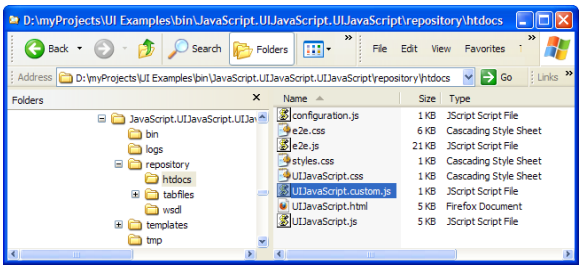
Debugging Custom JavaScript Functions

Code Generation

Custom JavaScript functions created on the <<UI>> controller are generated in a separate file. The file itself is located in the web root of the xUML UI package and has is names <controller>.custom.js.



The content of the custom.js file shows the generated code according to the definition within the xUML UI model. The formal parameter controller holds the current controller object. All public attributes and all functions defined on the UI controller are then assigned to this object. All these assignments are encapsulated in a JavaScript function that gets the e2e- and jQuery-module as formal parameters e2e and $ respectively:

On this Page:

Related Pages:

- Overview on the Used Web 2.0 Components
- Overview on the Used Frameworks
- Overview on the Generated Browser Code Elements
- Debugging Custom JavaScript Functions
- Optimizing Generated JavaScript

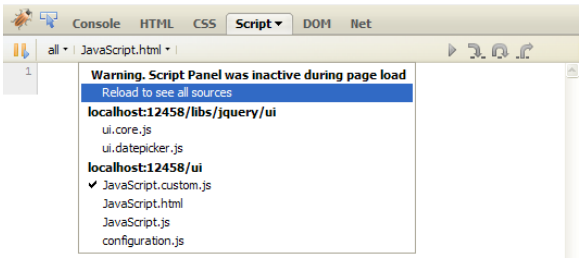
```
function (e2e, $) {
        e2e.controller.registerCustomization("JavaScript", function
(controller) {

                controller.numberOne = controller.e2eGetQuery().getAsString
("numberOne", null);
                controller.numberTwo = controller.e2eGetQuery().getAsString
("numberTwo", null);
                controller.selectedIndex = controller.e2eGetQuery().
getAsString("selectedIndex", null);
                controller.selectedText = controller.e2eGetQuery().
getAsString("selectedText", null);
                controller.selectedValue = controller.e2eGetQuery().
getAsString("selectedValue", null);

                controller.sumJQuery = function() {
                        alert('The sum is: '+ (parseInt
($("#JavaScript_numberOne").val()) +
                                                                        parseInt
($("#JavaScript_numberTwo").val())));
                }
                controller.sumBinding = function() {
                        alert(parseInt(this.numberOne)+ parseInt(this.
numberTwo));
                }
                controller.showSelectedIndex = function() {
                        alert("onChange event reads selected index:"
+$("#JavaScript_aComboBox").attr("selectedIndex"));
                }
                controller.multiply = function(value1, value2) {
                        return value1 * value2;
                }
                controller.setMultiplyResult = function() {
                        var input1 = parseInt($("#JavaScript_numberOne").
val());
                        var input2 = parseInt($("#JavaScript_numberTwo").
val());
                        var prod = this.multiply(input1, input2);
                        $("#JavaScript_productField").val(prod);
                }
                controller.showEventObject = function(event) {
                        alert('Event type:'event.type'\nEvent data: 'event.
data'\nEvent target: '+
                                event.target+'\nEvent timestamp: '+event.
timestamp);
                }
                controller.showSelectedProperties = function() {
                        alert("Selected Properties:\n\nselectedIndex:
"this.selectedIndex
                                "\nselectedText: "this.selectedText"
\nselectedValue: "+this.selectedValue);
                }
        });
}(E2E, jQuery));
```

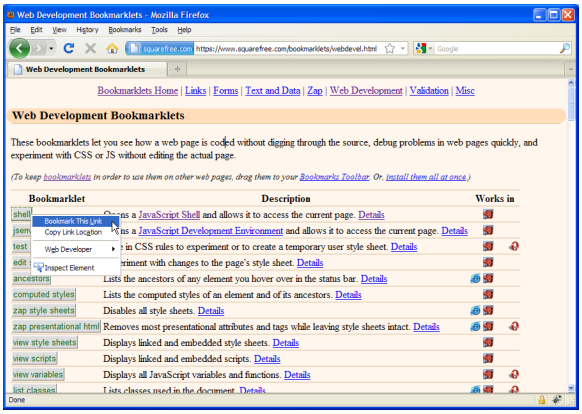
Debugging

This enables selective and easy debugging of the custom JavaScript code using the debugger of your browser. Within the debug console the custom.js file can be chosen and looked at separately e.g. in case of a bug.



JavaScript Shell

Another convenient utility is JavaScript shell. This tool allows to directly execute JavaScript within the context of the loaded web page. The tool is available as a bookmarklet from the following website: https://www.squarefree.com/bookmarklets/webdevel.html. The bookmarklet is added to the bookmarks by right-clicking on the shell button.



When the bookmarklet is called within the loaded xUML UI, it is possible to execute JavaScript code as well as all jQuery code with direct visible result within the currently loaded page.

Figure: JavaScript Shell