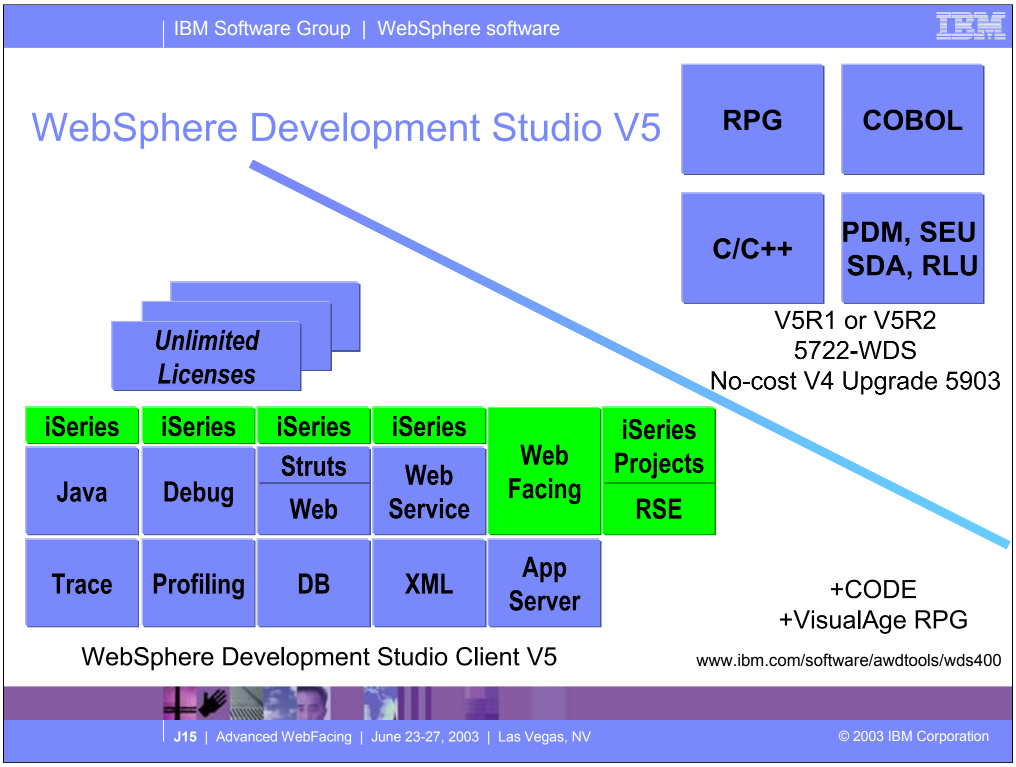
Customizing WebFacing

Invoking WebFaced applications

WebFacing Advanced capabilities

Summary

In the first section of this presentation we introduced WebFacing. Next we introduce Development Studio Client.



WebSphere Development Studio V5

RPG

COBOL

C/C++

PDM, SEU SDA, RLU

V5R1 or V5R2
5722-WDS
No-cost V4 Upgrade 5903

Unlimited Licenses

iSeries Java

iSeries Debug

iSeries Struts Web

iSeries Web Service

Web Facing

iSeries Projects RSE

Trace

Profiling

DB

XML

App Server

+CODE
+VisualAge RPG

WebSphere Development Studio Client V5

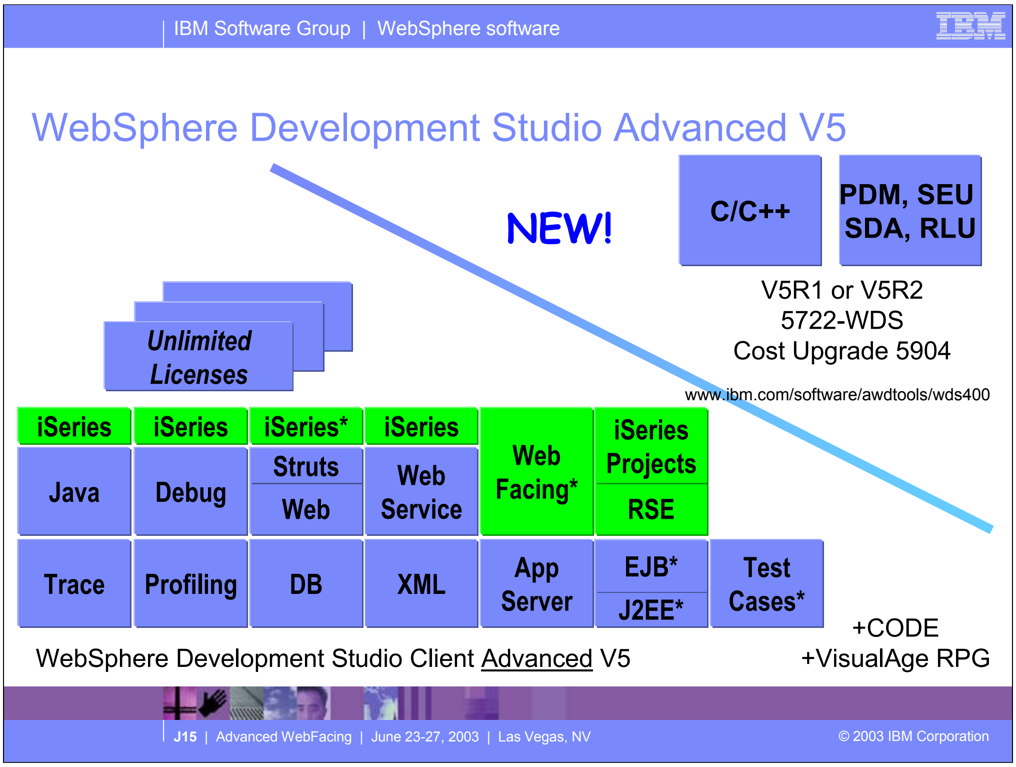
www.ibm.com/software/awdtools/wds400

There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, Development Studio Client is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.



WebSphere Development Studio Advanced V5

NEW!

C/C++

PDM, SEU
SDA, RLU

V5R1 or V5R2
5722-WDS
Cost Upgrade 5904

www.ibm.com/software/awdtools/wds400

Unlimited Licenses

iSeries Java

iSeries Debug

iSeries* Struts Web

iSeries Web Service

Web Facing*

iSeries Projects RSE

Trace

Profiling

DB

XML

App Server

EJB* J2EE*

Test Cases*

+CODE
+VisualAge RPG

WebSphere Development Studio Client Advanced V5

The Advanced edition of Development Studio Client, and Development Studio, is new as of April 25th 2003. Development Studio Advanced is currently the same as Development Studio, except the customer is entitled to unlimited licenses of Development Studio Client Advanced versus just Development Studio Client.

The difference of Development Studio Client Advanced over Development Studio Client is that it has additional tools (blue boxes with asterisks) and some enhancements to existing iSeries tools (green boxes with asterisks) which will be discussed later in this presentation. The majority of the new tools are related to Enterprise Java Bean (EJB) development. You'll see later that Development Studio Client Advanced is based on WebSphere Application Developer (Application Developer) versus WebSphere Studio Site Developer (Site Developer).

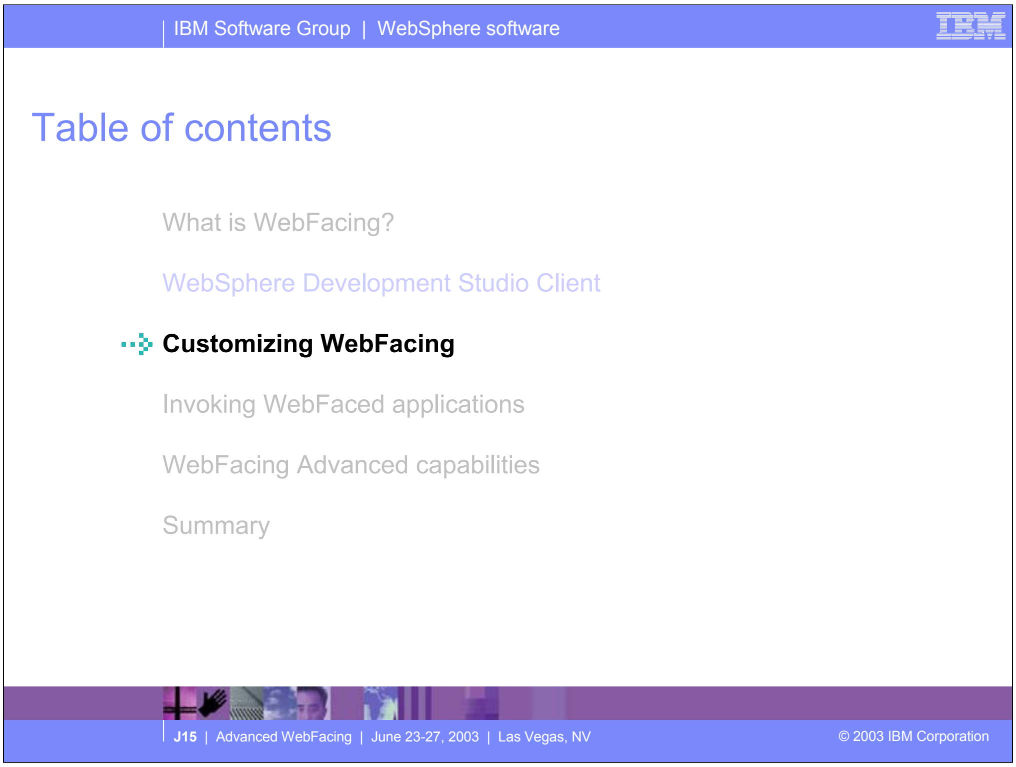


Table of contents

What is WebFacing?

WebSphere Development Studio Client

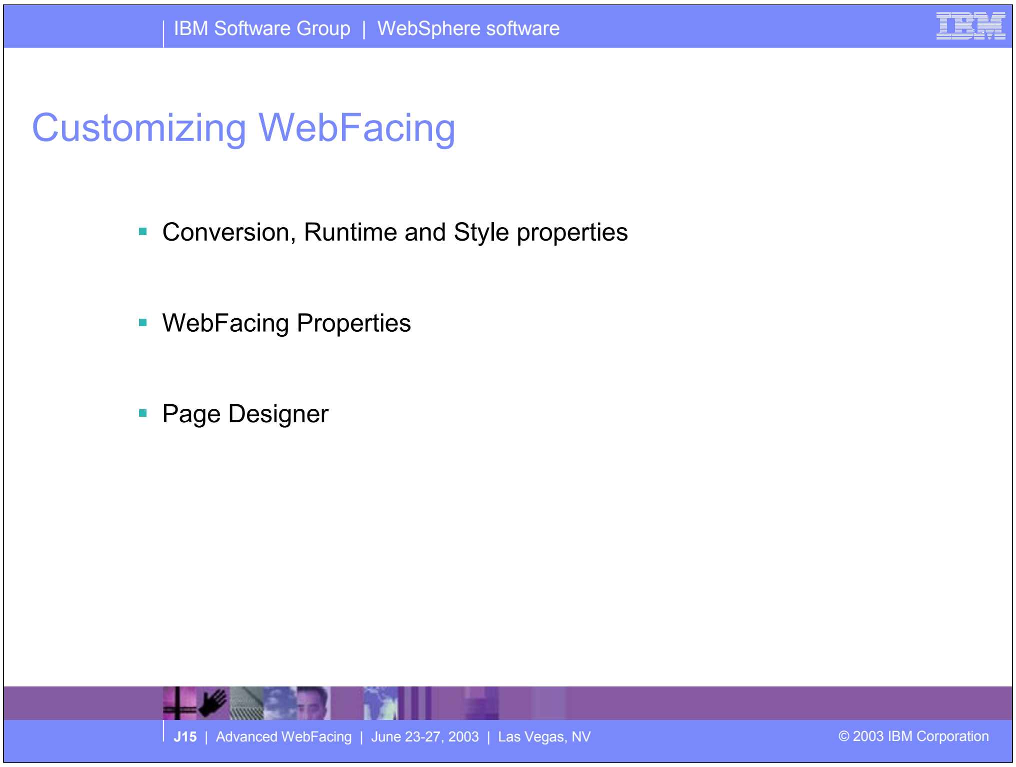
Customizing WebFacing

Invoking WebFaced applications

WebFacing Advanced capabilities

Summary

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.

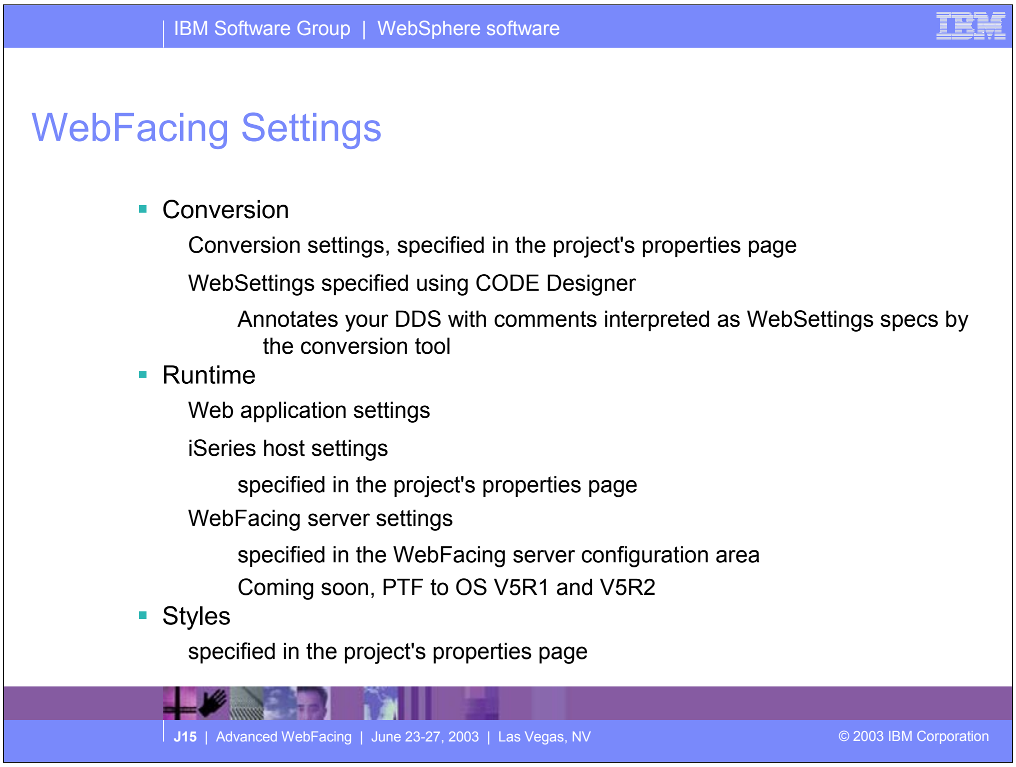


Customizing WebFacing

- Conversion, Runtime and Style properties
- WebFacing Properties
- Page Designer

This section reviews the 3 ways to customize WebFaced applications.

1. Before conversion, use properties dialog to change conversion, runtime and style properties.
2. Before conversion use the WebSettings feature of CODE designer. WebSettings are stored as comments in the DDS. With WebSettings you may have certain fields that may not be relevant on the Web. You can specify that a field contains an image name, and an <IMG> tag should be created for that field. You can specify that a field is a hyper-link. Clicking on the link at run-time will launch the URL specifed in WebSettings.
3. After conversion, WebFacing copies the Cascading Style Sheet (CSS) to the project. With some knowledge of CSS you can modify it to customize your pages. You can use Page Designer to edit the created JSPs. Note, conversion will overwrite customized JSPs



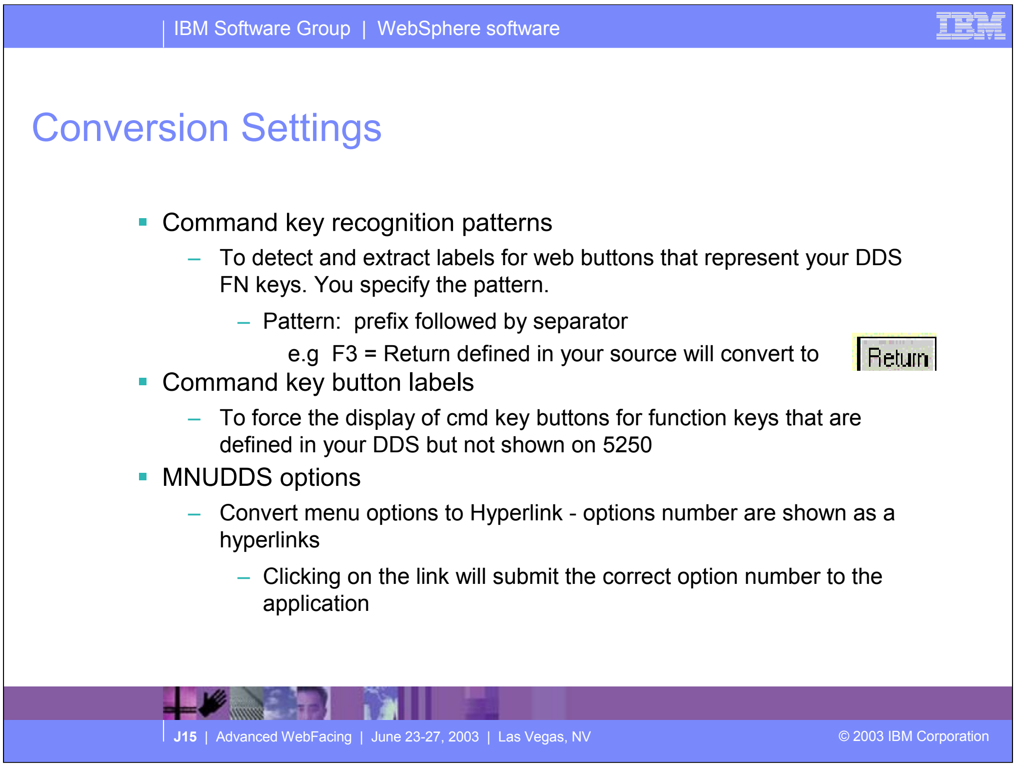
WebFacing Settings

- Conversion
  - Conversion settings, specified in the project's properties page
  - WebSettings specified using CODE Designer
    - Annotates your DDS with comments interpreted as WebSettings specs by the conversion tool
- Runtime
  - Web application settings
  - iSeries host settings
    - specified in the project's properties page
  - WebFacing server settings
    - specified in the WebFacing server configuration area
    - Coming soon, PTF to OS V5R1 and V5R2
- Styles
  - specified in the project's properties page

Conversion properties control how selected DSPF and UIM files are being converted for WebFacing use. Values for conversion properties are stored in the file conversion.rules under the config directory of the WebFacing project. You can use the Web Settings tab in CODE Designer to customize how your programs will look and function when accessed through a Web browser. An advantage to customizing your pages with Web Settings is that the instructions for the customizations are embedded as comments in your DDS source; since Web Settings become part of your source, changes that you make are not lost if you later reconvert a WebFacing project.

Run-time properties determine the behavior of the converted Web application when it is being used by an end user.

You can customize the look of the application area and the command keys using the Style properties. If you want to change the look of the layout and frame surrounding these areas, you must use a CSS editor to update the style files stored in the chrome directory.



Conversion Settings

- Command key recognition patterns
  - To detect and extract labels for web buttons that represent your DDS FN keys. You specify the pattern.
    - Pattern: prefix followed by separator

      e.g F3 = Return defined in your source will convert to Return
- Command key button labels
  - To force the display of cmd key buttons for function keys that are defined in your DDS but not shown on 5250
- MNUDDS options
  - Convert menu options to Hyperlink - options number are shown as a hyperlinks
    - Clicking on the link will submit the correct option number to the application

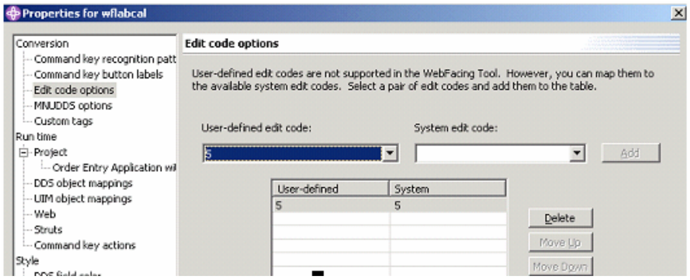
Use the **Command key recognition patterns** tab to specify how to detect the text for command key buttons to be displayed in a Web browser. When converted, command keys are displayed as buttons on your Web pages. Users click these buttons to launch an event defined in your DDS source or to get online help.

Use the **Command key button labels** tab to specify the default text for command keys identified during conversion that do not match a defined recognition pattern. For example, if no recognition pattern has been defined that would match F1=Help, you could specify the value for the **Command Key** field as F1 and the value for **Button label** as Help. When converted, command keys are displayed as 'Help' buttons on your Web pages. Users click these buttons to launch an event defined in your DDS source such as to refresh their page or to get online help.

If you have selected DDS members of type MNUDDS for conversion, use the **MNUDDS options** tab to convert the menu options into hypertext links. The command option number and any text included in the same field where that number is found will be converted into a hypertext link. The option of automatically converting commands into hypertext links is selected by default. In the **Separator** drop-down list, specify a separator to identify the menu options. The default separator is the period (.). The other separators available for selection are the equal sign (=), colon (:), and dash (-).

Conversion Settings

- Edit code options
  - User defined edit codes are not supported in Webfacing
  - However, you can map them to one of the available system edit codes
  - Each user-defined edit code can only have one mapping
  - The valid use-defined edit codes are 5-9
  - The valid system edit codes values are 1-4, A-D, J-Q, and W-Z



Edit code options

User-defined edit codes are not supported in the WebFacing Tool. However, you can map them to the available system edit codes. Select a pair of edit codes and add them to the table.

User-defined edit code: 5

System edit code:

User-defined	System
5	5

Use the **Command key recognition patterns** tab to specify how to detect the text for command key buttons to be displayed in a Web browser. When converted, command keys are displayed as buttons on your Web pages. Users click these buttons to launch an event defined in your DDS source or to get online help.

Use the **Command key button labels** tab to specify the default text for command keys identified during conversion that do not match a defined recognition pattern. For example, if no recognition pattern has been defined that would match F1=Help, you could specify the value for the **Command Key** field as F1 and the value for **Button label** as Help. When converted, command keys are displayed as 'Help' buttons on your Web pages. Users click these buttons to launch an event defined in your DDS source such as to refresh their page or to get online help.

If you have selected DDS members of type MNUDDS for conversion, use the **MNUDDS options** tab to convert the menu options into hypertext links. The command option number and any text included in the same field where that number is found will be converted into a hypertext link. The option of automatically converting commands into hypertext links is selected by default. In the **Separator** drop-down list, specify a separator to identify the menu options. The default separator is the period (.). The other separators available for selection are the equal sign (=), colon (:), and dash (-).

Runtime Settings – Web Application Settings

- To specify the IP address of the host where the native application runs
  - The Web application communicates with the WebFacing server: you also need to specify the IP port for the WebFacing server
- To specify whether the user gets prompted for password
  - Other settings: Web page height, Forced Deferred Write, Keyboard text entry mode: overwrite or insert
- Can be at project or link level

Host name The host name of the machine where your original application is located. This field cannot be blank.

Host port The host port that has been assigned for WebFacing communication on the iSeries server where your original application is located. By default, port 4004 is used.

Retain user ID and password for subsequent prompted signon To save the settings for user ID and password so that you have single signon for the session.

Prompt for signon To prompt users for a user ID and password when they try to access your application through a browser. If you use this option, the fields **User ID**, **Password**, and **Confirm password** must also be populated.

User ID The default User ID that will be used to connect to the machine where your original application is located.

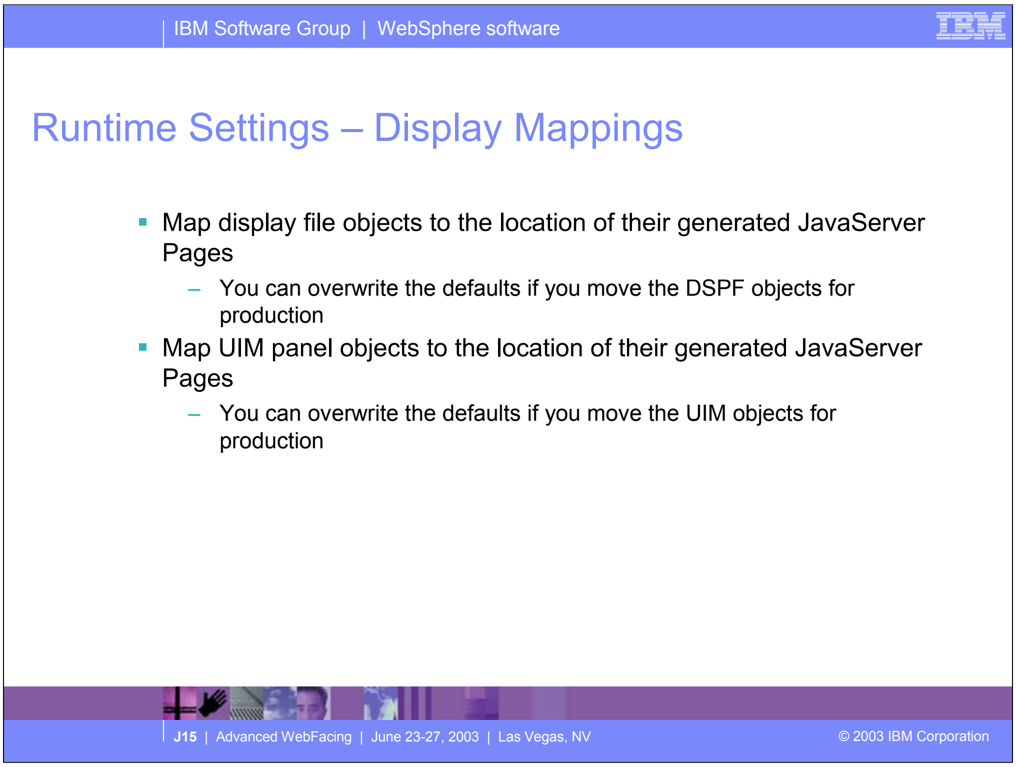
Password and Confirm password The password for the default User ID that will be used to connect to the machine where your original application is located. This password is encrypted and saved in the file wfapp.properties and is exported with your project when you set up a WebSphere Web application for your project.

Use user-defined signon page To provide your own signon page that you can modify before deploying the Web application. If you select it, the application will use your signon page for prompting if it is needed. Otherwise, the application uses a default signon page when prompting is needed.

Use fixed height for the converted Web page Ensures that each row in the converted web page is the same height in pixels. You might want to select this option if you experience problems with row alignment. For example, screens that are composed of CLRL, KEEP, and ASSUME records can sometimes have problems with alignment when converted for WebFacing. Note that this setting will increase the overall height of your converted web page.

Handle as if DDS files are compiled with DFRWRT(*YES) The DFRWRT keyword controls the writing of records to the display device (for a WebFacing application, the display device is the browser). In 5250 applications, DFRWRT is sometimes set to *NO. This means that all records are immediately written to the display device with no apparent delay. However, in a converted WebFacing application, the *NO parameter can cause significant delays. Using the property **Handle as if DDS files are compiled with DFRWRT(*YES)** , the default for WebFacing will be to defer writing. That is, WebFacing writes only when the application reads a record or when the application writes a record that has the FRCDTA keyword. You can disable this property in the properties page for a specified CL command or in the properties page for your project. The setting for the CL command overrides the setting for the project.

Use insert mode for keyboard text entry This option controls the insert mode for text entry on input fields in the browser. In a typical Web application, insert mode is on. In a typical 5250 session, insert mode is off (that is, the keyboard is in overwrite mode). By default, in a converted WebFacing application, insert mode is off, so it behaves like the original green-screen application. This allows a user to enter data without selecting or deleting text.



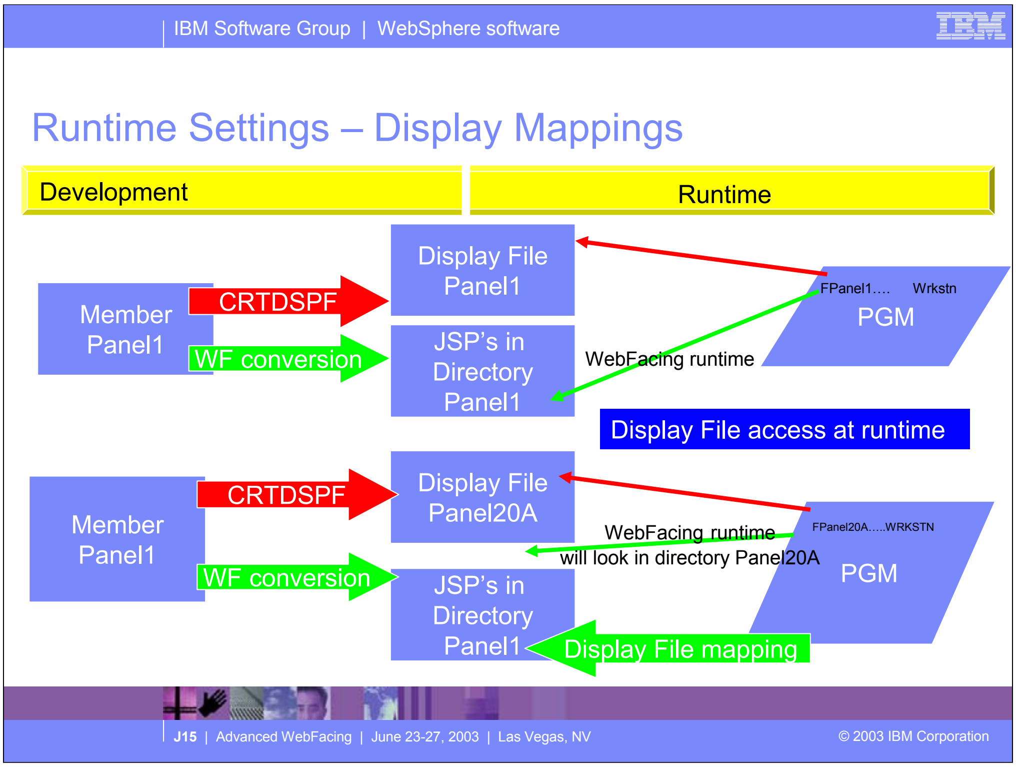
Runtime Settings – Display Mappings

- Map display file objects to the location of their generated JavaServer Pages
  - You can overwrite the defaults if you move the DSPF objects for production
- Map UIM panel objects to the location of their generated JavaServer Pages
  - You can overwrite the defaults if you move the UIM objects for production

Values for DDS object mapping are stored in the file DSPFObjectMapping.properties. Information about object mappings is available in the comments area of the corresponding screen. Mapping information is displayed in the editable text area of the screen. If you want to edit the .properties file, under the Navigator tab in the IDE, right-click the file and select **Open With** --> **Default Text Editor**.

UIM object mappings

Values for UIM object mapping are stored in the file UIMObjectMapping.properties. Information about UIM object mappings is available in the comments area of the UIM Object Mappings screen, and the actual mapping information is displayed in the editable text area of the screen. If you want to edit the UIMObjectMapping.properties file, under the Navigator tab in the IDE, right-click the file and select **Open With** --> **Default Text Editor**.



IBM Software Group | WebSphere software

Runtime Settings – Display Mappings

Development

Runtime

Member Panel1

CRTDSPF

WF conversion

Display File Panel1

JSP's in Directory Panel1

WebFacing runtime

FPanel1.... Wrkstn

PGM

Display File access at runtime

Member Panel1

CRTDSPF

WF conversion

Display File Panel20A

JSP's in Directory Panel1

WebFacing runtime will look in directory Panel20A

FPanel20A.....WRKSTN

PGM

Display File mapping

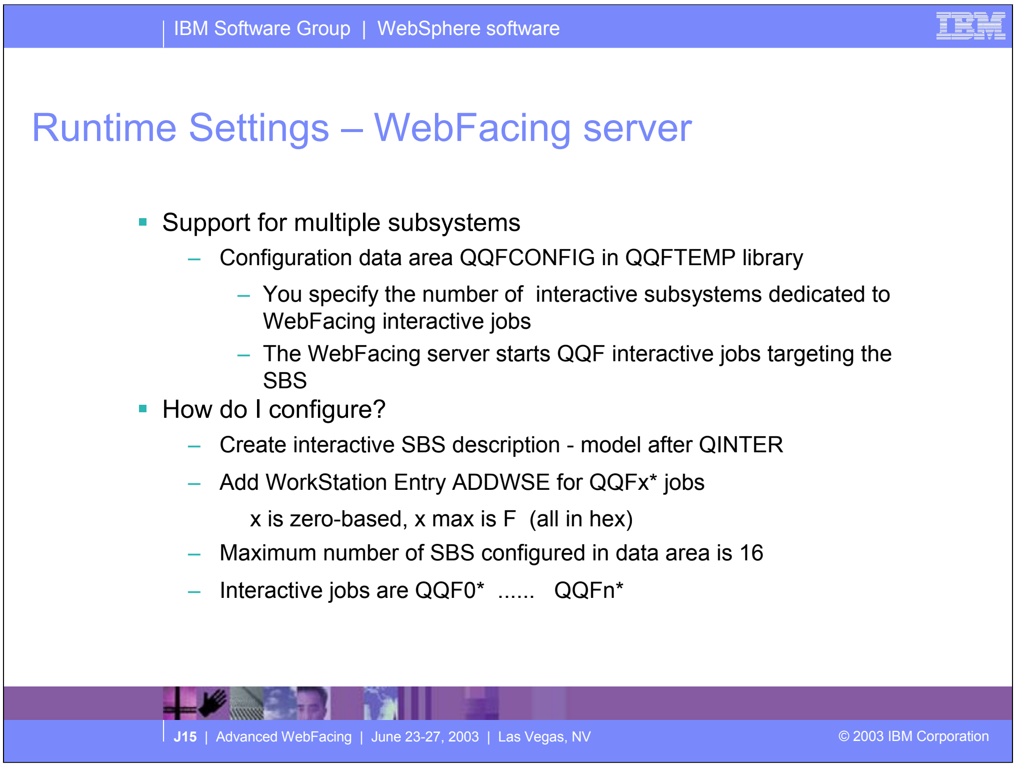
J15 | Advanced WebFacing | June 23-27, 2003 | Las Vegas, NV

© 2003 IBM Corporation

Values for DDS object mapping are stored in the file DSPFObjectMapping.properties. Information about object mappings is available in the comments area of the corresponding screen. Mapping information is displayed in the editable text area of the screen. If you want to edit the .properties file, under the Navigator tab in the IDE, right-click the file and select **Open With** --> **Default Text Editor**.

UIM object mappings

Values for UIM object mapping are stored in the file UIMObjectMapping.properties. Information about UIM object mappings is available in the comments area of the UIM Object Mappings screen, and the actual mapping information is displayed in the editable text area of the screen. If you want to edit the UIMObjectMapping.properties file, under the Navigator tab in the IDE, right-click the file and select **Open With** --> **Default Text Editor**.



Runtime Settings – WebFacing server

- Support for multiple subsystems
  - Configuration data area QQFCONFIG in QQFTEMP library
    - You specify the number of interactive subsystems dedicated to WebFacing interactive jobs
    - The WebFacing server starts QQF interactive jobs targeting the SBS
- How do I configure?
  - Create interactive SBS description - model after QINTER
  - Add WorkStation Entry ADDWSE for QQFx* jobs

    x is zero-based, x max is F (all in hex)
  - Maximum number of SBS configured in data area is 16
  - Interactive jobs are QQF0* ......   QQFn*

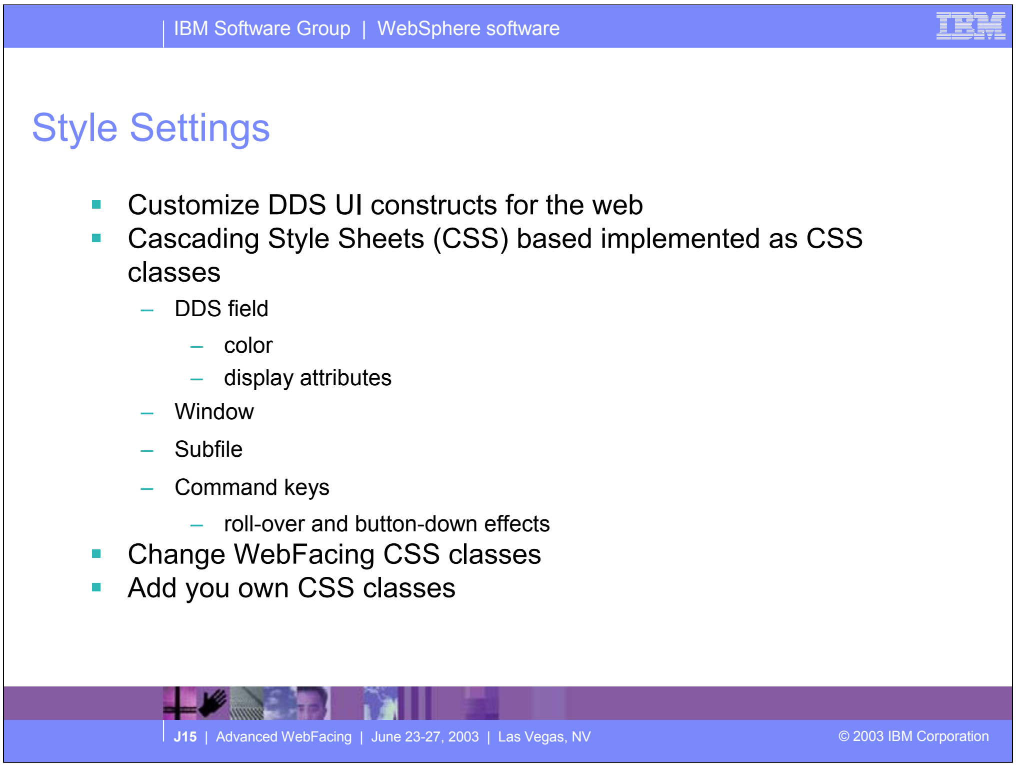
Depending on the number of users that you have and the number of WebFacing jobs that are created on your system, you may want to configure additional interactive subsystems to handle WebFacing jobs. Using multiple interactive subsystems improves the scalability of WebFacing by:

Increasing the total number of WebFacing jobs that can be run on your machine.

Load balancing: WebFacing jobs are assigned in a distributed manner so that they are spread optimally among configured subsystems.

Up to 16 additional subsystems can be configured for WebFacing. If no other subsystems are configured, by default, WebFacing jobs run in the QINTER subsystem. Assess the need for using additional subsystems based on your knowledge of how many users you have and the capabilities of your hardware. Keep in mind that an interactive job is created for each user logon whether the logon is through a 5250 session or a WebFacing session. Note, when additional subsystems are configured for WebFacing, QINTER is no longer used for WebFacing interactive jobs.

WebFacing interactive jobs are named by using a device name convention QQFn* where n represents which subsystem is being used in a sequence of up to 16. The value for n follows hex numbering conventions. In hex, the first 10 values for n are represented with the numbers 0 through 9 and the next six values (values 10 to 15) are represented with the letters A through F. For example, names for jobs in the first subsystem will begin with QQF0*, names for jobs in the twelfth subsystem will begin with QQFB*. If no additional subsystems are configured, WebFacing interactive jobs can be identified in the QINTER subsystem by names beggining with the string QQF0*.



Style Settings

- Customize DDS UI constructs for the web
- Cascading Style Sheets (CSS) based implemented as CSS classes
  - DDS field
    - color
    - display attributes
  - Window
  - Subfile
  - Command keys
    - roll-over and button-down effects
- Change WebFacing CSS classes
- Add you own CSS classes

Use the DDS field color screen to indicate how you want DDS field colors to be treated when they are converted for Web use. For example, you can specify that a blue DDS field be displayed as some other color on the Web. You can also control the text and background color for your fields when you use the reverse image display attribute.

Use the DDS field display attributes screen to indicate how you want DDS field display attributes to be treated when they are converted for Web use. For example, you can specify that a blinking DDS field be displayed as italic font on the Web.

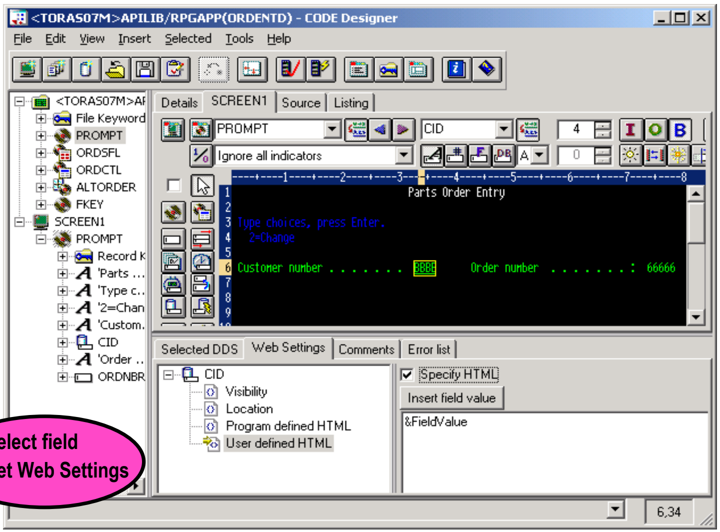
Use the Window screen to indicate how you would like DDS WINDOW records to look when they are converted for Web use. You can customize the look of the title, body, and shadow of the window.

Use the Subfile screen to indicate how the scroll bar and rows that are used to view a Subfile record will look after conversion.

Use the Command keys screen to specify how the text and buttons for command keys defined in your DDS source will display in a Web browser. When converted, command keys are displayed as buttons on your Web pages. Users click these buttons to launch an event defined in your DDS source such as to refresh their page or to get online help.

If you prefer to edit the Cascading Style Sheet (CSS) files directly, you can use the CSS editor supplied with the workbench.

WebSettings: Designer



```
                                Parts Order Entry

 Type choices, press Enter.
   2=Change

 Customer number . . . . . . . BBBB      Order number . . . . . . . :  66666
```

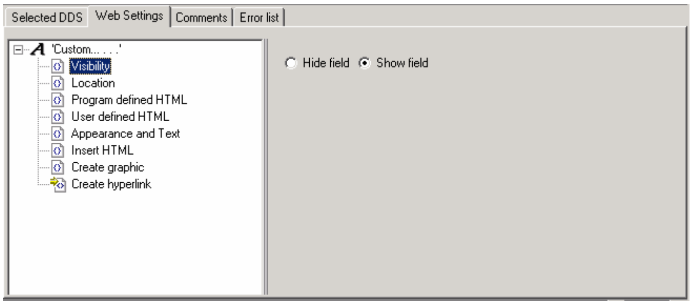
☑ Specify HTML

&FieldValue

1. Select field
2. Set Web Settings

CODE Designer allows you to design DDS screens graphically. In CODE Designer, each DDS object is represented by an icon or push button. For example, there are push buttons so that you can easily create named fields or text constants for your DDS screen. The Web Settings available for each DDS object vary depending on the object that you are working with. The online help lists the Web Settings available for each DDS object. In the online help refer to the section of this document *Web Setting descriptions* for more details on each setting. When you use Web Settings for an object, special comments are added to your DDS source which later get processed by the WebFacing conversion. Web Setting comments begin with the characters *%%WB.

Show / Hide Field on Web



○ Hide field ● Show field

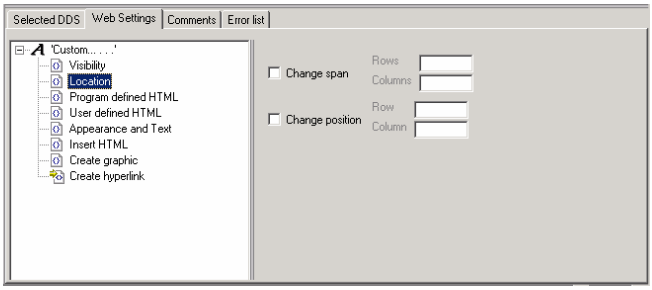
Options available for the Visibility setting are **Hide Field** or **Show Field**. If **Show Field** is chosen, the field will be displayed on your Web pages.

Change Field position and size

- Change span: to indicate the number of rows and columns that will be given to your field.

  Use: using a graphic in place of the text values that the program originally used to populate the field.

- Change position: to indicate the row and column that will be the starting position for your field.



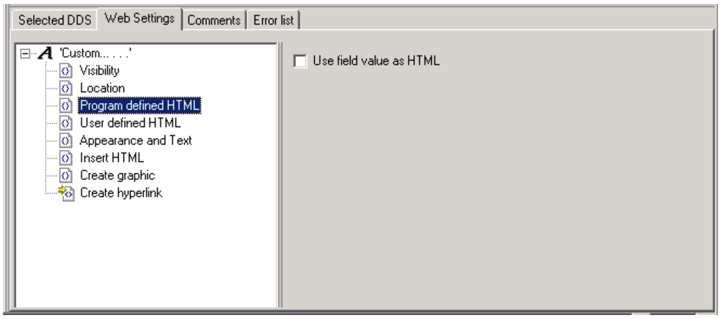
□ Change span Rows Columns

□ Change position Row Column

Options available for the Location setting are **Change span** and **Change position**. After your DDS source has been converted with the WebFacing Tool, your application is positioned on a Web page using an HTML table. The HTML table has the same number of rows and columns as your DDS screen -- that is, 24 by 80 or 27 by 132. With Location settings, you can change how your fields will be positioned on the HTML table after your source is converted.

Use field contents as HTML

- Use field value returned by the field as HTML markup that you would like to use in your page



Selected DDS | Web Settings | Comments | Error list

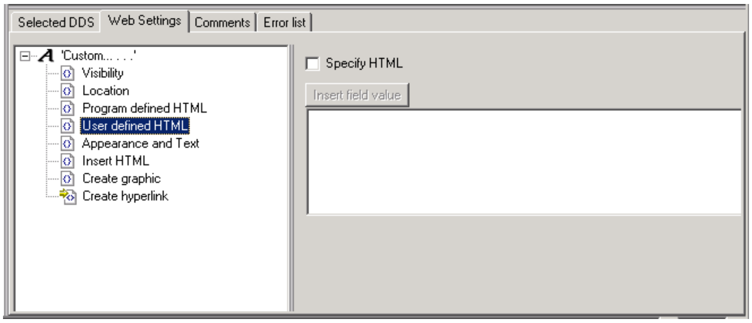
'Custom... . . .'
- Visibility
- Location
- Program defined HTML
- User defined HTML
- Appearance and Text
- Insert HTML
- Create graphic
- Create hyperlink

□ Use field value as HTML

With the Program defined HTML setting, check the box **Use field value as HTML** if the value returned by the field is HTML markup that you would like to use in your page.

Replace field with HTML

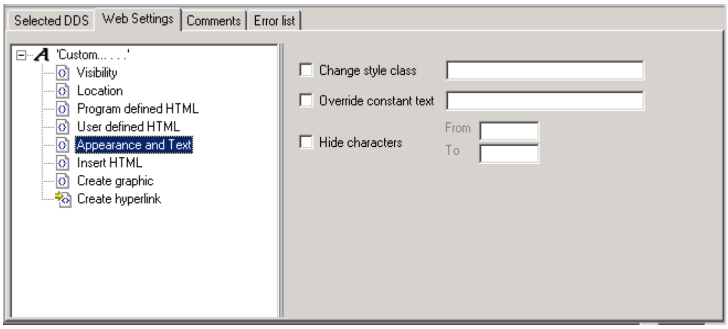
- Specify HTML text or tags that can be displayed with or in place of the value for your field.



User defined HTML allows you to specify HTML text or tags that can be displayed with or in place of the value for your field. To work with this setting, select the **Specify HTML** checkbox. In the entry area, enter the HTML tags you would like to use. If you want to use the value for your field as part of the HTML, click the button **Insert field value**. One example of how you could use this setting would be if you wanted the value for your field to be displayed with a bold font. To do this, use the <BOLD> tag with the &{FIELD.value} text.With this example, edit the entry area so that it displays <BOLD>&{FIELD.value}</BOLD>. If the code for &{FIELD.value} is not there by default, it can be added by clicking the **Insert field value** button.

Override Field on Web

- Overrides constant text (e.g field description)
- Overrides field value
- Hides specified characters



□ Change style class

□ Override constant text

□ Hide characters From To

Options available for the Appearance and Text setting are **Change style class**, **Override field value**, **Override constant text**, and **Hide characters**:

Change style class allows you to specify a style sheet class to be used with the field. As an example, you may want to display all numeric data with a particular font or color and you have defined a style sheet class numeric for this purpose. If you had a customer number field in your program such as custno, you could select the **Change style class** checkbox and enter the name for your numeric style sheet class.

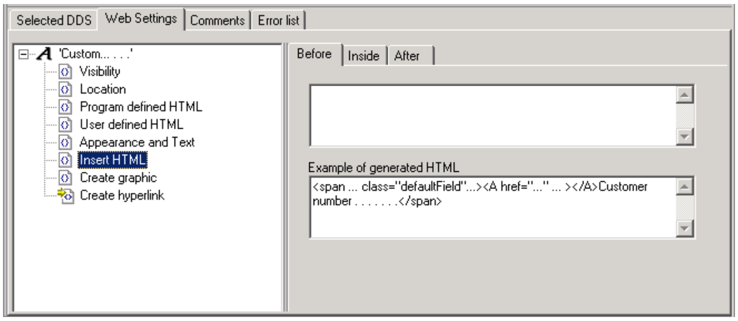
Override field value allows you to specify text that you want to display in place of the returned value for your field.

Override constant text allows you to specify text that you want to display in place the value set for a text constant.

Hide characters allows you to indicate a range of characters in your field that you do not want to display. For example, if your screen displays a menu of options for a user to choose from and each option is listed numerically (such as: **1. User tasks** - **2. Office tasks**), you might want to hide the first two characters so that only the text for the option is displayed. In an example like this, you would choose **Hide characters** and specify a range of From 1 To 2.

Customize field with HTML

- To customize your fields by adding additional HTML Before, Inside, and After the HTML tag that is automatically generated for the field.



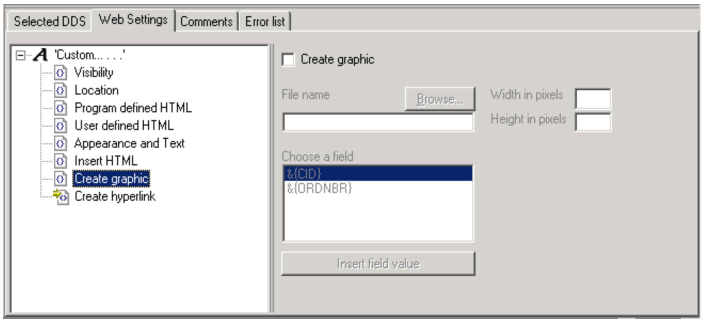
Example of generated HTML

```
<span ... class="defaultField"...><A href="..." ... ></A>Customer
number . . . . . . .</span>
```

The **Insert HTML** setting provides a way to customize your fields by adding additional HTML. HTML can optionally be added **Before**, **Inside**, and **After** the HTML tag that is automatically generated for the field. As an example, you might have an input field for employee numbers and at your company all employee numbers are prefixed with 64-.

Replace field with image

- Populate the File name field with the complete URL to an image file (e.g. http://www.ibm.com/image.gif), or with a field value



□ Create graphic

File name:

Width in pixels:

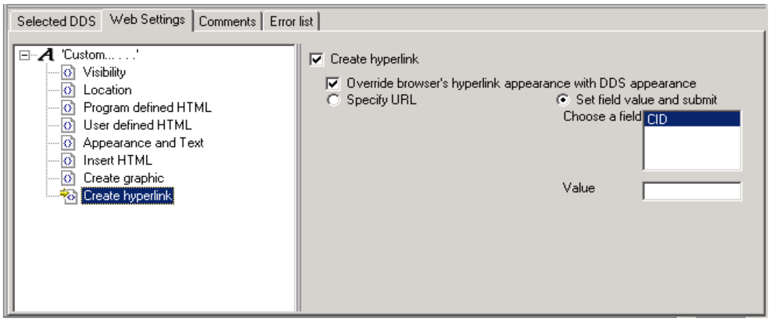
Height in pixels:

Choose a field: &{CID} &{ORDNBR}

Use the Create graphic setting if you want to display an image in place of your field. An example use for this setting would be a product database that includes images of inventoried items.

Display field as hyperlink

- Applies to output fields only
- Maps link to a field
- On click, submits specified value



- 'Custom... . . .'
  - Visibility
  - Location
  - Program defined HTML
  - User defined HTML
  - Appearance and Text
  - Insert HTML
  - Create graphic
  - Create hyperlink

☑ Create hyperlink

☑ Override browser's hyperlink appearance with DDS appearance

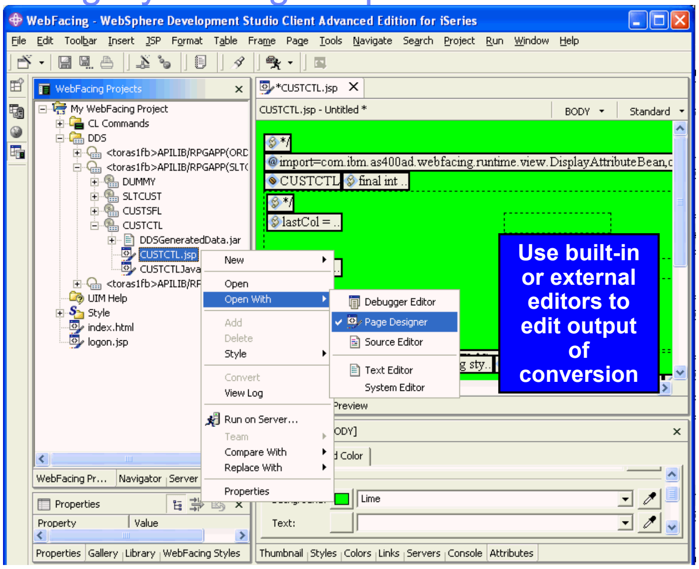
○ Specify URL ● Set field value and submit

Choose a field: CID

Value:

Use the Create hyperlink setting so that your field is displayed as a hyperlink. There are three different approaches available for creating hyperlinks. They are: **Specify static URL**, **JavaScript hyperlink**, and **Action hyperlink**. If you want to override the browser's settings for hyperlink appearance, select the checkbox **Override browser's hyperlink appearance with DDS appearance**. If you use the **Create hyperlink** setting along with the **Create Graphic** setting, your field can be displayed as a graphic that also acts as a hyperlink.

3 Customizing By Editing Output



Use built-in or external editors to edit output of conversion

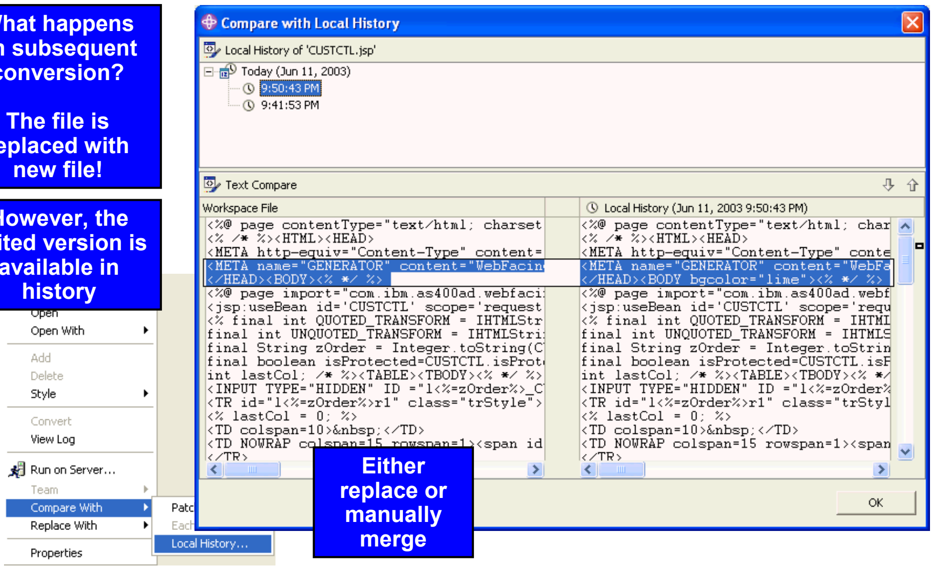
If you want to edit the layout and the frame surrounding the application area and command keys, edit the user-defined CSS files in the \chrome directory. To use the IDE's CSS editor, right-click the file that you want to edit and choose **Open with --> CSS Designer**. If you want to edit the layout of the frame, edit the file PageBuilder.jsp. To edit the frame style, right-click **Style --> Edit Style**. PageBuilder.jsp can then be edited using the Page Designer tool supplied by the IDE. The CSS file in the \chrome directory can be edited using the CSS Designer in the IDE. The Web perspective can be useful when editing CSS files. To open the Web perspective, select **Window --> Open Perspective --> Other --> Web**.

3 Customizing By Editing Output

What happens on subsequent conversion?

The file is replaced with new file!

However, the edited version is available in history



Workspace File

```
<%@ page contentType="text/html; charset
<% /* %><HTML><HEAD>
<META http-equiv="Content-Type" content=
<META name="GENERATOR" content="WebFacin
</HEAD><BODY><% */ %>
<%@ page import="com.ibm.as400ad.webfaci
<jsp:useBean id='CUSTCTL' scope='request
<% final int QUOTED_TRANSFORM = IHTMLStr
final int UNQUOTED_TRANSFORM = IHTMLStri
final String zOrder = Integer.toString(C
final boolean isProtected=CUSTCTL.isProt
int lastCol; /* %><TABLE><TBODY><% */ %>
<INPUT TYPE="HIDDEN" ID ="l<%=zOrder%>_C
<TR id="l<%=zOrder%>r1" class="trStyle">
<% lastCol = 0; %>
<TD colspan=10>&nbsp;</TD>
<TD NOWRAP colspan=15 rowspan=1><span id
</TR>
```

Local History (Jun 11, 2003 9:50:43 PM)

```
<%@ page contentType="text/html; char
<% /* %><HTML><HEAD>
<META http-equiv="Content-Type" conte
<META name="GENERATOR" content="WebFa
</HEAD><BODY bgcolor="lime"><% */ %>
<%@ page import="com.ibm.as400ad.webf
<jsp:useBean id='CUSTCTL' scope='requ
<% final int QUOTED_TRANSFORM = IHTMl
final int UNQUOTED_TRANSFORM = IHTMLS
final String zOrder = Integer.toStrin
final boolean isProtected=CUSTCTL.isF
int lastCol; /* %><TABLE><TBODY><% */
<INPUT TYPE="HIDDEN" ID ="l<%=zOrder%
<TR id="l<%=zOrder%>r1" class="trStyl
<% lastCol = 0; %>
<TD colspan=10>&nbsp;</TD>
<TD NOWRAP colspan=15 rowspan=1><span
</TR>
```

Either replace or manually merge

Changes to the styles made through the property pages or by direct editing apply only to the current project. To save them for use in other projects, right-click **Style** --> **Save as** and give your style a name. Then this named style becomes available for selection the next time you choose a Web style during project creation or the next time you select a style to replace the current project style using right-click **Style** --> **Select Style**.

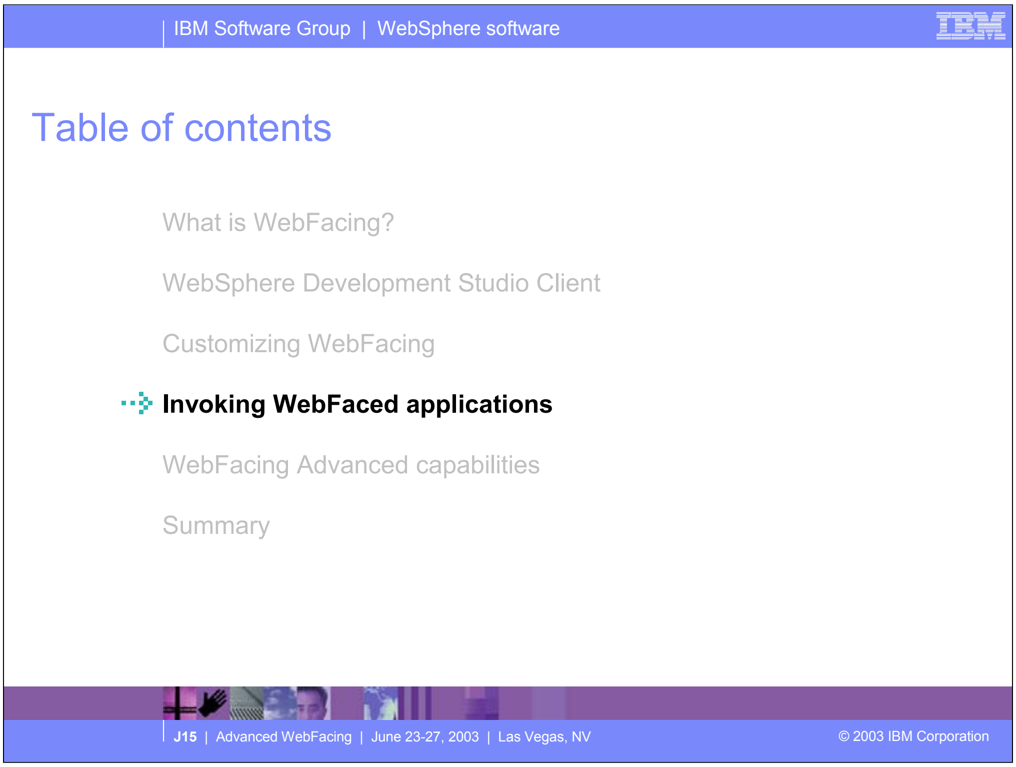


Table of contents

What is WebFacing?

WebSphere Development Studio Client

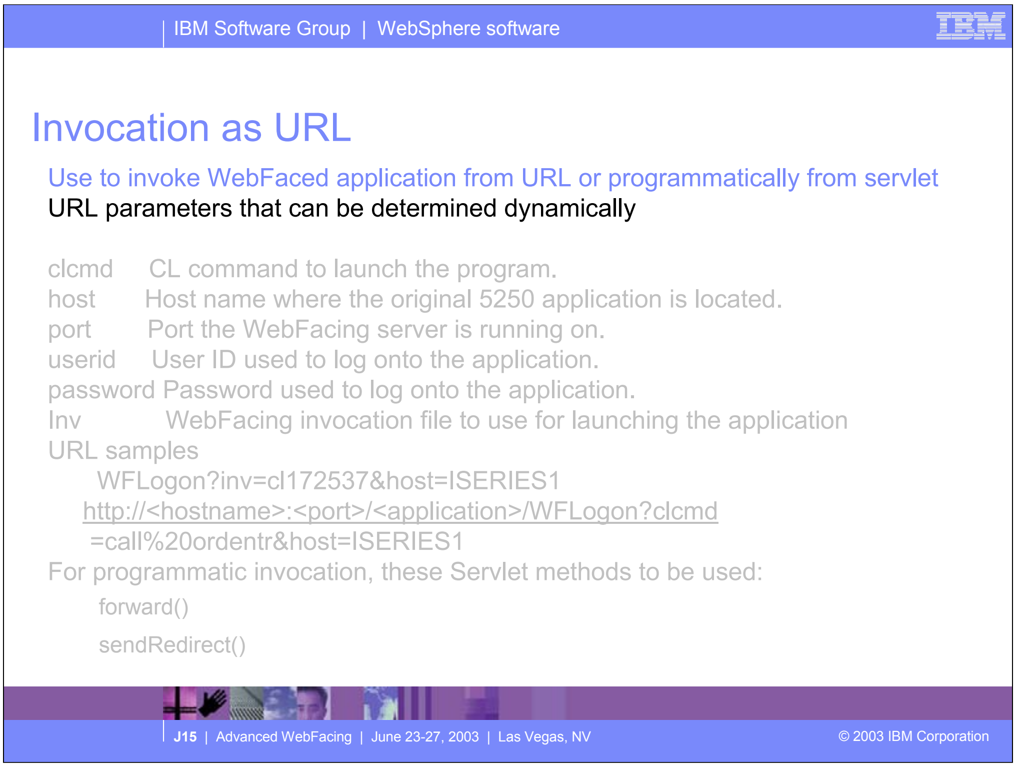
Customizing WebFacing

Invoking WebFaced applications

WebFacing Advanced capabilities

Summary

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.



Invocation as URL

Use to invoke WebFaced application from URL or programmatically from servlet

URL parameters that can be determined dynamically

clcmd CL command to launch the program.
host Host name where the original 5250 application is located.
port Port the WebFacing server is running on.
userid User ID used to log onto the application.
password Password used to log onto the application.
Inv WebFacing invocation file to use for launching the application

URL samples

WFLogon?inv=cl172537&host=ISERIES1

http://<hostname>:<port>/<application>/WFLogon?clcmd
=call%20ordentr&host=ISERIES1

For programmatic invocation, these Servlet methods to be used:

forward()

sendRedirect()

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.

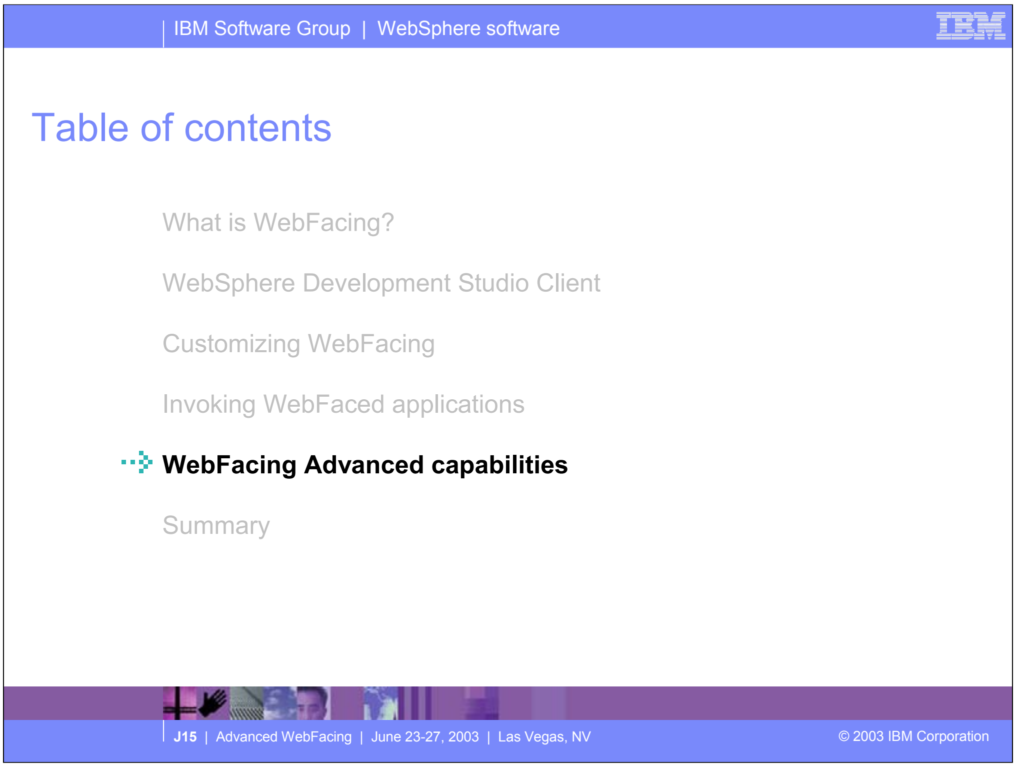


Table of contents

What is WebFacing?

WebSphere Development Studio Client

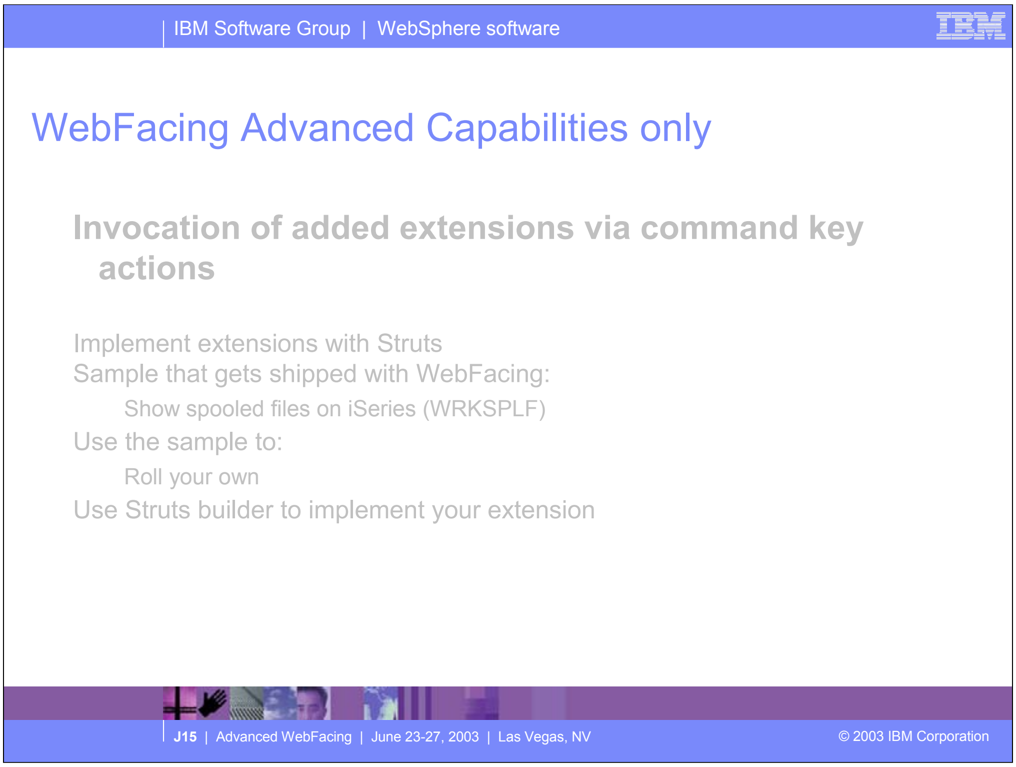
Customizing WebFacing

Invoking WebFaced applications

WebFacing Advanced capabilities

Summary

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.



WebFacing Advanced Capabilities only

Invocation of added extensions via command key actions

Implement extensions with Struts

Sample that gets shipped with WebFacing:

- Show spooled files on iSeries (WRKSPLF)

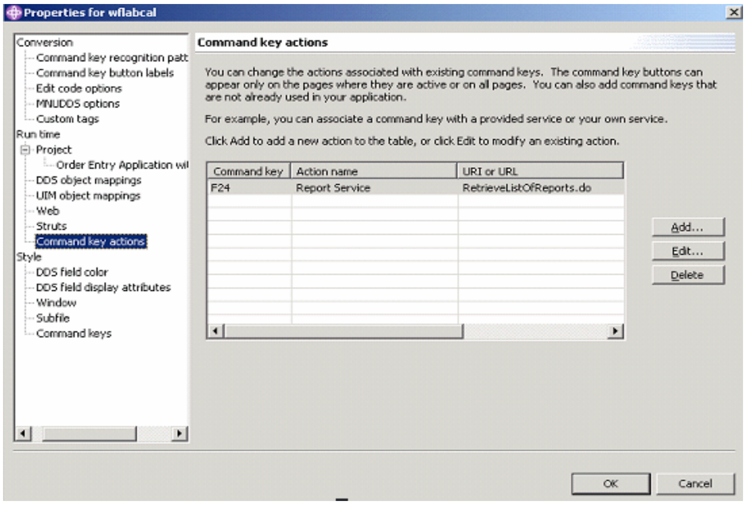
Use the sample to:

- Roll your own

Use Struts builder to implement your extension

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.

Add command key for WebFacing Service



Command key actions

You can change the actions associated with existing command keys. The command key buttons can appear only on the pages where they are active or on all pages. You can also add command keys that are not already used in your application.

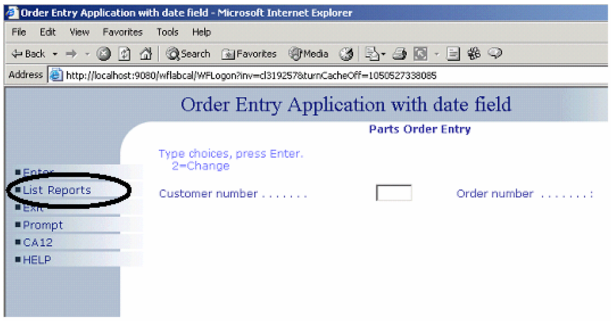
For example, you can associate a command key with a provided service or your own service.

Click Add to add a new action to the table, or click Edit to modify an existing action.

Command key	Action name	URI or URL
F24	Report Service	RetrieveListOfReports.do

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.

Function key button added to page


Order Entry Application with date field - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Enter
List Reports
Exit
Prompt
CA12
HELP

http://localhost:9080/wflabcal/WFLogon?inv=cl3192578turnCacheOff=1050527338085

Order Entry Application with date field

Parts Order Entry

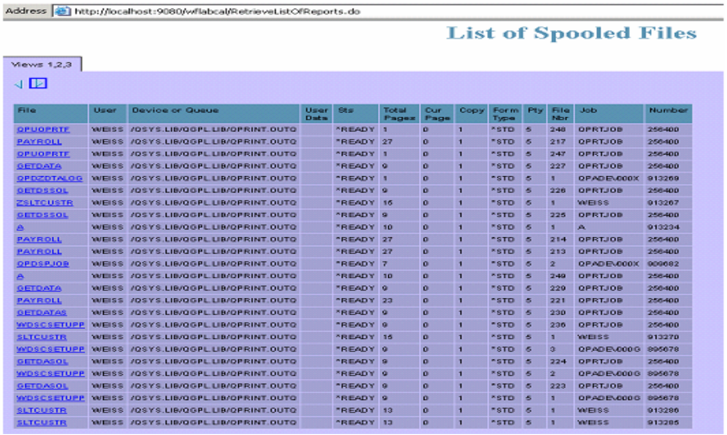
Type choices, press Enter.
2=Change

Customer number . . . .   Order number . . . . :

- Warning does not work from workbench browser

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.

Sample Report Service page



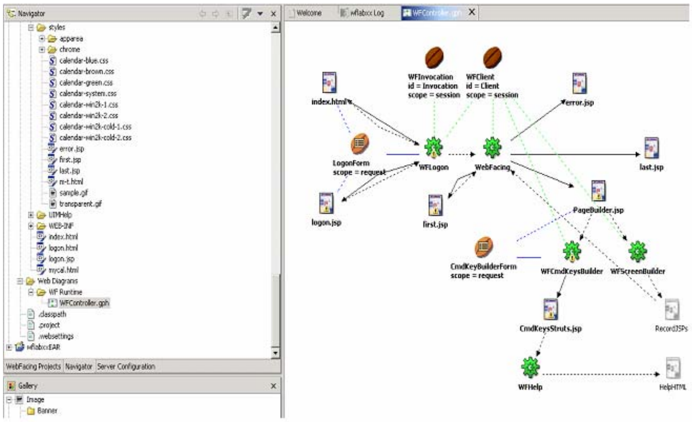
Address http://localhost:9080/wflabcal/RetrieveListOfReports.do

List of Spooled Files

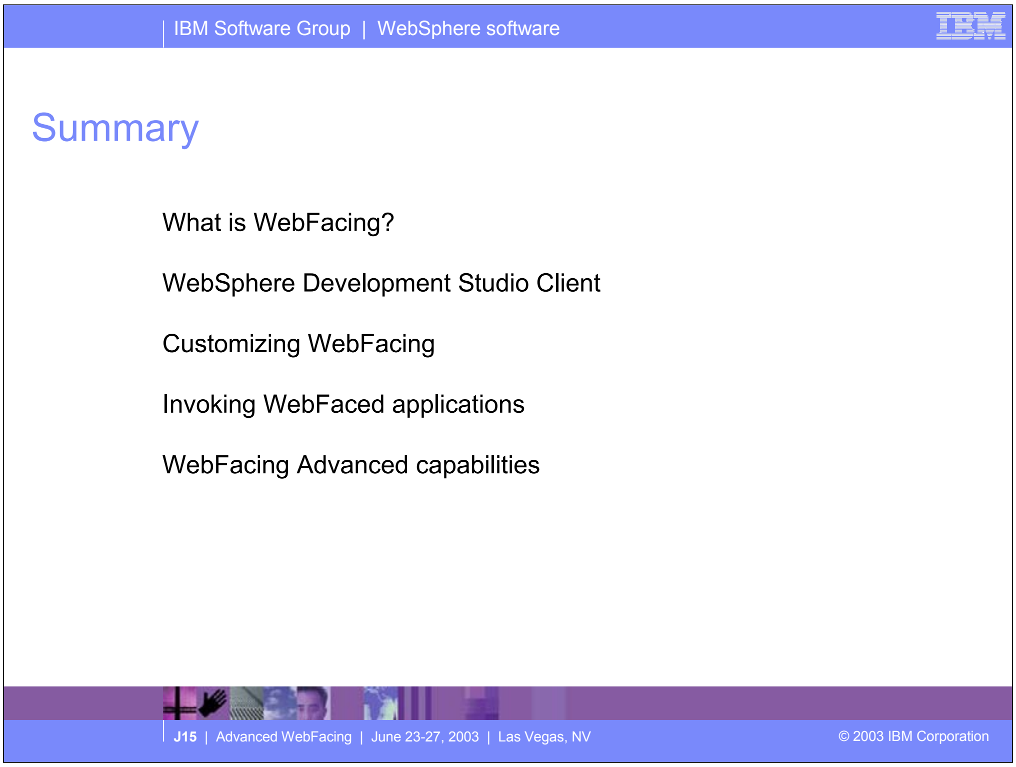
File	User	Device or Queue	User Data	Sts	Total Pages	Cur Page	Copy	Form Type	Pty	File Nbr	Job	Number
QPUOPRTF	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	1	0	1	*STD	5	248	QPRTJOB	256400
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	27	0	1	*STD	5	217	QPRTJOB	256400
QPUOPRTF	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	1	0	1	*STD	5	247	QPRTJOB	256400
GETDATA	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	227	QPRTJOB	256400
QPDZDTALOG	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	1	0	1	*STD	5	1	QPADEV000X	913269
GETDSSOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	226	QPRTJOB	256400
ZSLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	16	0	1	*STD	5	1	WEISS	913267
GETDSSOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	225	QPRTJOB	256400
A	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	10	0	1	*STD	5	1	A	913234
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	27	0	1	*STD	5	214	QPRTJOB	256400
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	27	0	1	*STD	5	213	QPRTJOB	256400
QPDSPJOB	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	7	0	1	*STD	5	2	QPADEV000X	909682
A	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	10	0	1	*STD	5	249	QPRTJOB	256400
GETDATA	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	229	QPRTJOB	256400
PAYROLL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	23	0	1	*STD	5	221	QPRTJOB	256400
GETDATAS	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	230	QPRTJOB	256400
WDSCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	236	QPRTJOB	256400
SLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	16	0	1	*STD	5	1	WEISS	913270
WDSCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	3	QPADEV000G	895678
GETDASOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	224	QPRTJOB	256400
WDSCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	2	QPADEV000G	895678
GETDASOL	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	223	QPRTJOB	256400
WDSCSETUPP	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	9	0	1	*STD	5	1	QPADEV000G	895678
SLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	13	0	1	*STD	5	1	WEISS	913286
SLTCUSTR	WEISS	/QSYS.LIB/QGPL.LIB/QPRINT.OUTQ		*READY	13	0	1	*STD	5	1	WEISS	913285

In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.

Struts diagram



In the second section of this presentation we introduced Development Studio Client. Now WebFacing customization is described.



Summary

What is WebFacing?

WebSphere Development Studio Client

Customizing WebFacing

Invoking WebFaced applications

WebFacing Advanced capabilities

In this presentation we introduced the IBM WebFacing Tool and the latest version of IBM WebSphere Development Studio Client for iSeries. Then we reviewed the features you can use to customize the results of a Webfacing application produced through the IBM WebFacing Tool.

More WebFacing information

Understanding the IBM WebFacing Tool

By Claus Weiss and Emily Bruner
ISBN 1-931182-09-4

IBM Press

Available now

IBM Press
Illustrated Guide Series
IBM
Understanding the IBM WebFacing Tool
A Guided Tour
• Based on WebSphere Development Studio Client V5 with hints for V4 users
• Learn how to use the WebFacing tool by converting a sample application
• Learn about Web settings and how to use them to enhance the WebFaced user interface
• Learn about WebFacing properties and how to use them to tailor WebFacing results
• Learn how to publish the results of the WebFacing conversion to WebSphere Application Server V3.5.6, 4, and 5
Claus Weiss

This book, the first in the IBM Press Guided Tour series, describes how to get started with IBM's new WebFacing tool and how to make progress with WebFacing efforts. The systematic approach presented here takes readers through WebFacing a sample green screen application using IBM WebSphere Development Studio Client. It also demonstrates how to use the different features in the tool to enhance the application after the initial conversion. Even debugging strategies are discussed. To maximize the speed and completeness of your learning experience, the author takes a practical, hands-on approach to explaining WebFacing features and how to use them. This approach, gained from involvement in defining the capabilities of the WebFacing tool allows you to benefit from both the step-by-step nature of the book as well as the most comprehensive treatment offered on WebFacing training. The book's many exercises will teach you a variety of features in the WebFacing tool and the WebSphere Development Studio Client product. The relationship between these products will also be explained as well as the reason for each of the various exercises.

More WebFacing information

IBM Redbook
The IBM WebFacing Tool

By Aleksandr Nartowitch
Dean Henkel
Steve O'Shannessy
Barry Owen
SG24-6801-00

IBM

Available now

IBM eserver

IBM

The IBM WebFacing Tool

Converting 5250 Applications to Browser-based GUIs

Follow this complete guide and easy examples to use the IBM WebFacing Tool

Web enable existing host applications on the IBM eserver iSeries server

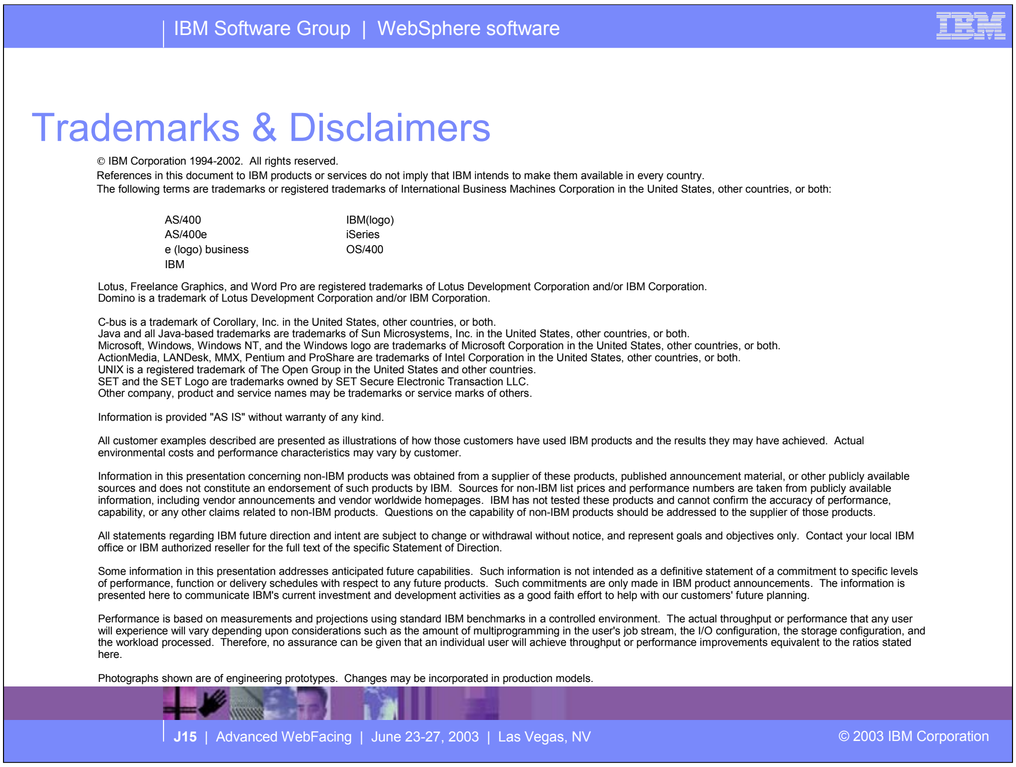
Use existing skills to create a Web interface

Aleksandr Nartovich
Dean Henkel
Steve O'Shannessy
Barry Owen

Redbooks

ibm.com/redbooks

This redbook explains how the application conversion works and how you can customize your Web interface. It explains how to work with JavaServer Pages and cascading style sheets. It also offers helpful performance considerations and troubleshooting tips. Plus it looks at how to deploy WebFaced applications to Apache Tomcat. This redbook is written for iSeries application developers who want an easy way to extend the life of existing host applications. It also applies to those who want to expand the reach of existing host applications to a wider set of clients.



Trademarks & Disclaimers

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

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

AS/400
AS/400e
e (logo) business
IBM
IBM(logo)
iSeries
OS/400

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.
Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.
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 trademarks of Microsoft Corporation in the United States, other countries, or both.
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.
SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
Other company, product and service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. 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.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Legal

Disclaimer

- Acknowledgment:

  This presentation is a collaborative effort of the IBM Toronto iSeries Application Development presentation team, including work done by:

  Satish Gungabeesoon, Maha Masri,

- Disclaimer:

  The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

- Reproduction:

  The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

IBM WebFacing Tool in IBM WebSphere Development Studio Client for iSeries

iSeries Application Development Team: IBM Toronto

WebSphere. software

e business software