



Transforming Green-Screens into Rich Web Apps Using Genie

Exercise 1: Working with Skins

Task

Create a custom skin that will be used in later exercises. Explore commonly used configuration settings in the Genie Administrator.

Concepts

This first exercise is designed to introduce basic Genie concepts:

- Creating a skin.
- Customizing the HTML template and CSS.
- Using the Genie Administrator.
- Configuring commonly used settings.

Procedure

1. First, let's bring up the Genie Administrator.

Navigate to the following URL in your web browser:

<http://Systemi:8080/profoundui/genieadmin>

You'll be prompted for your System i sign-on credentials before the Administrator displays.

In this example, we'll go through the most commonly performed.

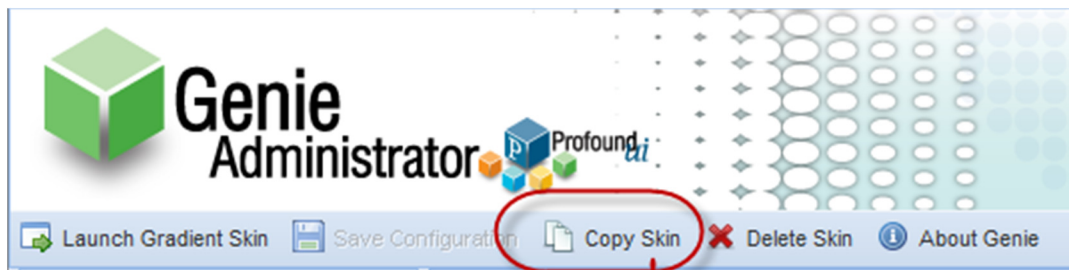
2. Create a copy of the 'Gradient' skin.

The Gradient skin is the default skin used in Genie. Typically, customers will create their own skins by copying from one of the product-provided selections.

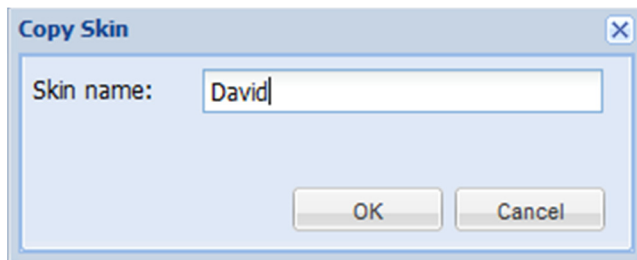
To do this, expand the Gradient skin in the "Skins" panel on the left side of the screen:



This selects the Gradient skin. You'll see its settings appear in the settings panel on the right. To copy the skin, click on the Copy Skin button in the toolbar:



This brings up the Copy Skin dialog. This is used to give the new skin a name. Use your name as the skin name. Use your last name also if there is another person in the lab with the same name. This will ensure that everybody has their own unique skin:



Click OK to create the skin.

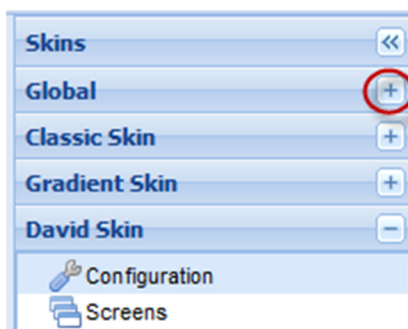
The skin will now appear in the Skins panel on the left. Expand the skin now. The skin's settings will now appear in the panel at the right.

There are 3 tabs in the settings panel that can be used to browse the settings. Try clicking on the tabs to explore the different settings available:

As you focus on each available field, you'll see some help information display in the bottom panel of the Administrator that explains each option.

You'll notice that some of the options are blank, or not set to anything. This is because settings can also be configured at the Global level.

Anything not set at the Skin level will be inherited from the Global Settings. To get a sense for this, try expanding the Global item in the "Skins" panel:



You'll notice that the same settings are available at the global level as well.

In a realistic environment, most settings would be made at the Global level since you would typically have just 1 skin.

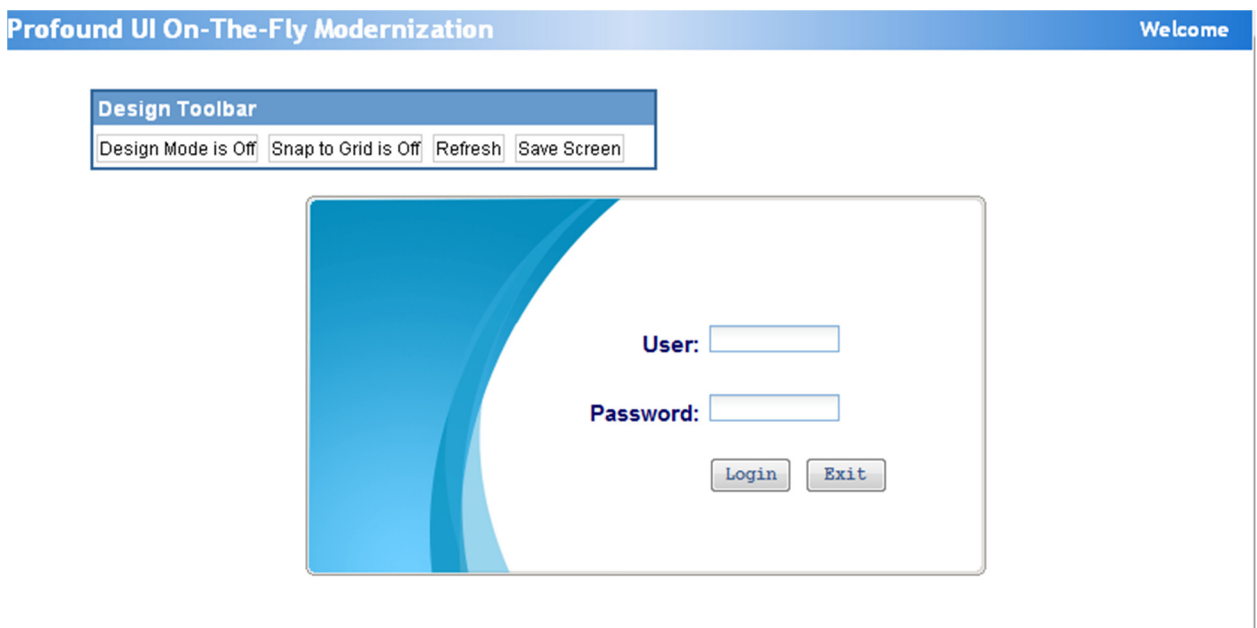
For the purposes of this lab, though, we will make the settings in our own skins. **Please do not change anything in the Global configuration.**

3. Launch a Genie session with your skin.

To do this, first expand your skin again in the Skins panel. Then, click the “Launch” button in the toolbar:



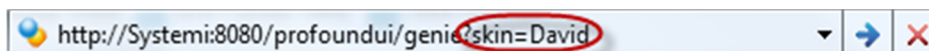
A browser window will open to a Genie session. It should look something like this:



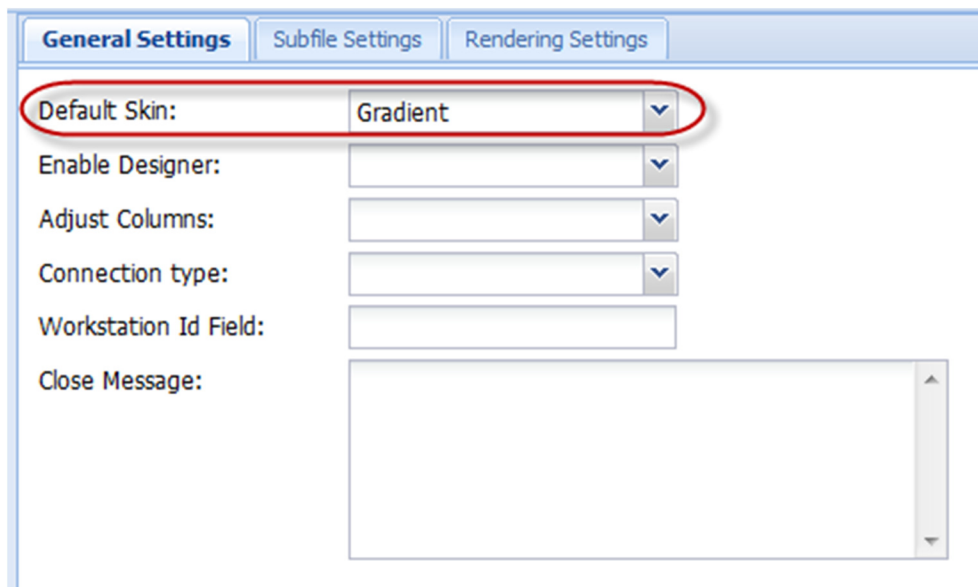
Believe it or not, what you are seeing is the standard System i sign on display. A customization has been applied that makes it look a lot different.

Already you can see some of the things that are possible.

Look at the address bar in your browser. You’ll notice that Genie has been launched with a query string parameter that specifies the skin name:



In a realistic configuration, you would typically set the “Default Skin” setting in the Global settings of the Administrator, instead. **Do not change the default skin in this lab.**



The screenshot shows a settings window with three tabs: 'General Settings', 'Subfile Settings', and 'Rendering Settings'. The 'General Settings' tab is active. It contains several configuration options: 'Default Skin' (a dropdown menu set to 'Gradient', highlighted with a red circle), 'Enable Designer' (a dropdown menu), 'Adjust Columns' (a dropdown menu), 'Connection type' (a dropdown menu), 'Workstation Id Field' (a text input field), and 'Close Message' (a large text area). The 'Default Skin' dropdown is currently set to 'Gradient'.

Please familiarize yourself with that setting, but again, do not make any changes to the Global configuration (when a default skin is configured, no special parameter is needed to launch Genie to that skin).

4. Sign into your Genie session and explore your applications.

When signing in, you will come to your normal IBM i menus. Experiment by navigating through a number of IBM i menus.

5. Explore the skin’s HTML template file.

Browse the HTML template file on the System i IFS at:

`/www/profoundui/htdocs/profoundui/userdata/genie skins/YOUR_SKIN_NAME/start.html`. You can use the WRKLNK command or use RDp to view an IFS file.

Customizing the file will require some familiarity with HTML and CSS coding. Information on this can be found in standard HTML and CSS references and tutorials.

The file looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<title> Profound UI </title>

<link href="/profoundui/userdata/beta/skins/David/David.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="/profoundui/userdata/beta/skins/David/custom.js"></script>
<link href="/profoundui/proddata/css/profoundui.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="/profoundui/proddata/js/genie_beta.js"></script>

</head>

<body>

<div class="mainDiv">

<div class="topDiv" align="center">
    <table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td id="gradient1" width="50%">
                Profound UI On-The-Fly Modernization
            </td>
            <td id="gradient2" width="50%">

                <span id="userSpan">Welcome</span>

            </td>
        </tr>
    </table>
</div>

<div id="middle" class="middleDiv centered">

<div class="insideDiv" style="position:relative; visibility:hidden;" id="5250">

</div>

</div>

</div>

</div>

</body>

</html>
```

Some important things to note are below.

- **The HTML <title> element:** This can be used to control the title of the browser window or tab.
- **The 5250 <div> element:** This is the container into which the 5250 application content will render.

The container must have an id of “5250”.

5250 content within this container will be in fixed positions, but the container itself can be moved around, centered, etc. The default Gradient skin centers the content using CSS.

- **The HTML <link> element:** This is used to link in the skin’s style sheet. All skins must have a style sheet which matches the skin name.

Additional style sheets can be linked in, if desired.

- **The HTML <script> elements:** These are used to link in JavaScript files. The “genie.js” file must always be linked into the skin.

You’ll notice also that the “custom.js” file is linked in from the skin folder. This is where you can place any custom JavaScript code.

Additional JavaScript files can also be linked in.

6. Explore the skin’s CSS.

Browse the CSS file in your skin’s folder. The CSS file is used to provide browser styles for each 5250 display attribute.

Try searching the file for ‘A20’.

This corresponds to 5250 display attribute x’20’ (standard green). You’ll find CSS classes that look like this:

```
/* Regular Green */
DIV.A20 {
    color: #003399;
    font-weight: bold;
}

INPUT.A20 {
    color: #003399;
    font-weight: bold;
}
```

For specifics on CSS, please consult any CSS tutorial or reference.

Genie assigns a CSS class to all 5250 fields, based on the display attribute. The CSS class name is 'A' + AttributeCode, where 'AttributeCode' is the hex code for that attribute.

In Genie, output fields are represented by HTML <div> elements. Input fields by <input> elements.

The CSS tag name selectors can be used to apply different style rules between output/input fields with the same original attribute code, although in this example the styling is the same.

In this case, any standard green fields are mapped to a blue color and given a bold font weight.

Try changing the DIV.A20 class like this:

```
DIV.A20 {  
    color: red;  
    font-weight: bold;  
}
```

Then, reload your Genie session by hitting the browser's refresh button. You should see all fields that are normal green in 5250 display now in red.

This is how you can control mapping of 5250 attributes to browser colors, fonts, etc. All such global changes should be done this way using the style sheet.

Conclusion

After completing this exercise, you should be familiar with using the Administrator to create a skin and configure basic settings.

You should also be comfortable with the usage of the basic files that comprise a skin.

Exercise 2: Working with the Designer

Task

Get to know the Genie designer, and make some useful screen customizations while we're at it.

Concepts

This first exercise is designed to introduce basic screen designer concepts:

- Setting up screen detection.
- Repositioning fields.
- Using field set containers to group fields.
- Automating keystrokes with buttons.

Procedure

1. Run the example application in Genie.

Start a Genie session and get to a command line. Set the library list like this:

- **PUILABFILE**
- **PUILAB**

Call RPG program PRODMNTR. The application should look as below.

Product Number: Product Name:

2=Change 5=Display

Opt	Product	Product Name	Price	Qty	Special Flag
▼	904	ABC ACCESSORY BELTS	7.23	50	Y
▼	889	ABC AVALANCHE PROBE	123.00	50	N
▼	839	ABC BELAY SEAT	15.61	50	N
▼	961	ABC BIG WALL GEAR SLING	58.00	50	Y
▼	804	ABC BLAST PACK	85.95	50	N
▼	717	ABC BLENDS	7.90	50	N
▼	908	ABC CHALK BAGS	12.00	50	N
▼	909	ABC CHALK BAGS	12.00	50	N
▼	882	ABC COURIER BAG	55.00	50	N
▼	923	ABC CRAGSTER BOLT KIT	125.00	50	N
▼	958	ABC DIRTBAGGER ROPE TARP	24.00	50	N
▼	805	ABC GUIDE HARNESS	30.95	50	N
▼	718	ABC HEUVOS	5.35	50	N
▼	714	ABC HIGH STEP ETRIERS	26.95	50	N
▼	806	ABC MONKEY JUNIOR HARNESS	35.95	50	N
▼	910	ABC ROCK STAR CHALK BAG	15.00	50	N
▼	962	ABC SENTENTIAL GEAR SLING	15.50	50	N

Here you can see Genie's automatic subfile enhancement at work. It's placed a nice-looking grid widget over the subfile and has converted the subfile options displayed on the screen into a combo box that can be used to submit the options.

The screen that we'll be customizing is the detail screen.

Take option 2 on any subfile record. The detail screen should appear as below:

Product Number. . .	904
Product Name. . . .	ABC ACCESSORY BELTS
Long Description. .	
Price	7.23
Quantity.	50
Category.	81 Climbing Accessories
Special Pricing . .	Y

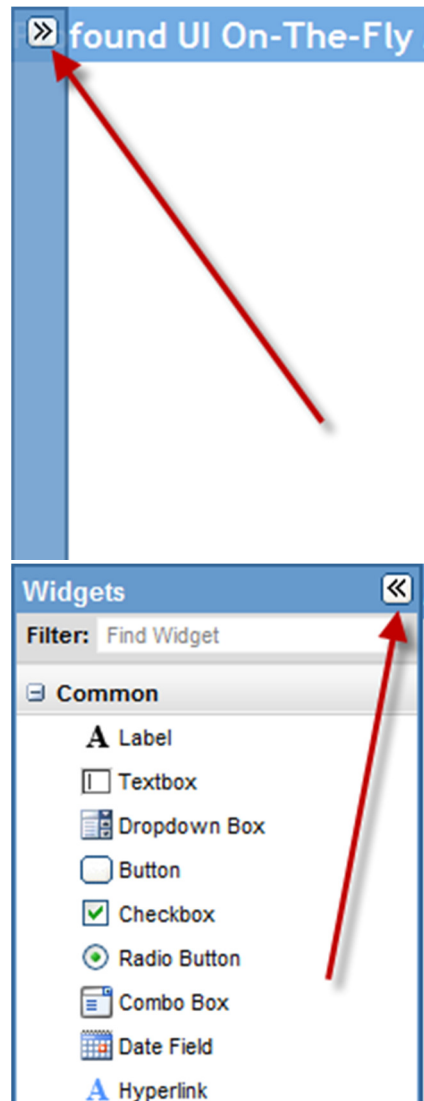
2. Enter design mode.

Click on the button "Design Mode is Off" on the Design Toolbar:

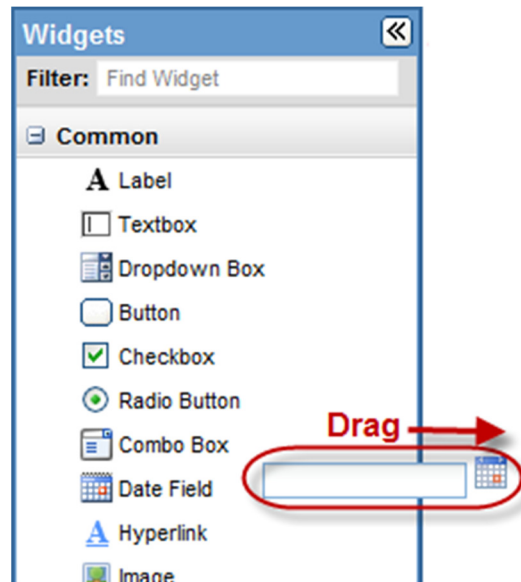


This will bring up design mode.

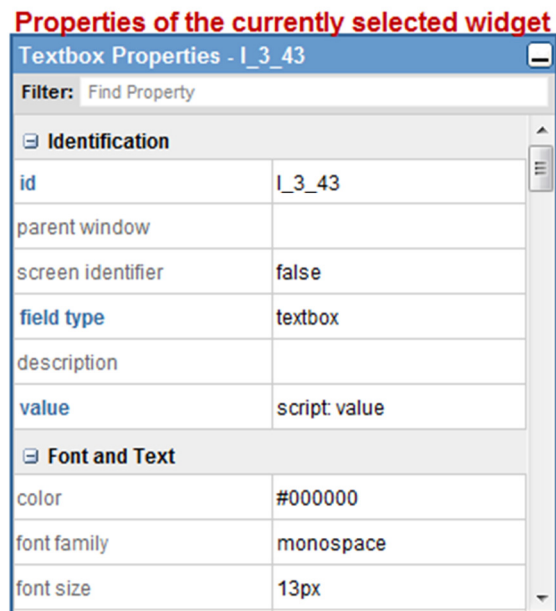
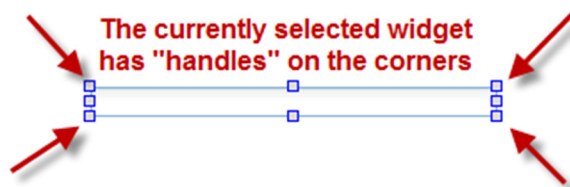
3. The Designer is made up of a widget panel and a properties dialog.
 - a. The widgets panel is on the left of the page and is collapsed by default. You can expand and collapse the panel as shown below



- b. To add a new widget to your screen, simple drag a widget from the panel and drop it on the screen



- c. The other main component of the designer is the properties dialog. This dialog shows the properties of the currently selected widget.
- d. Clicking on a widget will select it. You can tell when a widget is selected because it will have “handles” on the corners.



- e. You can change the properties (for example color) of a widget by changing the (color) property in the dialog.
- f. You can move a widget by selecting it and dragging it to the new location.
- g. You can select multiple widgets by dragging a “rubber band” around the widgets



- h. You can also select multiple widgets by holding down the Ctrl key on the keyboard while clicking each widget. This is a good way to move multiple widgets together.

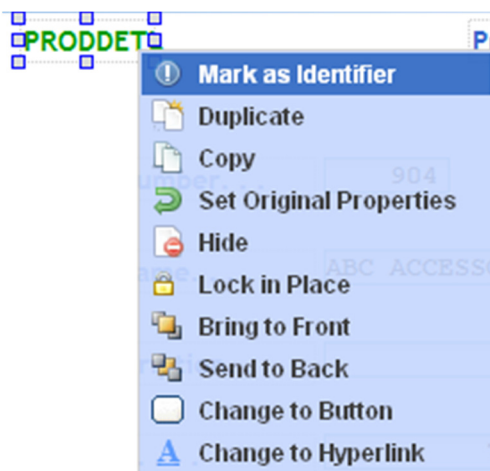
Hold down Ctrl key while clicking widgets



4. Setup screen detection.

The ideal candidate for screen detection here is the field that shows the screen record format name.

Right-click on the field and select “Mark as Identifier”.



This sets the field as an identifier. When screens are first loaded, Genie will look for any screen identifiers to try to match the screen with a designer customization.

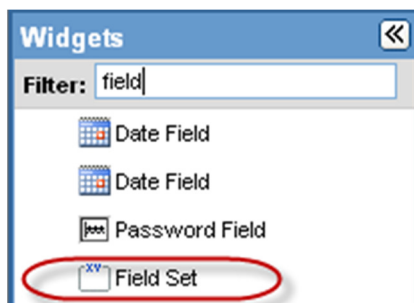
It's very important that identifiers chosen are entirely unique for this reason.

Choosing this field as an identifier also sets the screen name in the Properties Window to "PRODDCTL". Genie automatically does this for the first identifier chosen. This can be changed to anything you want.

5. Add fields to a fieldset container.

The first thing we'll do to this screen is drop all the fields into a fieldset container. This provides a convenient method of grouping the fields so that we can move them around together.

In the Widgets Toolbox, type 'field' into the search box at the top. Then, drag a fieldset widget onto a blank space on the screen:



You will need to resize the field set to make it large enough to accommodate all the fields.

Then, select all fields except for the buttons on the bottom. You can do this by dragging the "rubber band" around them all with the mouse.

Then, pick them up and drag them into the field set. The screen should look something like this now:

PRODDCTL

Product Maintenance

17:31:58

Field Set

Product Number. . .

904

Product Name. . . .

ABC ACCESSORY BELTS

Long Description. .

Price

7.23

Quantity.

50

Category.

81

Climbing Accessories

Special Pricing . .

☒

If the elements are properly dropped into the field set, you should be able to pick up the field set and drag everything around at once.

You can set the field set title, if you'd like, by double-clicking on it and typing the desired text.

6. Adjust the buttons.

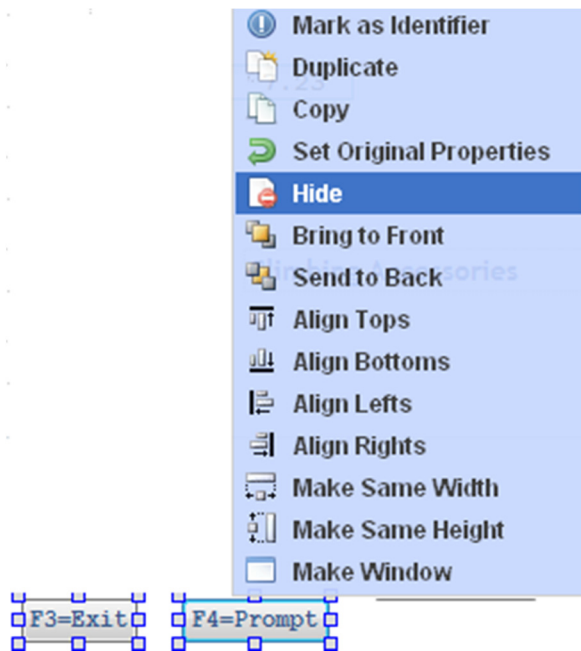
The original green-screen display defined 3 function keys. These were transformed into push buttons by Genie based on the function key label text on the screen.

In our example, the prompt key will no longer be relevant as we are going to change the prompt into a dropdown box in a later exercise.

Also, the screen has an Exit and Cancel button. This is common on green-screen panels, but not very “web-like”.

The typical web user would expect a Submit and Cancel button.

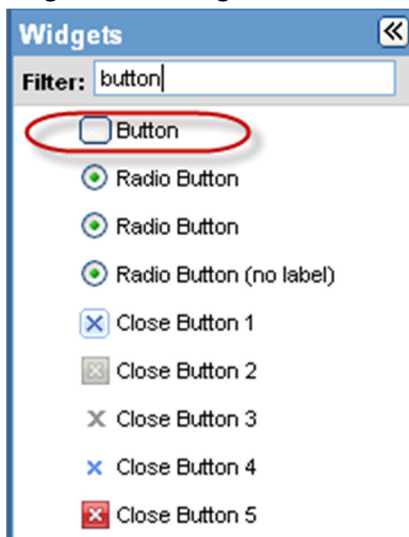
To make the screen look more web-like, hide the Exit and Prompt buttons. You can do this by right-clicking on them and selecting “Hide”.



7. Add a Submit button.

Green-screen users are accustomed to pressing the Enter key to submit screens. A web user, though, would find it very strange if there were no Submit button!

Drag a button widget onto the screen:



You can set the button text by double-clicking on the button and then typing in 'Submit'. You may also wish to edit the text on the Cancel button to get rid of the function key number.

Reposition the buttons to your liking, and the screen will look something like this:

Product Information

Product Number. . .	904
Product Name. . . .	ABC ACCESSORY BELTS
Long Description. .	
Price	7.23
Quantity.	50
Category.	81 Climbing Accessories
Special Pricing . .	☐ Y

8. Add a key press action for the new submit button.

Just adding the Submit button onto the screen is not enough. We must make the button perform an action when clicked.

Genie includes a JavaScript API function **pressKey()** that can be used to press any valid function key or the enter key.

To make our submit button press the enter key, set its "onclick" property as shown below:

Button Properties - Button1	
Filter:	Find Property
Events	
onblur	
onchange	
onclick	pressKey("enter");
ondblclick	

9. Test the customizations so far.

Click the “Save Screen” button on the design toolbar. You’ll get a message confirming the screen was saved successfully.

Then, click the “Design Mode is On” button on the design toolbar to exit design mode.

You can now test the customizations so far. Already, the screen should be looking a bit more web-like. You should also be able to submit the screen using the new button that we added.

Conclusion

After completing this exercise, you should be familiar with basic usage of the screen designer. Further exercises will demonstrate other common customizations.

Exercise 3: Dynamic Images

Task

Add a dynamic product image to the product maintenance example program.

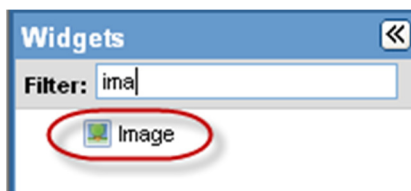
Concepts

This exercise is designed to introduce the following concepts:

- The image widget.
- Retrieving data from the screen.

Procedure

1. Launch the PRODMNTR application from the previous exercise in a Genie session.
2. Navigate to the detail screen by selecting option 2 on any subfile record.
3. Add an image widget onto the screen:



Drag the image widget onto the screen to the right of the product information. The screen should look something like this:

PRODDCTLProduct Maintenance21:21:57

Product Information

Product Number. . .904

Product Name. ABC ACCESSORY BELTS

Long Description. . .

Price 7.23

Quantity. 43

Category. 81Climbing Accessories

Special Pricing . .Y

Image Id. 939

Submit

Cancel

F3=Exit

F4=Prompt



4. Set the image source.

To display an image, all we have to do is set the widget's "image source" property. This can be set to a specific image file (located within the document root of the http server) using the file browser that is provided by the Properties window.

However, in this case we want to display the image dynamically for each product. To do this, we'll need to pull some information from the screen.

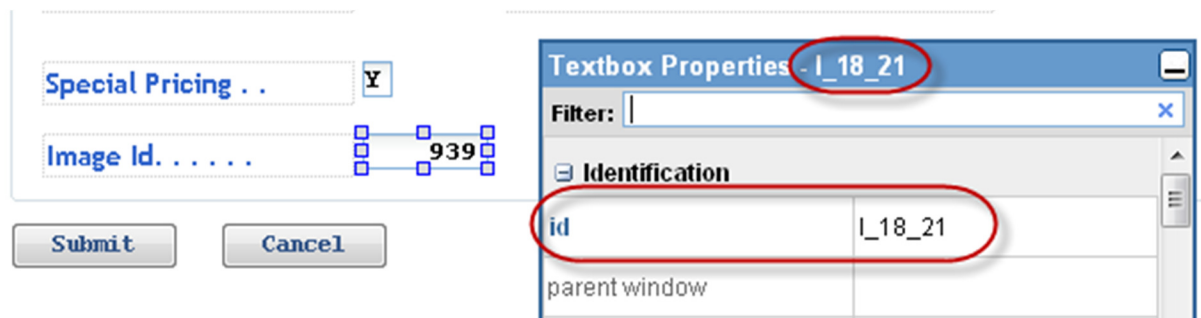
Sample images are provided for this example.

Field "Image Id" on the screen can be used to locate the image. The sample images are named **"IMAGE_ID.gif"**.

To pull the image id from the screen, we'll use a bit of JavaScript. First, though, we must be able to identify the field.

For this purpose, Genie provides a unique identifier for all fields on the screen. This is based on the field's original row/column position on the 5250 screen.

To see the “Image Id” field’s id, simply select it by clicking on it. You’ll see the field id in the title bar of the Properties Window, and also in the “id” property:



If you are familiar with HTML, you can guess that this is actually HTML element id. If not, it’s enough to know for now that this is a unique identifier that can be used to “find” the field later.

The format of the field id is “X_ROW_COL”, where X is either ‘D’ (output field) or ‘I’ (input field) and the row and column numbers are based on the original 5250 screen positioning.

Make a note of the field id –“I_18_21”.

Now that we know the field’s id, we can get the value from it using JavaScript. Genie provides an API **get()** that can be used for this. All that we have to do is pass the unique id.

Find the image element’s “image source” property. Set the property to this:

script: "/profoundui/userdata/product_images/" + get("I_18_21") + ".gif"

Let’s examine this line of code.

Previously, we have looked at properties to static values. In this case, we are setting the “image source” to a dynamic value.

This is achieved through the “script:” prefix. This tells Genie not to take the property value literally, but rather to evaluate it as a JavaScript expression.

The first part of the expression **"/profoundui/userdata/product_images/"** is a JavaScript string constant – the static portion of the path to the image file in the HTTP server’s document root folder.

The next part of the expression is where it gets interesting. Here, the return value of the **get()** API is appended to the string. The id of the field we want to retrieve the value from is passed to **get()**.

Finally, the **.gif** extension is appended, resulting in the complete file path.

5. Save the screen and test.

Save the screen customization as in the previous example. Exit design mode and try selecting a few different product records from the subfile.

The correct image file should display for each product.

Conclusion

After completing this exercise, you should be familiar with usage of the image widget, and also with basic property scripting. You should understand that the **get()** API can be used to read values from the screen for use in JavaScript expressions.

Exercise 4: Checkbox Transformation

Task

Replace the yes/no input field on the product maintenance example program with a checkbox.

Concepts

This exercise is designed to introduce the following concepts:

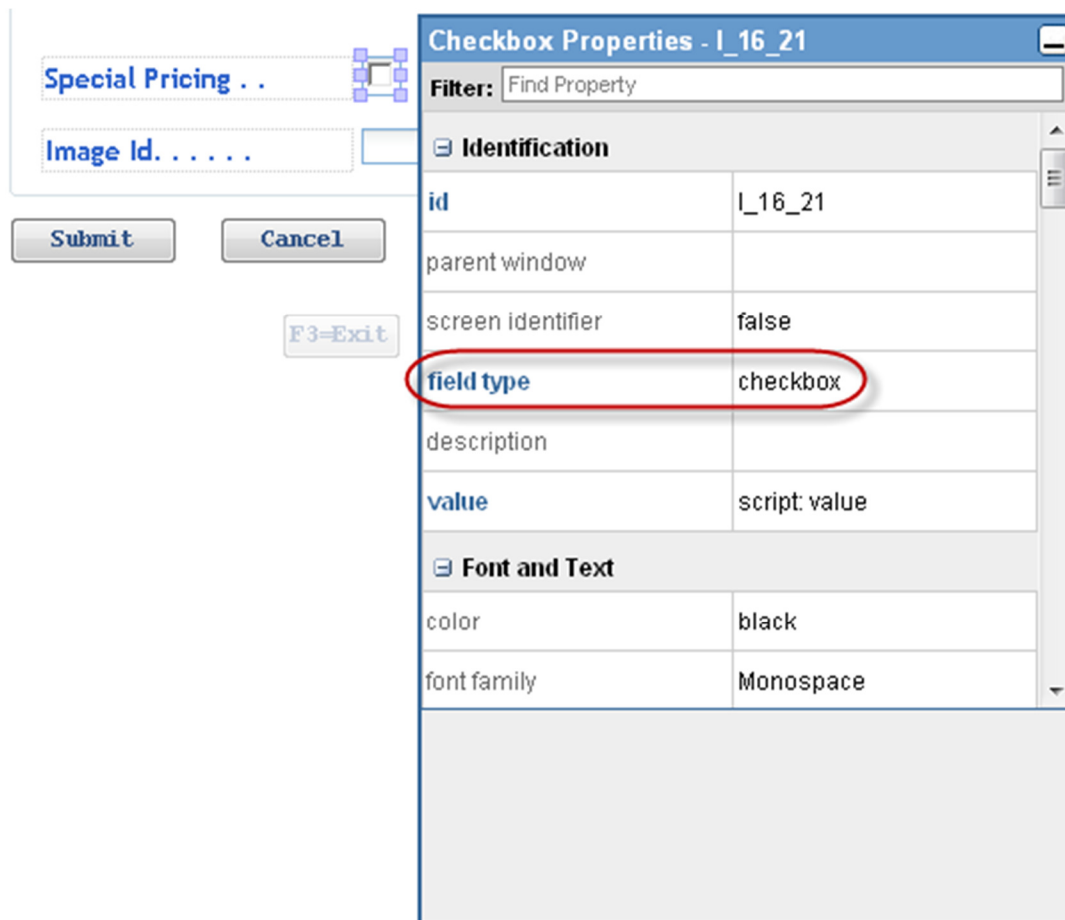
- Field transformations.
- Checkboxes.

Procedure

1. Launch the PRODMNTR application from the previous exercise in a Genie session.
2. Navigate to the detail screen by selecting option 2 on any subfile record.
3. Transform the “Special Pricing” field into a checkbox.

In this field, you can enter a ‘Y’ for yes, or a blank for no. This is not very intuitive for a web user.

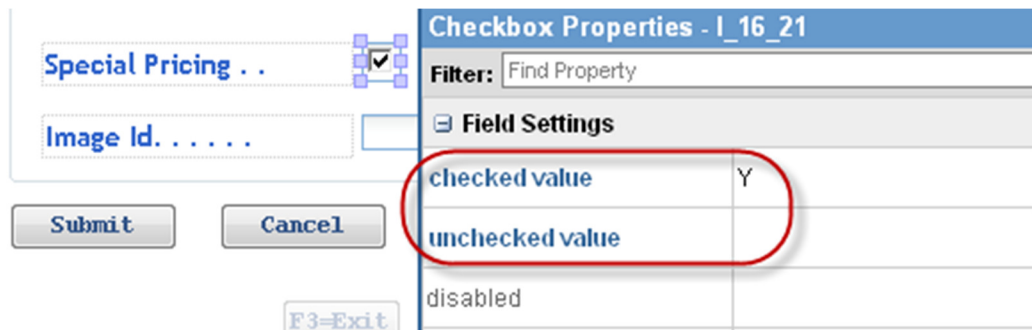
Transform the field into a checkbox by changing the “field type” property to “checkbox” as shown below:



The key properties on a checkbox are the “checked value” and “unchecked value” properties. These specify what values need to be shown in the 5250 field to render the checkbox as checked or unchecked.

These also control what values are written back to the 5250 field when the screen is submitted to the server.

In this case, the field takes a ‘Y’ or a blank, so set the properties as below:



4. Test the application.

Save and exit design mode. Verify that the checkbox properly reflects the state of the field when the detail screen loads.

Also verify that the record updates with the proper value in the field when the screen is submitted.

Conclusion

After completing this exercise, you should be comfortable with the checkbox widget and also with the concept of field transformations.

In Genie, customizations are performed largely by changing appearances only. The same values are fed back into the fields in the RPG application.

It seems like a simple concept, but this can be used to make screens much more user friendly. Already, our screen is starting to look more and more like a “real” browser screen.

Exercise 5: Database-driven Selection

Task

Add a database-driven dropdown box to replace the category prompt window on the product maintenance example program.

Concepts

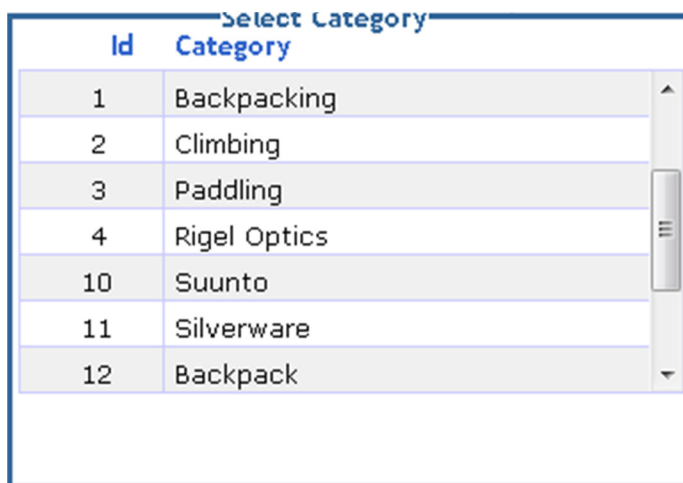
This exercise is designed to introduce the following concepts:

- Database driven components.
- Dropdown boxes.

Procedure

1. Launch the PRODMNTR application from the previous exercise in a Genie session.
2. Navigate to the detail screen by selecting option 2 on any subfile record.
3. Transform the category id field into a dropdown box.

This program allows the user to prompt for the category id using the F4 key. When the F4 key is pressed on the category id field, the category selection window is displayed:



The screenshot shows a window titled "Select Category" with a table of categories. The table has two columns: "Id" and "Category". The categories listed are Backpacking, Climbing, Paddling, Rigel Optics, Suunto, Silverware, and Backpack. A vertical scrollbar is on the right side of the table.

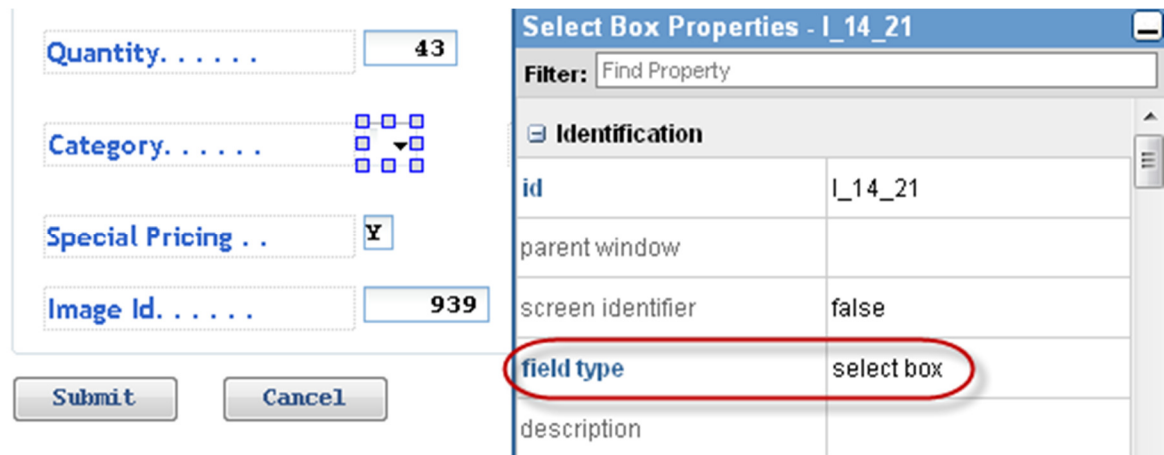
Id	Category
1	Backpacking
2	Climbing
3	Paddling
4	Rigel Optics
10	Suunto
11	Silverware
12	Backpack

The user can select a record and press enter to write the selected category id back to the main display.

While this sort of thing is second nature to an experienced green-screen user, it may not be very friendly for a web user.

In this case, a simple dropdown box could be used to select from the valid categories.

We can transform the category id field into a dropdown box by selecting it and setting its “field type” property to “select box” as shown below. Do that now:

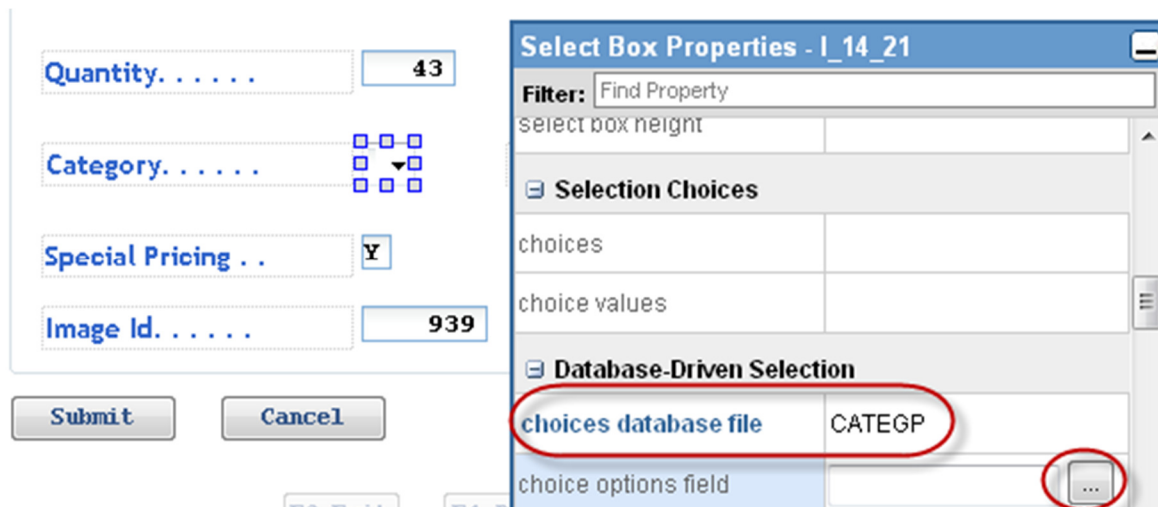


In the dropdown box, we want the user to see the category names, but we still need to return the numeric category id into the 5250 field for proper functioning of the application.

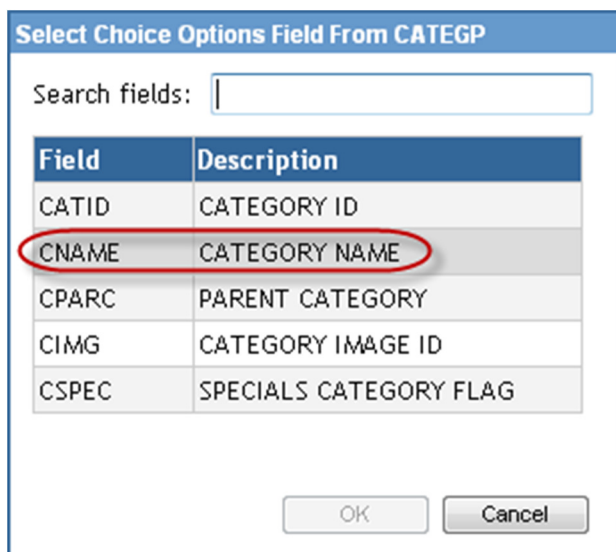
There are a variety of ways we could load the dropdown. For one, we could simply hardcode a list of values using the “choices” and “choice values” properties. This might be appropriate for simple dropdowns with options that never change.

However, in this case we want to load the categories dynamically from our categories file. To do that, use the set of properties under the “Database Driven Selection” category.

Type in the file name CATEGP under “choices database file” as shown below. Then click on the “browse” button on the “choice options field” property.



This brings up the field listing dialog. Here we can select the field the user will see in the dropdown. Select CNAME by clicking on the field name:



Field CNAME holds the category name. This field will display in the dropdown box.

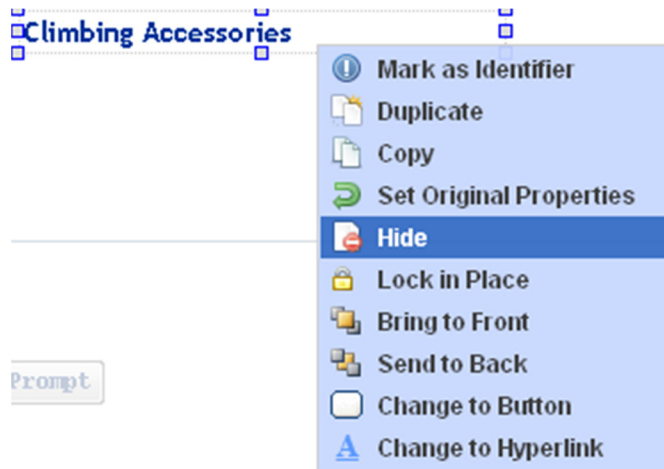
Repeat the process for the "choice values field" this property controls the field that will be used for the value that is passed back to the 5250 field. In this case, select the category field CATID.

4. Hide the category name output field.

The original application shows the category name beside the category id field. This will no longer be necessary as the dropdown box will show the category name.

Right-click the category name output field and select "Hide".

Genie Lab 5: Database Driven Selection



The screen should now look something like this:

 A screenshot of a web application titled 'PRODDCTL' with a 'Product Maintenance' tab. The main content area is a 'Product Information' form. It contains several input fields: 'Product Number...' with value '904', 'Product Name...' with value 'ABC ACCESSORY BELTS', 'Long Description...' (empty), 'Price...' with value '7.23', 'Quantity...' with value '43', 'Category...' with a dropdown menu showing 'Climbing Accessories', 'Special Pricing...' with a checked 'Y' checkbox, and 'Image Id...' with value '939'. To the right of the form is a small image of a climbing accessory. At the bottom of the form are 'Submit' and 'Cancel' buttons. In the top right corner, a timestamp '22:59:11' is displayed.

Note that we have not sized the dropdown box. You should only size the box if you want to limit it to a specific width.

When the box is not sized, it will automatically expand to accommodate the longest option text when the options are loaded.

5. Test the application.

Save and exit design mode. The list of categories should load into the dropdown box automatically.

Verify that the dropdown box positions to the correct category when the detail screen loads, and also that the application is able to update the record properly with the selected category.

Conclusion

After completing this exercise, you should be familiar with usage of drop down boxes, and database-driven selection.

The instructor may also demonstrate some additional dropdown box options at this time.

You should also have a good field for how field transformations work in Genie. The concept of all widgets is similar to the checkbox and dropdown examples we have looked at so far. The designer widgets all work sort of “on top of” the original 5250 fields, feeding in the necessary values to make the application function normally.