



IBM Software Group

Session 430279

IBM WebFacing Tool in WebSphere Development Studio Client:  
Advanced Topics

Satish Gungabeesoon ([satish@ca.ibm.com](mailto:satish@ca.ibm.com))  
iSeries Application Development Team: IBM Toronto

**WebSphere.** software



Spring 2006 | Advanced Topics - IBM WebFacing

© 2006 IBM Corporation

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?**

What is WDHT?

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

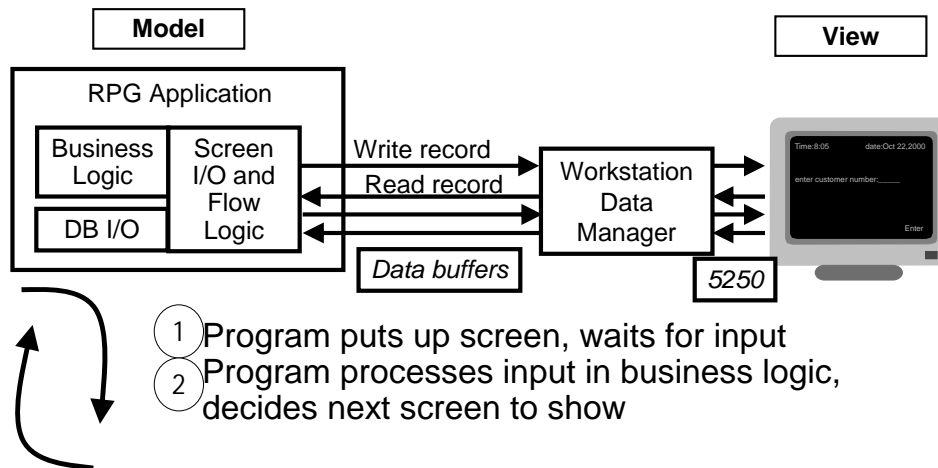
- What it is:
  - Development Tool
    - Converts DSPFs to Web UI
    - Application converted at development time, not at runtime
  - Runtime Intercept
    - \*PGM data fed to Web UI
- What is not:
  - 5250 datastream converter!
    - No 5250 datastream ever created
    - Unless using WDHT for unconverted Display Files (New in V 6.0.1)
- 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.

## iSeries Classic Program Model

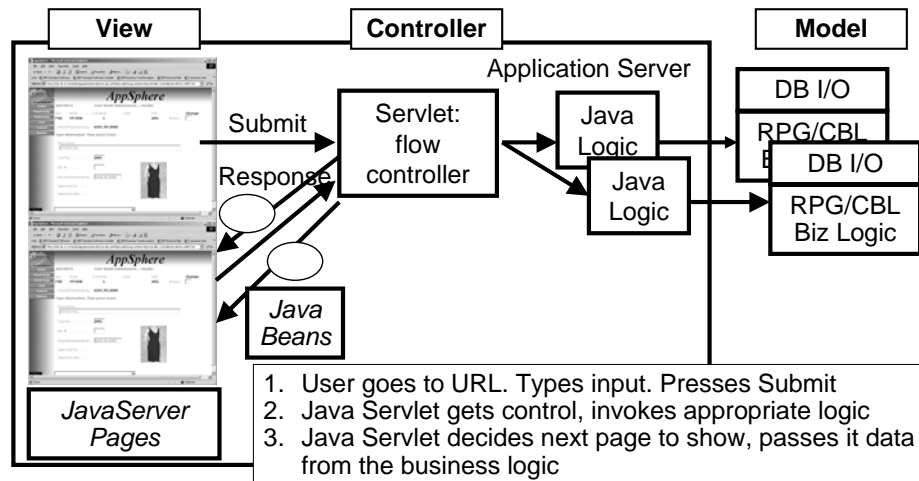


© 2006 IBM Corporation

In the green-screen environment, the application performs READs and WRITEs to the workstation. The application data is sent to Workstation Data Manager. The Workstation Data Manager merges application data with the display file. The Workstation Data Manager generates a 5250 datastream that is sent to the display.

In this model, you typically have one model object and multiple views on that object. A view is a window onto the model. Presentation logic typically goes in the view. Business logic goes in the model.

## Web Application Program Model



© 2006 IBM Corporation

The e-business program model follows the Model-View-Controller paradigm, organizing the application into three separate components:

- Model: the application model with corresponding data representation and business logic
- View: data presentation, providing views for user input
- Controller: to dispatch requests and control data flow

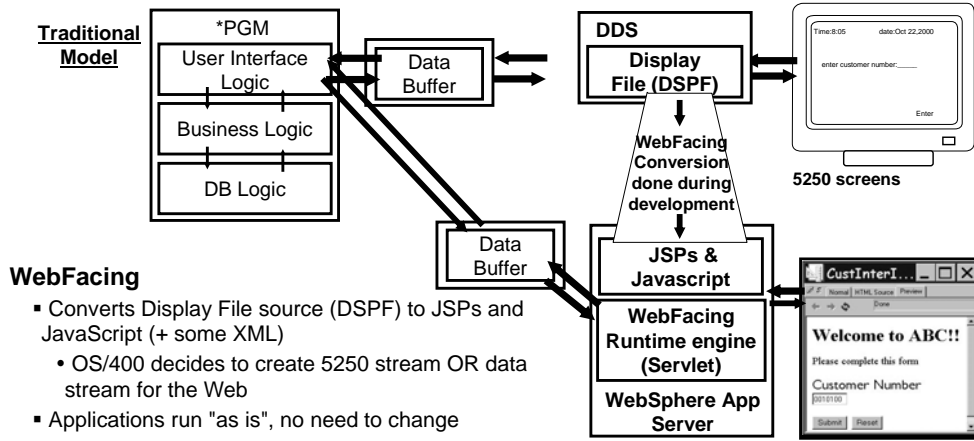
Here you can see the controller is added. The controller handles the interactions between the view and the model. When the model changes it updates the view, when the user does something with the view the controller informs the model.

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 controller.

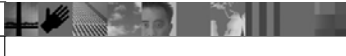
Because the job was started by the WebFacing server, the controller knows this is a WebFacing request and passes the data and control to the WebFacing server which runs on the iSeries. Control returns to the Webfacing runtime servlet that runs in WebSphere Application Server. The WebFacing servlet locates the appropriate JSPs and Java beans. The WebFacing 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.



## Better User Interface: IBM WebFacing Tool

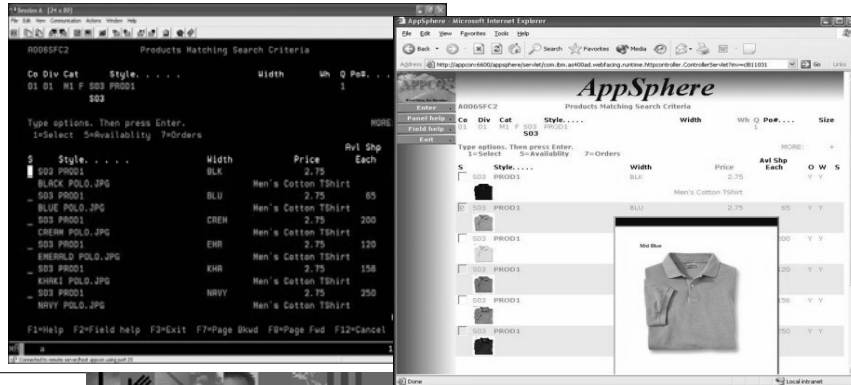


**Quickly web-enable many applications**



## The IBM WebFacing Tool

- ✓ Part of WebSphere Development Studio client
- ✓ No user enablement license required
- ✓ Eliminates interactive (OLTP) workload for 5250 applications
- ✓ Requires DDS source for screen formats in refaced applications
- ✓ Creates permanent Web artifacts which can be customized further



(Screen Conversion Thought)

Explain - supports both 5250 & web interface?

## Table of contents

- ▣ What is WebFacing?
- What is WDHT?**
- Customizing WebFacing
- Invoking WebFaced applications
- WebFacing Advanced capabilities
- Summary



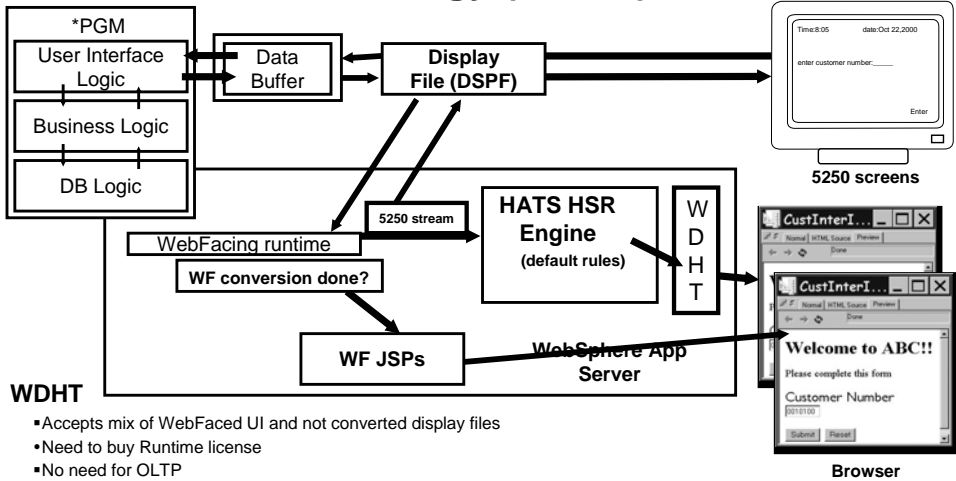
In the first section of this presentation we introduced WebFacing. Next we introduce Development Studio Client.





# IBM WebFacing Deployment Tool with HATS technology (WDHT)

## Traditional Model



## WDHT

- Accepts mix of WebFaced UI and not converted display files
- Need to buy Runtime license
- No need for OLTP

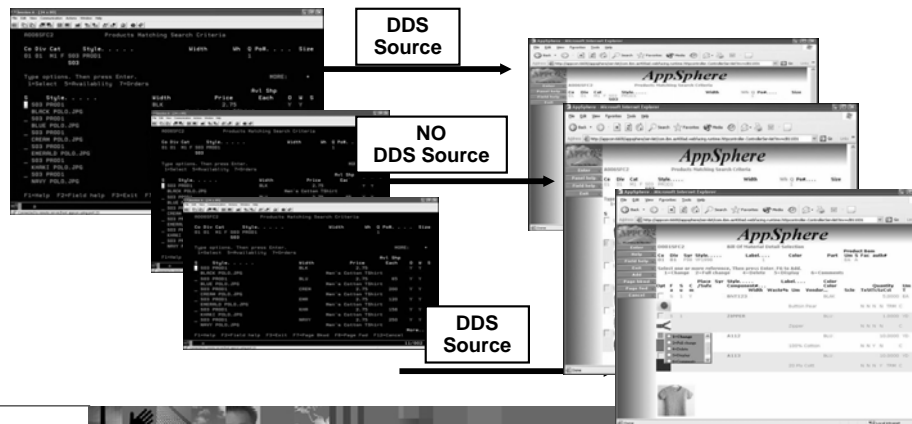
Quickly web-enable many applications, **mix of WebFaced and non WebFaced UI** allowed

No need to install HATS toolkit for WDHT enablement in WDSvc workbench




## IBM WebFacing Deployment Tool with HATS technology (WDHT)

- Accepts mix of WebFaced UI and not converted display files
- Need to buy Runtime license
- No need for OLTP



(Screen Conversion Thought)

Explain - supports both 5250 & web interface?

IBM Software Group | WebSphere software 

# WebSphere Development Studio

Current 5722-WDS customers with software subscription for V5R3, to upgrade to WDS V6.0 use feature #: 2656  
Available after GA

**Upgrade from WDS 6.0 to 6.0.1 using Rational Product updater**

*Unlimited Licenses*

**New** WDS Lite  
Technology preview

iSeries	iSeries	iSeries	iSeries	Web Facing WDHT support	iSeries Projects	+CODE +VisualAge RPG	
Java™	Debug	Struts Web	Web Service		RSE		
JSF	EGL Java generation	Trace	Profiling	DB	XML	App Server	HATS Toolkit

[www.ibm.com/software/awdtools/iseries](http://www.ibm.com/software/awdtools/iseries)

WebSphere Development Studio Client V6.0.1 based on RWD V6


© 2006 IBM Corporation

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.

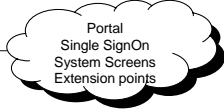

Development Studio has been a huge success, with over 80,000 licenses sold. Just as every development machine used to have PDM and SEU, every development machine will now have all the modern Application Development tools from IBM. This ubiquity is especially important for business partners who build 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.

IBM Software Group | WebSphere software 

## WebSphere Development Studio Client Advanced Edition 6.0.1

*Workstation License order through Passport Advantage* **Upgrade from WDSC 6.0 to 6.0.1 using Rational Product updater**

[http://www.lotus.com/services/passport.nsf/WebDocs/Passport\\_Advantage\\_Home](http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home)

iSeries	iSeries	iSeries *	iSeries	Web Facing * WDHT support	iSeries		
Java	Debug	Struts Web	Web Service	Projects RSE			
JSF	EGL Java generation	Trace	Profiling	DB	XML	App Server	HATS Toolkit
	EGL * COBOL generation	EJB * J2EE *	Test * Cases	Portal *			

[www.ibm.com/software/awdtools/iseries](http://www.ibm.com/software/awdtools/iseries)

WebSphere Development Studio Client V6.0.1 based on RAD V6

© 2006 IBM Corporation

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.

## Table of contents

- What is WebFacing?
- ▣▣▣▣ What is: WebFacing, HATS, and WDHT?
- 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.

## Customizing WebFacing

- WebFacing Properties
  - Conversion
  - Runtime
  - Style
- WebFacing WebSettings
- JSP customization using 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 specified 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 Properties and WebSettings


- Conversion
  - Conversion settings, specified in the project's properties page
- Runtime
  - Web application settings
  - iSeries host settings
    - specified in the project's properties page
  - WebFacing server settings
    - specified in the WebFacing server configuration area
- Styles
  - specified in the project's properties page
- WebSettings
  - Add conversion rules to DDS source

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.  
More details to come...
- **Command key button labels**  
To force the display of cmd key button descriptions for function keys that are defined in your DDS but not shown on 5250  
More details to come... 
- **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
- **XML Record Metadata**  
Allows to jar XML data buffer descriptions

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.

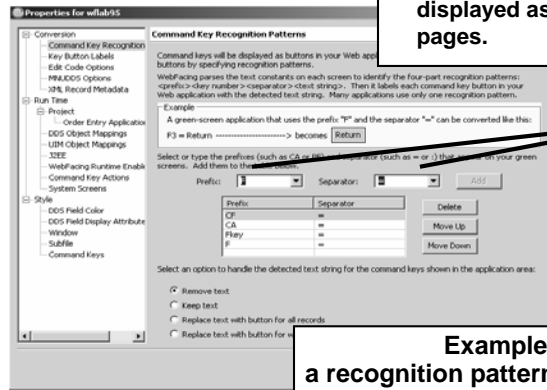
Use the **Command key button labels** tab to specify the default text for command keys identified during conversion that do not match a defined recognition pattern. For example, if no recognition pattern has been defined that would match F1=Help, you could specify the value for the **Command Key** field 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: Command Key Recognition

- 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.



Specify Prefix and Separator

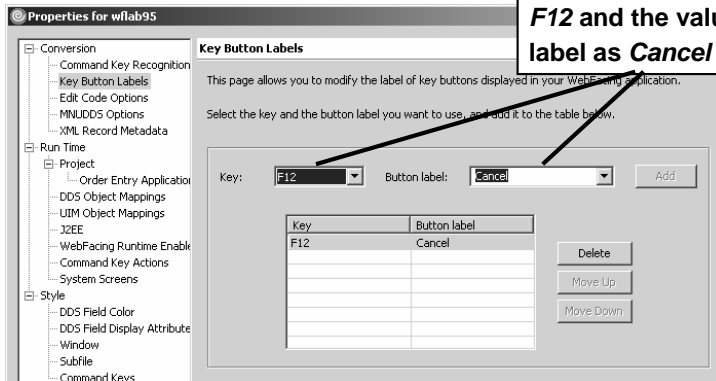
**Example:**  
a recognition pattern with a prefix F and separator = converts the source F4=Prompt to a button with the text Prompt

To edit the properties for a WebFacing project, in the WebFacing Projects view, right-click the icon for your Project, CL commands, DDS, UIM Help, or Style folder and select **Properties**. 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. 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: Key Button Labels

Specify the default text for keys identified during conversion that do not match a defined recognition pattern.

**Example:**  
if no recognition pattern has been defined that would match **F12=Cancel**, you could specify the value for the Key field as **F12** and the value for Button label as **Cancel**



This page allows you to modify the label of key buttons displayed in your WebFacing application.

Select the key and the button label you want to use, and add it to the table below.

Key:  Button label:

Key	Button label
F12	Cancel

To edit the properties for a WebFacing project, in the WebFacing Projects view, right-click the icon for your Project, CL commands, DDS, UIM Help, or Style folder and select **Properties**. 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. 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

- Edit code options

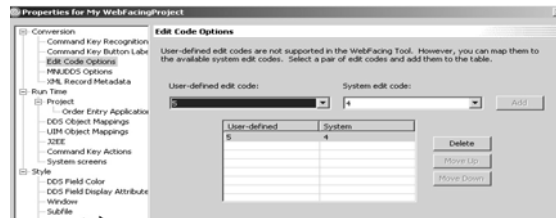
User defined edit codes are not supported in WebFacing

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 user-defined edit codes are 5-9

The valid system edit codes values are 1-4, A-D, J-Q, and W-Z



© 2006 IBM Corporation

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
    - Force Deferred Write
    - Default keyboard text entry mode: overwrite or insert

**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.

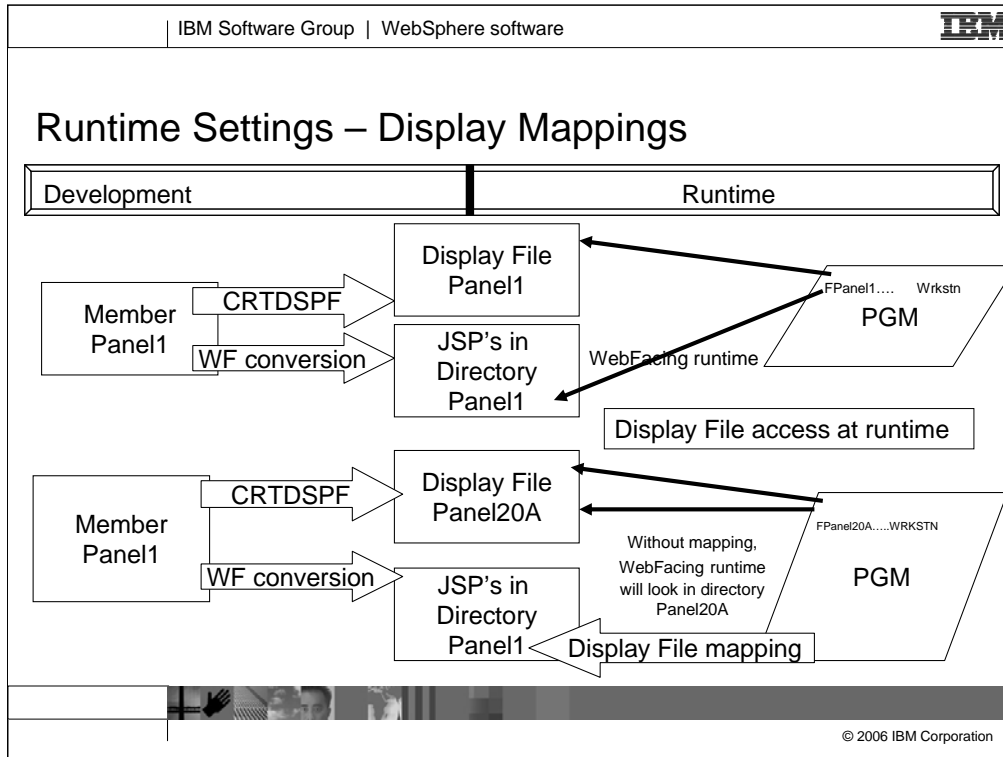
## Runtime Settings – Display Mappings

- 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**.



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 beginning 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.

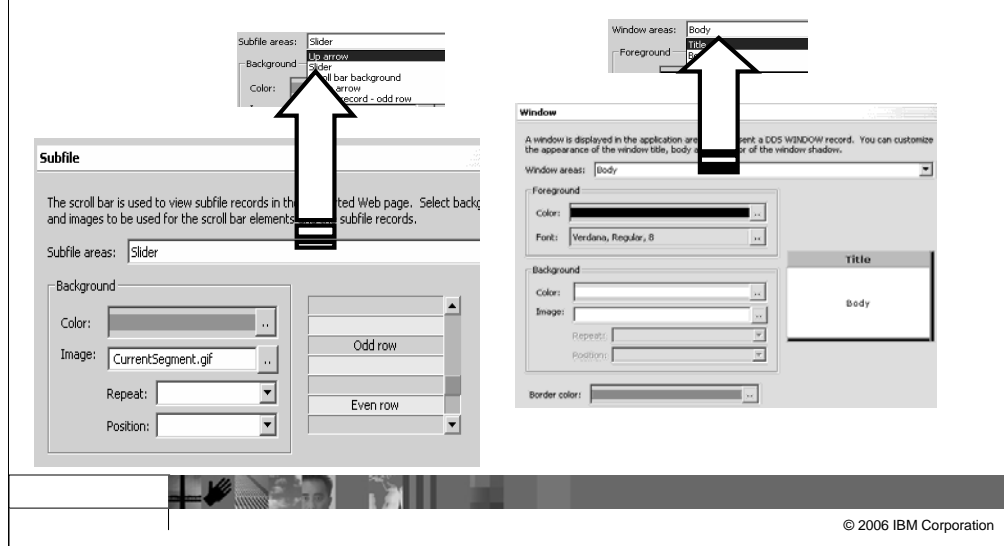
Use the Subfile screen to indicate how the scroll bar and rows that are used to view a Subfile record will look after conversion.

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.



## Style Settings: Window and Subfile examples.



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.

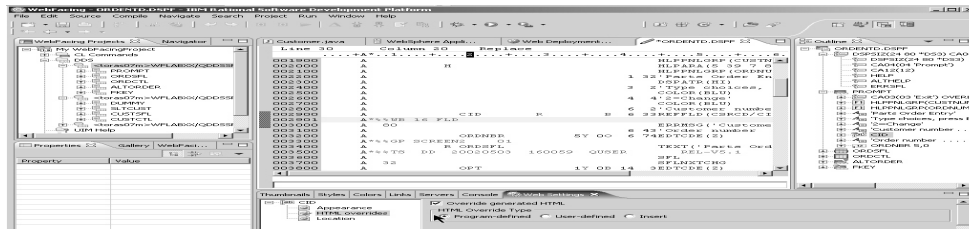
Use the Command keys screen to specify how the text and buttons for command keys defined in your DDS source will display in a Web browser. When converted, command keys are displayed as buttons on your Web pages. Users click these buttons to launch an event defined in your DDS source such as to refresh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

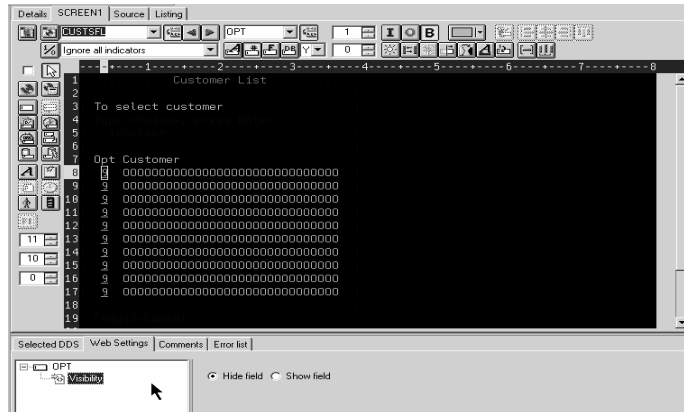


New in V6.0 create WebSettings directly in RSE or WebFacing perspective



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

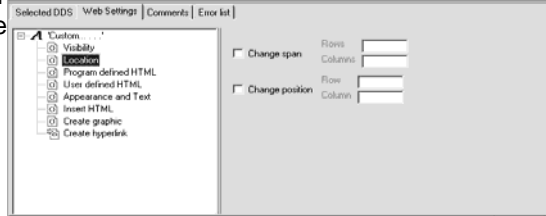


Options available for the Visibility setting are **Hide Field** or **Show Field**. If **Show Field** is chosen, the field will be displayed on your Web pages.

## 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



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.



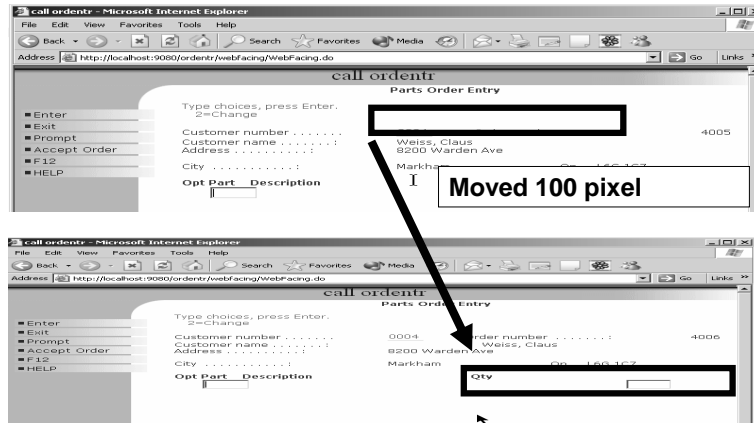
# Use CSS positioning (new in V6.0.0.1)

The screenshot displays the IBM WebSphere interface. The main window shows a table with columns for line numbers, codes, and text. A callout box titled "Want to move things around on the web page" points to a table element in the DOM tree on the right. Below the table, the "Web Settings" panel is open, showing the "Position" section with the "CSS position" option selected. The "Row" is set to 7 and the "Column" is set to 33. The "Offset" fields are set to 0 pixels for both horizontal and vertical offsets. The "Disable tabbing" option is unchecked.

Line	Code	Text
004500	A**%GP	SCREEN1 01
004600	A	
004700	A**%TS	SD 19971031 170141 A970107A
004800	A	
004900	A	
005000	A	
005100	A	
005200	A	
005300	A	
005400	A	
005500	A 40	
005600	A 41	
005700	A 44	
005800	A N49	
005900	A	
006000	A	CUSTOMER 30 7 2'Customer name . . . .
006001	A**%WB	21 FLD 7 +0   33 +100   false
006100	A	
006200	A	CADDR1 R 0 8 2'Address . . . . .
006300	A	CADDR2 R 0 9 33REFFLD(CSRCD/CADDR1 *LI
006400	A	
006500	A	CCITY R 0 10 2'City . . . . .
006600	A	
006700	A	CSTATE R 0 10 33REFFLD(CSRCD/CCITY *LI
006800	A	
006900	A	CZIP R 0 10 61REFFLD(CSRCD/CZIP *LIB
007000	A	
007100	A	
007200	A	PARTNR R B 13 DSPATR(HI)
007300	A 81	
007400	A 84	

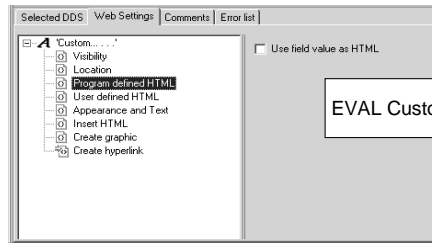
## Use CSS positioning (new in V6.0.0.1)

Before and after applying WebSetting



## Use field contents as HTML

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



- One example of how you could use this setting would be if your RPG program returned value of the CustomerName field with imbedded <H2> HTML tag.

```
EVAL CustomerName = '<H2>' + CustomerName + '</H2>'
```

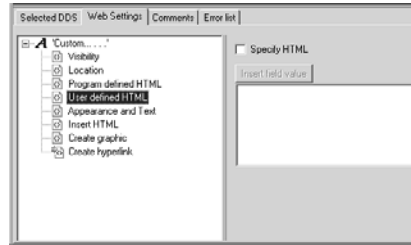
- One can use WebFacing environment API 'QqfEnvironment' to generate imbedded HTML only if RPG program runs in WebFacing 'mode'

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.

For example, in RPG you could say something like: `EVAL CustomerName = '<H2>' + CustomerName + '</H2>'`

## Replace field with HTML

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



- 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, but don't want RPG to imbed HTML markup into the field value.
- 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.

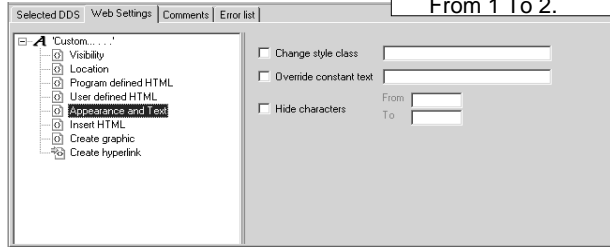
**User defined HTML** allows you to specify HTML text or tags that can be displayed with or in place of the value for your field. To work with this setting, select the **Specify HTML** checkbox. In the entry area, enter the HTML tags you would like to use. If you want to use the value for your field 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

- For example, if your screen displays a menu of options for a user to choose from and each option is listed numerically
  - 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**.



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.

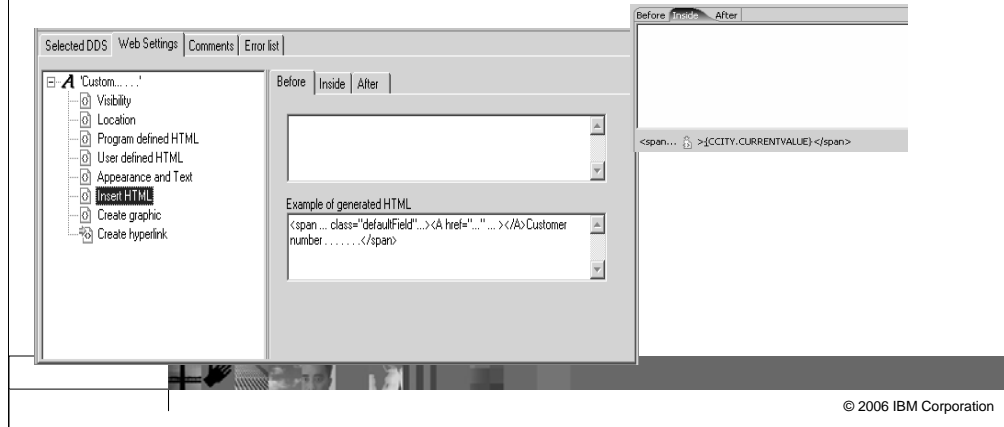
**Override field value** allows you to specify text that you want to display in place of the returned value for your field.

**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

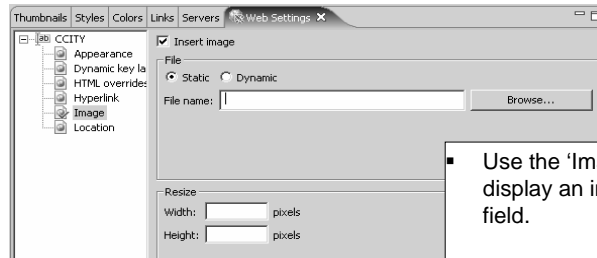
- To customize your fields by adding additional HTML:
    - Before (the HTML tag)
    - Inside (last attribute of the HTML tag)
    - After (the HTML tag)
- the HTML tag that is automatically generated for the field.



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



- Use the 'Image' 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.

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.

The screenshot displays a web application interface. At the top, there is a header with the IBM Software Group logo and the text 'WebSphere software'. Below the header is a table with the following columns: 'SELECT', 'ORDER NO', 'TOTAL PRICE', 'DATE', and 'STATUS'. The table contains 16 rows of data, with the first row having a 'SELECT' value of 'B' and an 'ORDER NO' of '6666666'. A text box on the right side of the table contains the following text: 'Applies to output fields only', 'Maps link to a field', and 'On click, submits specified value'. Below the table is a configuration panel with the following sections: 'Selected DDS', 'Web Settings', 'Comments', and 'Error list'. The 'Web Settings' section is expanded to show 'ORDERNO' with sub-items: 'Visibility', 'Location', 'Program defined HTML', 'User defined HTML', 'Appearance and Text', 'Insert HTML', 'Create graphic', 'Create hyperlink', and 'Dynamic key label'. The 'Create hyperlink' section is checked and has the following options: 'Override browser's hyperlink appearance with DDS appearance' (checked), 'Specify static URL' (radio button), 'JavaScript hyperlink' (radio button), and 'Action hyperlink' (radio button). The 'Action hyperlink' section has the following options: 'Position cursor to the field' (checked), 'Submit' (checked), 'Function key' (radio button), and 'JavaScript function call' (radio button). The 'Function key' is set to 'ENTER'.

SELECT	ORDER NO	TOTAL PRICE	DATE	STATUS
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0
B	6666666	-66,666.66	yyyy-mm-dd	0

Selected DDS | Web Settings | Comments | Error list

- ORDERNO
  - Visibility
  - Location
  - Program defined HTML
  - User defined HTML
  - Appearance and Text
  - Insert HTML
  - Create graphic
  - Create hyperlink
  - Dynamic key label

Create hyperlink

Override browser's hyperlink appearance with DDS appearance

Specify static URL

Choose a field: [ ] [Insert field value]

URL: [ ]

Target: [ ]

JavaScript hyperlink

Action hyperlink

Position cursor to the field

Submit

Function key: [ENTER]

Enter data: [ ]

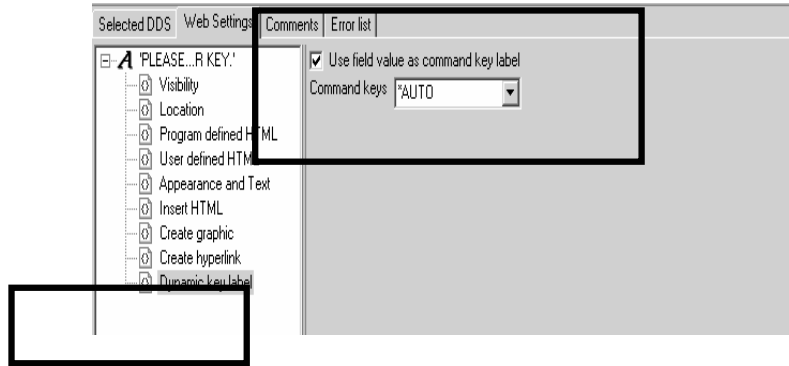
JavaScript function call: [ ]

© 2006 IBM Corporation

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.

## Dynamic key label Websetting

If your command key descriptions are supplied inside of fields coming from your program, use this Web Setting



## Record format Web Settings Fkey Label

The screenshot displays the IBM Code Designer interface for editing a record format. The main window shows a list of fields for a 'Customer List' record, with columns for field number, type, and value. A dialog box titled 'Web Settings' is open, showing a list of key labels: 'Key labels', 'Key order', and 'Insert into script'. The 'Key labels' option is selected. The dialog also includes a 'New Label' input field and 'Remove' and 'Change' buttons.

Field	Type	Value
1	S	00000000000000000000000000000000
2	S	00000000000000000000000000000000
3	S	00000000000000000000000000000000
4	S	00000000000000000000000000000000
5	S	00000000000000000000000000000000
6	S	00000000000000000000000000000000
7	S	00000000000000000000000000000000
8	S	00000000000000000000000000000000
9	S	00000000000000000000000000000000
10	S	00000000000000000000000000000000
11	S	00000000000000000000000000000000
12	S	00000000000000000000000000000000
13	S	00000000000000000000000000000000
14	S	00000000000000000000000000000000
15	S	00000000000000000000000000000000
16	S	00000000000000000000000000000000
17	S	00000000000000000000000000000000
18	S	00000000000000000000000000000000
19	S	00000000000000000000000000000000



# Record format Web Settings Key Order

The screenshot shows the IBM Code Designer interface for a record format. The main window displays a table of fields for the record format 'ORDCTL'. A dialog box titled 'Change order and visibility of command keys' is open in the foreground, allowing configuration of command and function keys.

Field	Posi...	Len...	Type	Shift	Usa...	Reference	Sample
Custom...	7,2	29	Text const...				Customer name ..
CUSTOMER	7,33	30	Alphanume...	A - Alphanum...	Out...		000000000000
Address...	8,2	29	Text const...				Address .....
CADDR1	8,33	20	Alphanume...	A - Alphanum...	Out...	CSRCD/CADDR1 WFLAB88/CST...	000000000000
CADDR2	9,33	20	Alphanume...	A - Alphanum...	Out...	CSRCD/CADDR2 WFLAB88/CST...	000000000000
City .....	10,2	29	Text const...				City .....
CCITY	10,33	20	Alphanume...	A - Alphanum...	Out...	CSRCD/CCITY WFLAB88/CSTM...	000000000000
CSTATE	10,56	2	Alphanume...	A - Alphanum...	Out...	CSRCD/CSTATE WFLAB88/CST...	00
CZIP	10,51	10	Alphanume...	A - Alphanum...	Out...	CSRCD/CZIP WFLAB88/CSTM...	0000000000
Opd Pa...	12,2	78	Text const...				Opd Pa... Descrip...
PARTNBR	13,6	6	Alphanume...	A - Alphanum...	Both	ITRCD/ID WFLAB88/ITEM	BBBBB
PARTDSC	13,14	24	Alphanume...	A - Alphanum...	Out...	ITRCD/NAME WFLAB88/ITEM	000000000000
PARTQTY	13,67	5,0	Numeric	Y - Numeric only	Both	STRCD/STQTY WFLAB88/STOCK	99999

Command keys	Hidden	Display order
CF06	<input type="checkbox"/>	
CA04	<input type="checkbox"/>	
CAT2	<input type="checkbox"/>	
Function keys	HELP	

## Key Label Precedence rules

- There are several ways you can define the label for a function key in your WebFacing applications.
- If more than one label is defined for a key, WebFacing conversion and run-time will use the following priority table to determine which label will appear in the browser.

Priority	Source of key label text	Scope of setting
1 (highest)	<b>Command Key Actions</b> in WebFacing project Run Time properties (Advanced edition)	Project
2	<b>Dynamic key labels</b> Web Setting	Field
3	<b>Key labels</b> Web Setting	Record
4	<b>Key labels</b> Web Setting	File
5	<b>Command Key Recognition Patterns</b> in WebFacing project Conversion properties Note: Function keys labels cannot be obtained using this method. Only command key text constants can be recognized in this way during conversion.	Record
6	DDS-defined "text" parameter for keyword	Record
7	DDS-defined "text" parameter for keyword	File
8 (lowest)	<b>Key Button Labels</b> in the WebFacing project Conversion properties	Project

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 specified 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



# Record format Web Settings Insert into script

The screenshot shows the IBM Code Designer interface for a project named '<ORR4S07M>-WFLAB88/QDD5SRC(ORDENTD)'. The main window displays a table of Record ORDCtl fields. A tree view on the left shows the project structure with 'ORDCTL' selected. Below the table, a 'Selected DDS' pane shows 'Web Settings' and 'Comments' tabs. A 'Scripting' window is open, showing a list of fields to be inserted into a script, with 'CUSTOMER' selected.

Field	Positi.	Len.	Type	Shift	Usa.	Reference	Sample
*Custom.....	7, 2	29	Text const...				Customer name .....
CUSTOMER	7, 33	30	Alphanum...	A - Alphanum...	Out...		00000000000000000000
*Address.....	8, 2	29	Text const...				Address .....
CADDR1	8, 33	20	Alphanum...	A - Alphanum...	Out...	CSRCD/CADDR1 WFLAB88/CST...	00000000000000000000
CADDR2	9, 33	20	Alphanum...	A - Alphanum...	Out...	CSRCD/CADDR2 WFLAB88/CST...	00000000000000000000
*City.....	10, 2	29	Text const...				City .....
CCITY	10, 33	20	Alphanum...	A - Alphanum...	Out...	CSRCD/CCITY WFLAB88/CSTMR	00000000000000000000
CSSTATE	10, 56	2	Alphanum...	A - Alphanum...	Out...	CSRCD/CSSTATE WFLAB88/CST...	00
CZIP	10, 61	10	Alphanum...	A - Alphanum...	Out...	CSRCD/CZIP WFLAB88/CSTMR	0000000000
*Opt Pa.....	12, 2	78	Text const...				Opt Part Description
PARTNBR	13, 6	6	Alphanum...	A - Alphanum...	Both	ITRCD/ID WFLAB88/ITEM	BBBBBB
PARTDSC	13, 14	24	Alphanum...	A - Alphanum...	Out...	ITRCD/NAME WFLAB88/ITEM	00000000000000000000
PARTQTY	13, 67	5, 0	Numeric	Y - Numeric only	Both	STRCD/STQTY WFLAB88/STOCK	99999

## Web Settings Source (1 of 5)

- Key labels (\*%%WB 11)

Specify text labels used in Web pages to represent keys

- Key order (\*%%WB 9)

Specify whether keys are visible on your Web pages and the order in which the keys will be displayed

- Display size (\*%%WB 10)

Change the size of HTML table used to display application in a browser

- Visibility (\*%%WB 4)

Hide or show field on Web pages



## Web Settings Source (2 of 5)

- **Location (\*%%WB 2,15)**

Change how fields are positioned on the HTML table after your source is converted

Change span allows you to indicate the number of rows and columns in the HTML table that are given to your field

Change position allows you to indicate the row and column in the HTML table that are the starting position for your field.

- **Program-defined HTML (\*%%WB 16)**

Use this if value returned by the field is HTML to be used in the page

- **User-defined HTML (\*%%WB 17)**

Allows you to specify HTML text or tags that can be displayed with or in place of the value for your field



## Web Settings Source (3 of 5)

- **Appearance and Text (\*%%WB 1,3,5)**

- Change style sheet class to use for field

- Specify text to display in place of the returned value for your field, or in place of a text constant

- Indicate range of characters in your field that you do not want to display

- **Insert HTML (\*%%WB 6,7,8)**

- Add HTML before, inside and after automatically generated HTML tag

- **Options for VALUES (\*%%WB 14)**

- Field specified with VALUES keyword, displayed on Web page as single choice checkbox

- Options allow customization of label or text displayed



## Web Settings Source (4 of 5)

- Create graphic (\*%%WB 13)

- Display an image in place of text field

- Create hyperlink (\*%%WB 12)

- Set field to display as hyperlink

- Static URL

- JavaScript Hyperlink

- JavaScript function to call when click link

- Action Hyperlink

- Position cursor to a field, submit a form, or combination of both

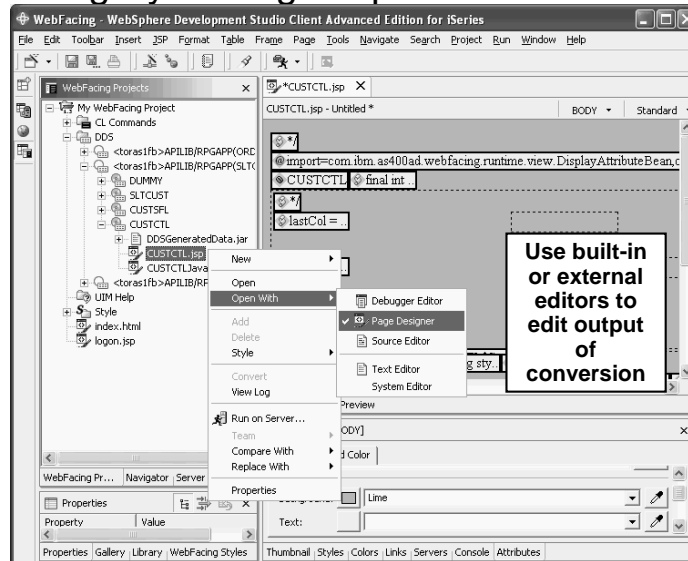


## Web Settings Source (5 of 5)

- **Insert into script (\*%%WB 18)**  
Customize the JSP files that are created for regular records or subfile control records
- **Send to browser (\*%%WB 19)**  
Can be used for H or hidden DDS fields  
Modifies the generated JSP files for your records so hidden fields are created
- **Dynamic key labels (\*%%WB 20)**  
Can be used for output fields  
Enables text for the command key to be determined by the runtime value of the field



### 3 Customizing By Editing Output



© 2006 IBM Corporation

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

Open  
Open With  
Add  
Delete  
Style  
Convert  
View Log  
Run on Server...  
Team  
Compare With  
Replace With  
Properties

Compare with Local History

Local History of 'CUSTCTL.jsp'

Today (Jun 11, 2003)

- 9:50:43 PM
- 9:41:53 PM

Text Compare

Workspace File	Local History (Jun 11, 2003 9:50:43 PM)
<%@ page contentType="text/html; charset <% /* %><HTML><HEAD> <META http-equiv="Content-Type" content=" <META name="GENERATOR" content="WebFacin </HEAD></BODY></% * %></%>	<%@ page contentType="text/html; char <% /* %><HTML><HEAD> <META http-equiv="Content-Type" conte <META name="GENERATOR" content="WebFes </HEAD></BODY bgcolor="lime"><% * %></%>
<%@ 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="1<%zOrder%>_C <TR id="1<%zOrder%>r1" class="trStyle"> <% lastCol = 0; %> <TD colspan=10>&nbsp;</TD> <TD NOWRAP colspan=15 rowspan=1><span id </TR>	<%@ 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="1<%zOrder%> <TR id="1<%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?

What is WDHT?



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.

## 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=INV1&host=My\_I5**
  - Use invocation file INV1**
  - Connect to server My\_I5**
  - & is delimiter between multiple parameters**
- **Sample URL**  
**[http://<hostname>:<port>/<application>/WFLogon?clcmd=call%20ordentr&host=My\\_I5](http://<hostname>:<port>/<application>/WFLogon?clcmd=call%20ordentr&host=My_I5)**
- For programmatic invocation, these Servlet methods to be used:
  - forward()** handled on server
  - sendRedirect()** send to browser and then request is sent back

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

## Table of contents

What is WebFacing?

What is: WebFacing, HATS, and WDHT?

Customizing WebFacing



Invoking WebFaced applications

**WebFacing Advanced capabilities**

Summary

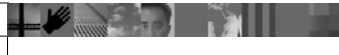


In the first section of this presentation we review WebFacing and what it is. Next we introduce Development Studio Client. Then customizing a WebFacing application is described.

## WebFacing Advanced Capabilities only

### **Invocation of added extensions via command key actions**

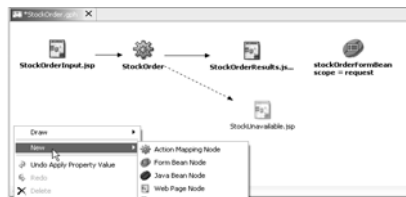
- Implement extensions with Struts
- Sample that gets shipped with WebFacing is:  
    Show spooled files on iSeries (WRKSPLF)
- Use the sample to:  
    Roll your own
- Use Struts builder to implement your extension



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

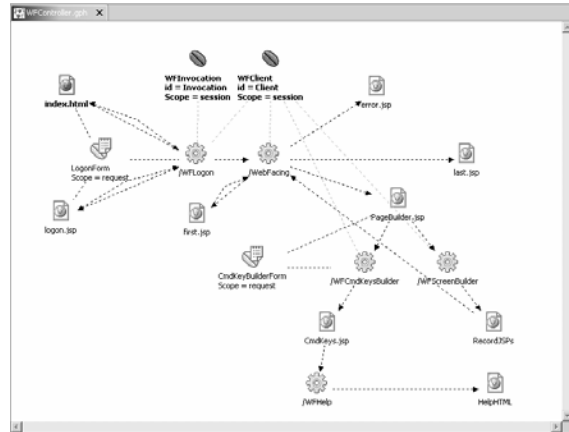
## WebFacing Tool – is based on Struts

- What is Struts
  - A framework for building web applications based on Model-View-Controller
- Why use it - Easier integration
  - Extension points can integrate with other Struts-based services
  - Dedicated diagram editor helps organize your web application

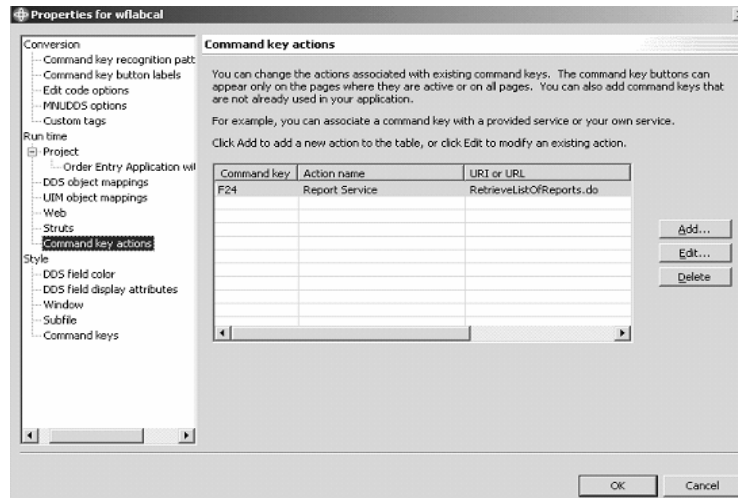


## WebFacing Tool - Struts Diagram

- Struts diagram helps understand structure
  - WebFacing diagram is read-only and provided for documentation
- Helps extends WebFacing
  - WRKSPFL first extension



## Add command key for WebFacing Service



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

## WebFacing Tool - Example

■ **WRKSPLF example:**

Our original screen has no reference to F9  
Add button called "List Reports"

Notice no printing button

The screenshot shows a web browser window displaying a 'call guides1' interface. The interface includes a date '03/04/03', a title 'CeIDial - Customer Inquiry', and a prompt 'Enter a customer number and press Enter.' Below this is a text input field labeled 'Customer number . . . . .'. To the right of the input field is the text 'Number, F4 for list'. A vertical menu on the left side of the interface contains the following options: Enter, Exit, Prompt, Print, Cancel, and HELP. A dialog box titled 'Add command key action' is overlaid on the right side of the browser window. The dialog box contains the following fields: 'Command key:' with a dropdown menu showing 'F9'; 'Action name:' with a dropdown menu showing 'Report Service'; 'URI or URL:' with a text field containing 'RetrieveListOfReports.do'; 'Target frame:' with a dropdown menu showing '\*NEW'; and 'Button label:' with a dropdown menu showing 'List Reports'. There is also a checkbox labeled 'Enable only when the command key is active' which is currently unchecked. The dialog box has 'OK' and 'Cancel' buttons at the bottom.

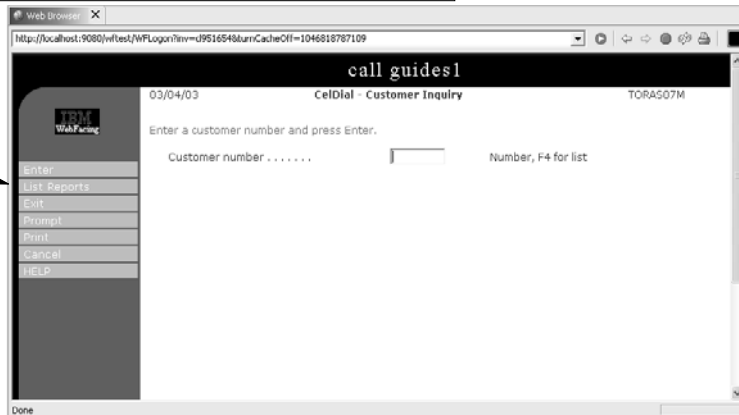


## WebFacing Tool - Example

■ **WRKSPLF example:**

After adding the command key, just restart application  
This key is now accessible via F9 or the button

Notice new  
"List Reports"  
button





# Sample Report Service page

- Control where the new window will go:
  - \*NEW – open new window
  - \*SAME – use current window

Add command key action

Command key:

Action name:

URL or URL:

Target frame:

Button label:

Enable only when the command key is active

OK Cancel

List of Spooled Files - Microsoft Internet Explorer

Address: http://localhost:9000/wf/test/RetrieveListOfReports.do

### List of Spooled Files

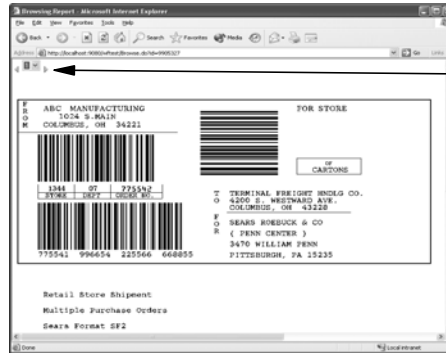
Views 1,2,3

File	User	Device or Queue	User Data	Sts	Total Pages	Cur Page	Copy	Form Type	Print File	Job	Number
BARSM	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	BARSM	*READY	1	0	1	*STD	5	13	GPADEV000F 905327
BARSM	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	BARSM	*READY	1	0	1	*STD	5	12	GPADEV000F 905327
EWLTR	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	EWLTR	*READY	1	0	1	*STD	5	19	GPADEV000F 905327
EWLTR	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	EWLTR	*READY	1	0	1	*STD	5	2	GPADEV000F 905327
EWLTR	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	EWLTR	*READY	1	0	1	*STD	5	29	GPADEV000F 905327
SEARS	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	SEARS	*READY	1	0	1	*STD	5	9	GPADEV000F 905327
BARSM	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	BARSM	*READY	1	0	1	*STD	5	1	GPADEV000F 905327
EWLTR	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	EWLTR	*READY	1	0	1	*STD	5	21	GPADEV000F 905327
BARSM	VOUTSN	AGSYS.LIBGGPL.LIBGGPRINT.OUTPUT	BARSM	*READY	1	0	1	*STD	5	20	GPADEV000F 905327

- Lots of spooled files for my userid, so 3 pages were created.
- Spooled files become links.

## WebFacing Tool - Example

- **WRKSPLF example:**  
Selecting a Spooled File link brings up the GIF image of each page



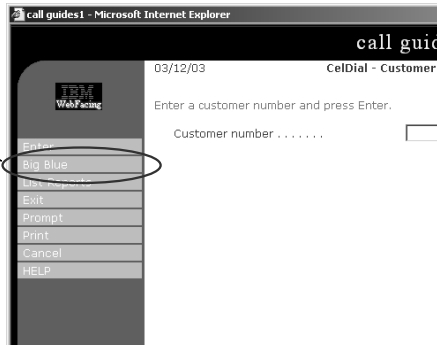
- User can page through using previous & next buttons
- User can also select a page from the drop-down



## WebFacing Tool – more Command Keys

- Associate keys with any web apps  
Including URL's

Notice new  
"List Reports"  
button



#### DDS INVITE

- Support for applications that use the DDS keyword INVITE and CL commands, such as SNDRCVF or RCVF WAIT(\*NO), to perform asynchronous input/output of data to the screen

#### DDS DSPATR(pfield)

- Support for applications that use program-to-system fields for setting the display attributes of protected fields

#### Job / System attributes

- Performance improvement for applications that require job attributes information for resolving certain DDS keywords



In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.

## Advanced Edition

### Portal

- Access DDS based RPG, COBOL, or CL applications from portal pages by generating portlets.

The generated portlets can be run locally from the IDE provided by WSC for testing and can be deployed to IBM's WebSphere Portal Server for running in production mode.

### Authentication

- Use single signon to allow users who have been authenticated by the WebSphere server to traverse WebFacing web or portlet applications without having to re-login

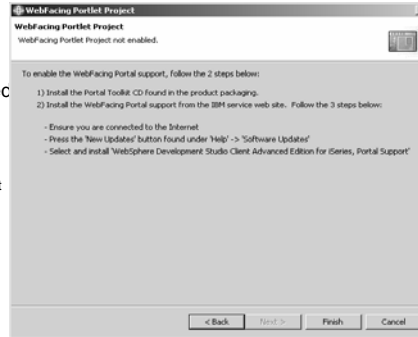
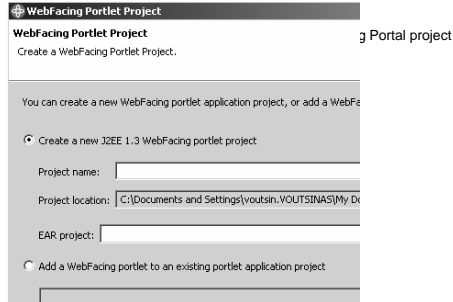
Uses Enterprise Identity Manager (EIM) (V5R2 and up)



In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.

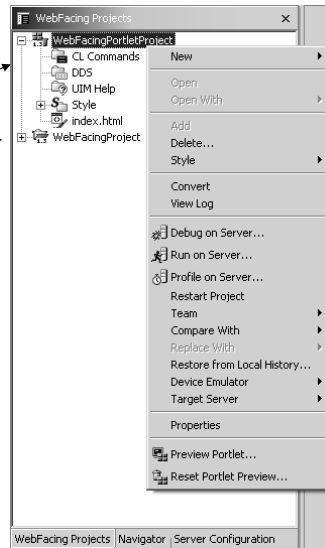
# WebFacing Portlet's

- Requires the installation of the Portal Toolkit V5.0.2.2
- Until then the support is disabled
- You then have 2 options:



## WebFacing Portlet's

- Same tooling now handles both types of projects
  - J2EE WebFacing projects
  - Portlet WebFacing projects
- Same options apply to both
  - Convert
  - View Log
  - Add DDS, UIM Help, CL Commands
- Ability to run on server also available
  - Part of the Portal Toolkit installation
  - WebSphere Portal Unit Test Environment V5.0.2.1
  - Just "Run on Server"





## WebFacing Portlet's

- Welcome to Portlets:

- WebFacing Portlet

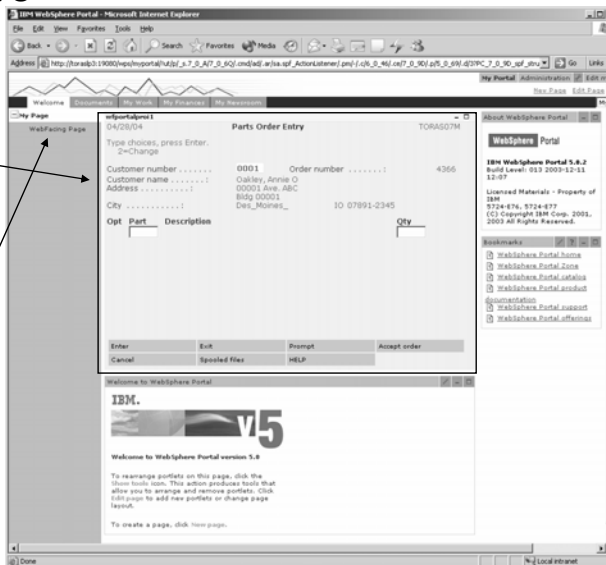
- The entire webpage is a Portal





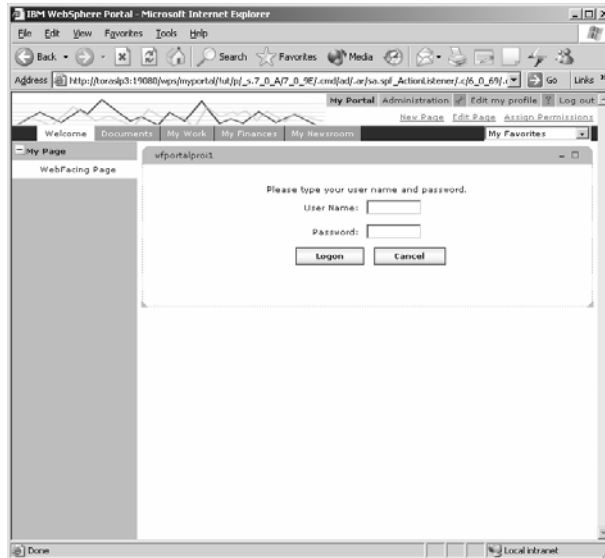
## WebFacing Portlet's

- After you sign onto the WebFacing application  
View expands for app
- You now have the option of multiple WebFacing app's on 1 page or:
- Separate pages in the same browser



## WebFacing Portlet's

- Notice on this page we've decided to take up the entire area
- Lots of freedom
- But after you signon, the portlet will expand to the required area.
- **NOTE:** each portlet is a separate job on the iSeries



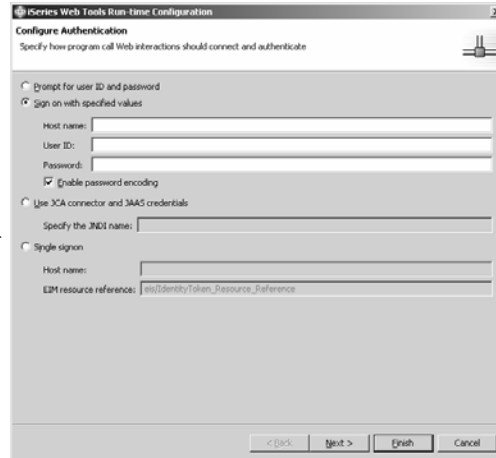
## WebFacing/Web tools – Single signon

- Single signon support using Enterprise Identity Mapping (EIM)
  
- Why would you care?
  - With a Portal website, you could have multiple applications on a single page
  - Signing onto 1 application, signs onto all (or a selection)
  - After signing onto the PC, you could share that authentication with the portlet's or other web applications.



## WebFacing/Web tools – Single signon

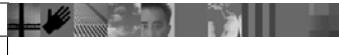
- iSeries web tooling also has Single signon
- Can share the authentication between WebFacing and other web applications
- Now available in tooling



## What's new in WebFacing for Version 6.0

### WebFacing decoupled from CODE components

- WebFacing conversion is now based entirely on a new DDS engine running in Eclipse
  - CODE Designer install is optional
- Positions WebFacing for future DDS to Web customization tooling running in Eclipse
- New WebSettings view in Eclipse
  - Driven from a new DDS outline view
  - On any perspective you can add the web settings view by the following action: Open View > Other > WebFacing > Web Settings



In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.



# Integrated WF development environment

© WebFacing - LK0215D.DSPF - IBM Rational Software Development Platform

File Edit Source Compile Navigate Search Project Run Window Help

WebFacing  
Remote System...

WebFacing... 1  
CL Commands  
DDS  
<toras1fb>TEST  
...  
Properties  
Property Value

\*LK0215... 3  
COMMDSP...  
Line 11 Column 48 Replace  
...+...3...+...4...+...5...  
000100 ESC R O 4 6REFFLD  
000101 fld={PAICAT.REF}|value= 1  
000110 SFLCTL (:  
000120 0302 120112 RHARVEY REL-  
000130 SFLSIZ (1  
000140 SFLPAG (1  
000150 WINDOW (:  
000160 KEEP  
000170 RTNCSRL  
000180 BLINK

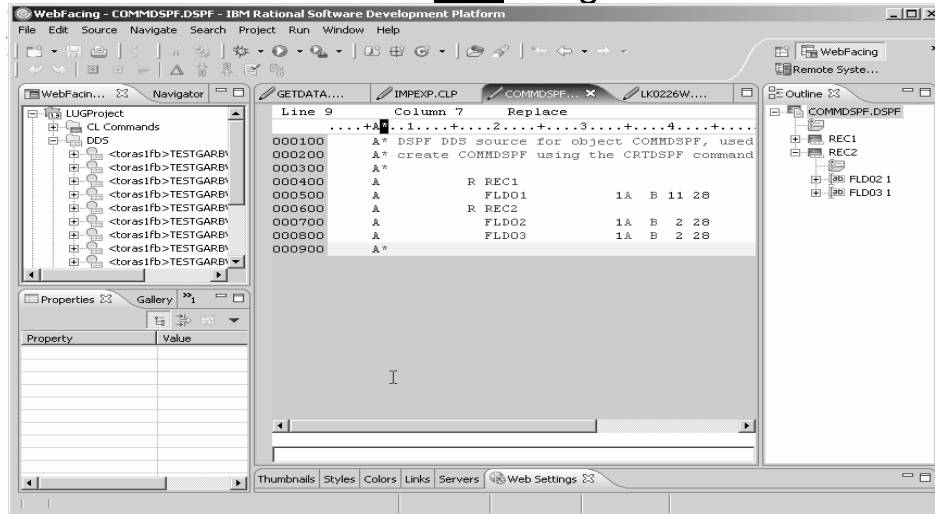
Outline  
LK0215D.DSPF  
DPSPIZ(24 80 \*D53) PRINT CF12 ROLLUP  
DPSPIZ(24 80 \*D53)  
PRINT  
CF12  
ROLLUP  
SFL  
SFL  
SFL  
PAICAT  
REFFLD(RPMCAT/PAICAT \*LIBL/PMCAT)  
Reference \*LIBL/PMCAT RPMCAT/PAICAT  
PADESC  
REFFLD(RPMCAT/PADESC \*LIBL/PMCAT)  
Reference \*LIBL/PMCAT RPMCAT/PADESC

Web Settings  
Dynamic key labels  
 Use the field value as the label for the function key: \*AUTO

© 2006 IBM Corporation

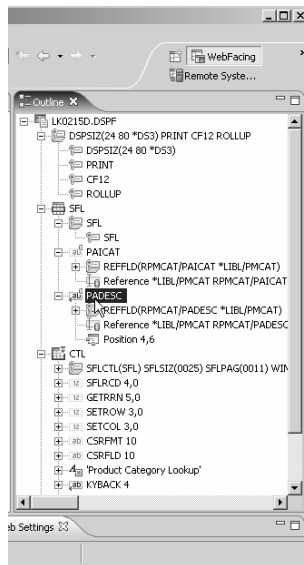


## Better DDS *edit* integration

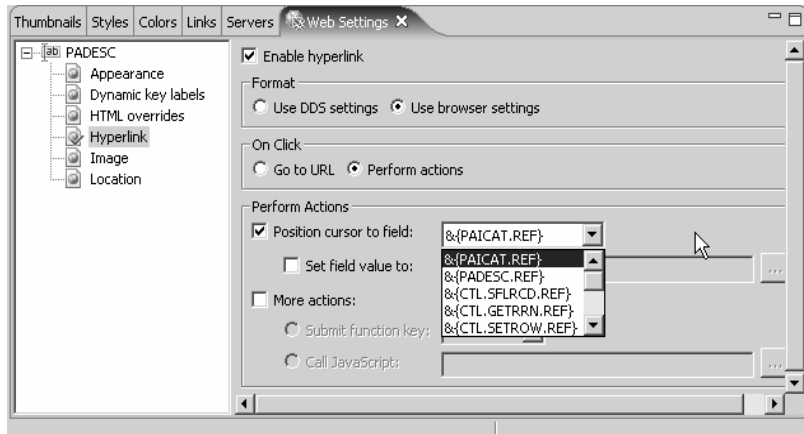




## Outline view for display files



## Web setting view in eclipse



Real-time source update!



WebFacing Version 6.0.1 introduces WebFacing Deployment Tool with HATS Technology (WDHT)



In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.

## WDHT

Integration with HATS: WebFacing supports calls to non-WebFaced applications

- WebFacing will dynamically transform non-WebFaced applications
  - Deployed runtime is chargeable
  - Requires the purchase of WDHT licenses
    - **WebFacing Deployment Tool with HATS Technology**
  - Does not require the HATS toolkit
  - Does not support HATS customization of screens
- Works in both Standard and Advanced version of WDS*c*
- Supported on i5/OS V5R4 and up
- Requires the following products:
  - WDS*c* 6.0.1 - GAed 01/2006
  - WDHT 6.0.1 - GAed 03/2006
  - Remember to download and apply WDS*c* service update 6.0.1.1



In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.

Summary

# References

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.

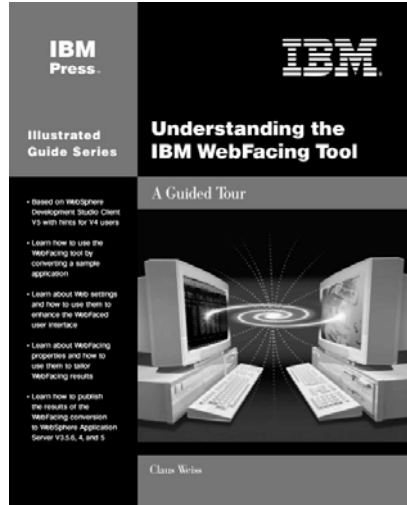
## More WebFacing information

### Understanding the IBM WebFacing Tool

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

IBM Press

Available now



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

### **IBM Redbook The IBM WebFacing Tool**

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

IBM

Available now



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.

Redbook for the more adva

SG24-6331-00  
[www.ibm.com/redbooks](http://www.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.



## New Redbook

SG24-6961-01  
[www.ibm.com/redbooks](http://www.ibm.com/redbooks)



© 2006 IBM Corporation

Java and e-business are key to the future of the iSeries server. Web-enabling your 5250 applications allows you to quickly participate in the e-business arena by using existing applications and programming skills.

This Redbook discusses WebSphere Development Studio Client for iSeries V5.0 of which the WebFacing tool is a part. Development Studio Client includes several powerful tools that target the iSeries application developers:

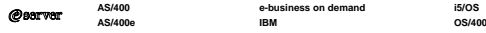
- WebSphere Studio Site Developer Advanced (in WebSphere Development Studio Client Standard Edition for iSeries V5.0) or WebSphere Studio Application Developer (in WebSphere Development Studio Client Advanced Edition for iSeries V5.0) - the new IDE for developing applications. The unique characteristic of this IDE is the ability to add new features in the form of the plugins. Anyone can develop a new plugin and install it into the tool without creating a "plumbing" infrastructure. WSSD and WSAD include the development environment for creating plugins.
- Development Studio Client includes several iSeries specific features, like the IBM WebFacing Tool and the iSeries Web development tools, that are installed as plugins.
- Cooperative Development Environment (CODE) - a workstation-based tool that supports the development of the applications in many different host languages, including RPG and Java.
- VisualAge RPG - If you are already an experienced RPG IV programmer, you can create graphical user interfaces to RPG programs very quickly in VisualAge RPG.
- Integrated iSeries Debugger.



# Trademarks and Disclaimers

© IBM Corporation 1994-2006. All rights reserved.  
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 of International Business Machines Corporation in the United States, other countries, or both:



Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.  
Intel, Intel Logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.  
Linux is a trademark of Linus Torvalds 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.  
UNIX is a registered trademark of The Open Group in the United States and other countries.  
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.  
Other company, product or 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 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 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.

© 2006 IBM Corporation

## 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.



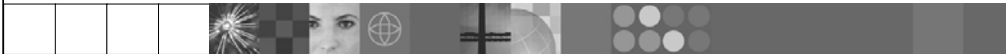


IBM Software Group

## IBM WebFacing Tool in IBM WebSphere Development Studio Client

iSeries Application Development Team: IBM Toronto

**WebSphere.** software



Spring 2006 | Advanced Topics - IBM WebFacing

© 2006 IBM Corporation