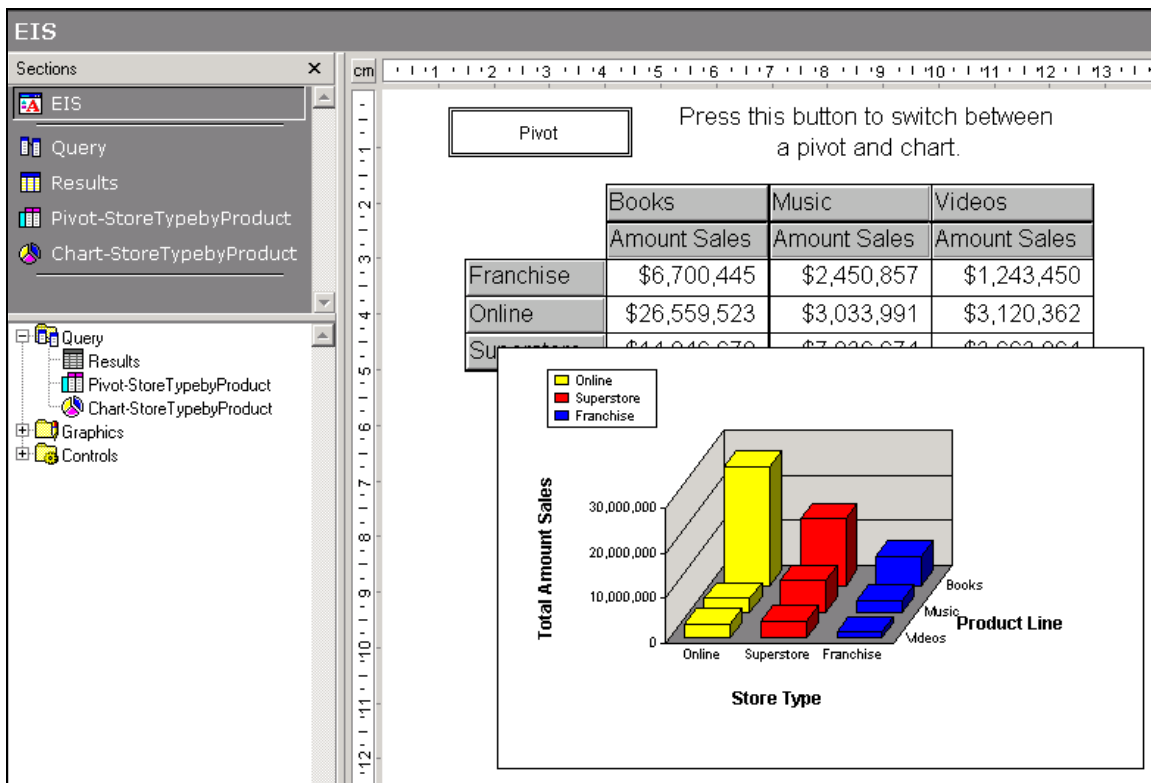


Advanced JavaScript Supplement

Switching between Pivots and Charts

Objective: Demonstrate how to use a control button to switch back and forth between a pivot and a chart within the same EIS.

First, create a pivot and chart. Both the pivot and chart need to be brought into the EIS and resized (select pivot > right click for speed menu > check Auto-Size). Use the adjustment icon on the Section Toolbar to line up the pivot and charts, making them the same size.



	Books	Music	Videos
	Amount Sales	Amount Sales	Amount Sales
Franchise	\$6,700,445	\$2,450,857	\$1,243,450
Online	\$26,559,523	\$3,033,991	\$3,120,362
Superstore	\$14,040,070	\$7,030,074	\$3,000,000

Next, add a button. Change the button's title to "Pivot" (in the button's Properties). Add the following code for the button:

```
// Switch between the Pivot and Chart  
// Note: both the Pivot and the Chart have to be  
// located in the EIS and specially sized.
```

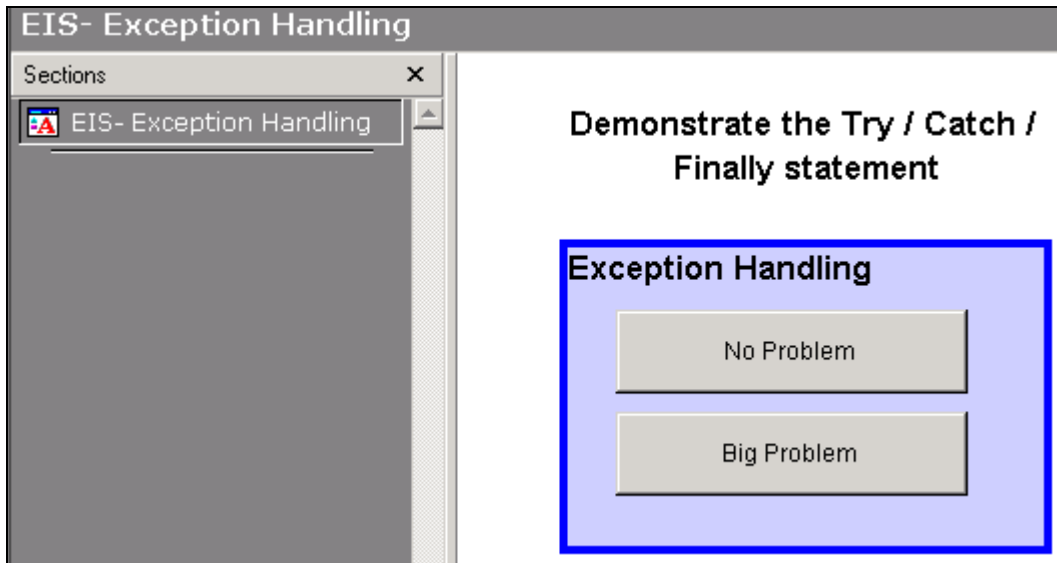
```
if ( CommandButton1.Text == "Pivot" )
{
Chart1.Visible=false;
Pivot1.Visible=true;
CommandButton1.Text="Chart";
}
else

{
Chart1.Visible=true;
Pivot1.Visible=false;
CommandButton1.Text="Pivot";
}
```

The text will change automatically when users click the button and toggle between the pivot and chart.

Error Handling

Objective: Demonstrate how to use the **try – catch – finally** statements in JavaScript to provide error handling in Brio.



Code for “No Problem” Button:

```
try
{
//exception
Application.Alert ("Start")
}
catch (e)
{
//e.getMessage();
Application.Alert ("Big
ERROR!!!")
}
finally
{
//always executed

Application.Alert ("Finished")
}
```

Code for “Big Problem” Button:

```
try
{
//exception
Application.Alert ("Start")

ActiveDocument.Sections ["Query
DoesntExist"].Activate ()
}
catch (e)
{
//e.getMessage();
Application.Alert (String(e))
}
finally
{
//always executed

Application.Alert ("Finished")
}
```

All the code that should be error checked can be placed in the **try** block. If an error occurs, execution will jump to the **catch** block, then to the **finally** block. If no error occurs in the **try** block, execution will skip the **catch** block and pass to the **finally** block. The **finally** block cannot be left off. The **catch(e)** statement is a function passing the argument **e**, which in the example, gets printed by the alert box using the JavaScript Function **String(e)**.

In the event that you know your code works, but still generates a strange error, using the **try – catch – finally** statements could be a good work around solution.