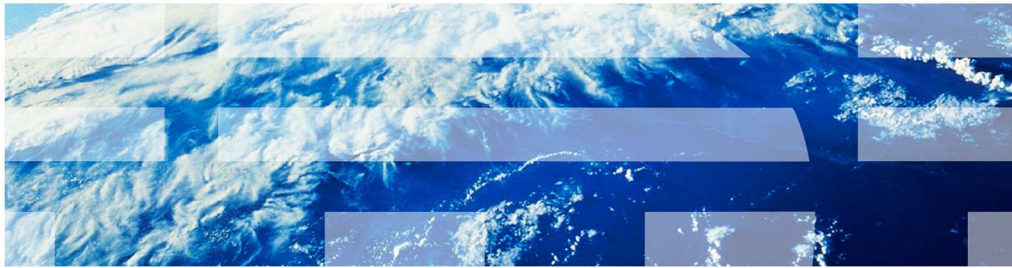


IBM Business Process Manager V8.5

Coaches enhancements



This presentation provides details about the coaches enhancements in IBM Business Process Manager V8.5.

Table of contents

- JavaScript code completion
- Substitution in Custom HTML
- Prototype-level event handlers
- Enhanced visibility support

You will learn about how to use JavaScript code completion to speed up the development of coach views. You will also learn about the enhancements to Custom HTML snippets and how to control the behavior of event handlers to improve performance. Finally, you will learn how to use rules and scripts to control the visibility of items in a coach.



Section

JavaScript code completion

3

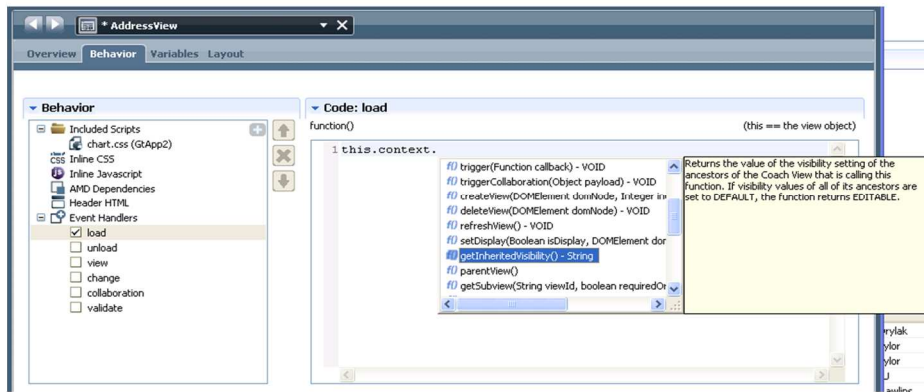
Coaches enhancements

© 2013 IBM Corporation

This section provides details about JavaScript code completion for coaches.

Overview

- Code completion does not occur in previous releases, when you author JavaScript code for the event handlers
- You can now invoke code completion by pressing <ctrl><space>
- Code completion includes a description of the object or function



4

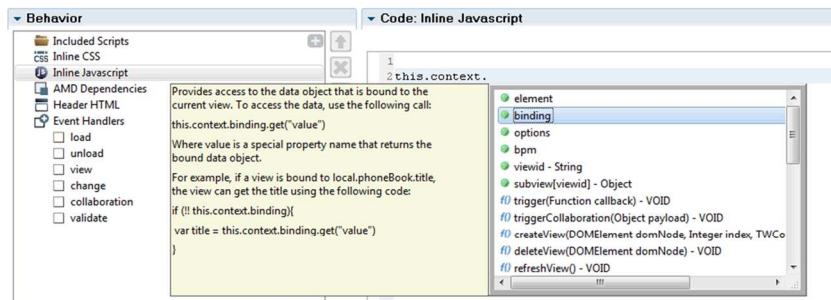
Coaches enhancements

© 2013 IBM Corporation

A documented API is provided for use when you author a coach view. In IBM Business Process Manager V8.5, when you are writing event handler code or in-line JavaScript, it is now possible to invoke code completion in the editor. The code completion will offer the API that can be called at the current cursor position, along with documentation about the API.

Example: this.context

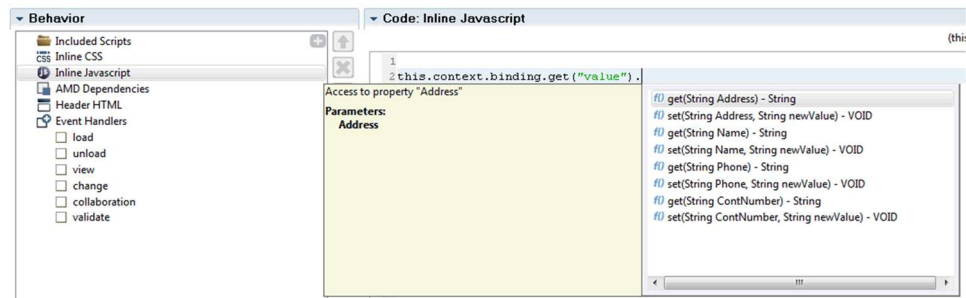
- Content assist for this.context:



This is an example of code completion for the context object. Notice the details about the selected API object, “binding”.

Code completion for complex data types

- Content assist for variable/data type:




The code completion capabilities include navigation of complex data types, making it easy to write code to access any part of the data tree.

Details

- The "context" object provides access to
 - static members (element, bpm, viewid, subview)
 - dynamic members (binding and options)
 - helper functions
- The code completions for dynamic elements (bindings, options):
 - provide convenient access to the entire data-structure, any level deep
 - accommodate any return type
 - support lists
- The "event" object code completion is enabled in the change, collaboration and validate event handlers

The context object has code-completion suggestions for helper functions, binding and option dynamic members, and static members, such as "element". When code completion is invoked for a binding or option, all aspects of the data are covered, including nesting of complex types, all return types, and lists. Code completion is also available for the parameters passed into event handler functions. The "event" parameter on the change, collaboration and validate event handlers can be explored using code completion.

8



Section

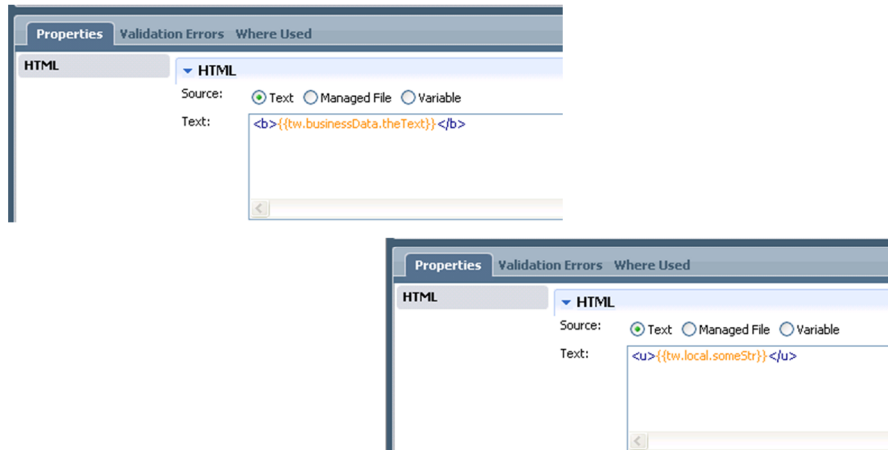
Substitution in custom HTML

8 Coaches enhancements © 2013 IBM Corporation

The next slide describes how to substitute variables in custom HTML within coaches.

Substitution in Custom HTML

- V8.5.0.0 supports easy access of data in the Custom HTML advanced item



9

Coaches enhancements

© 2013 IBM Corporation

In previous releases, when you use custom HTML, it is difficult to access data and incorporate the value of the data into the HTML. In V8.5, the ability to enter substitution variables allows the values of data to be evaluated and embedded in the HTML. The evaluation occurs when the page is generated to be shown and thus is not dynamic. If the value of the data is changed after the client is shown, then the HTML is not updated.



Section

Prototype-level event handlers

10

Coaches enhancements

© 2013 IBM Corporation

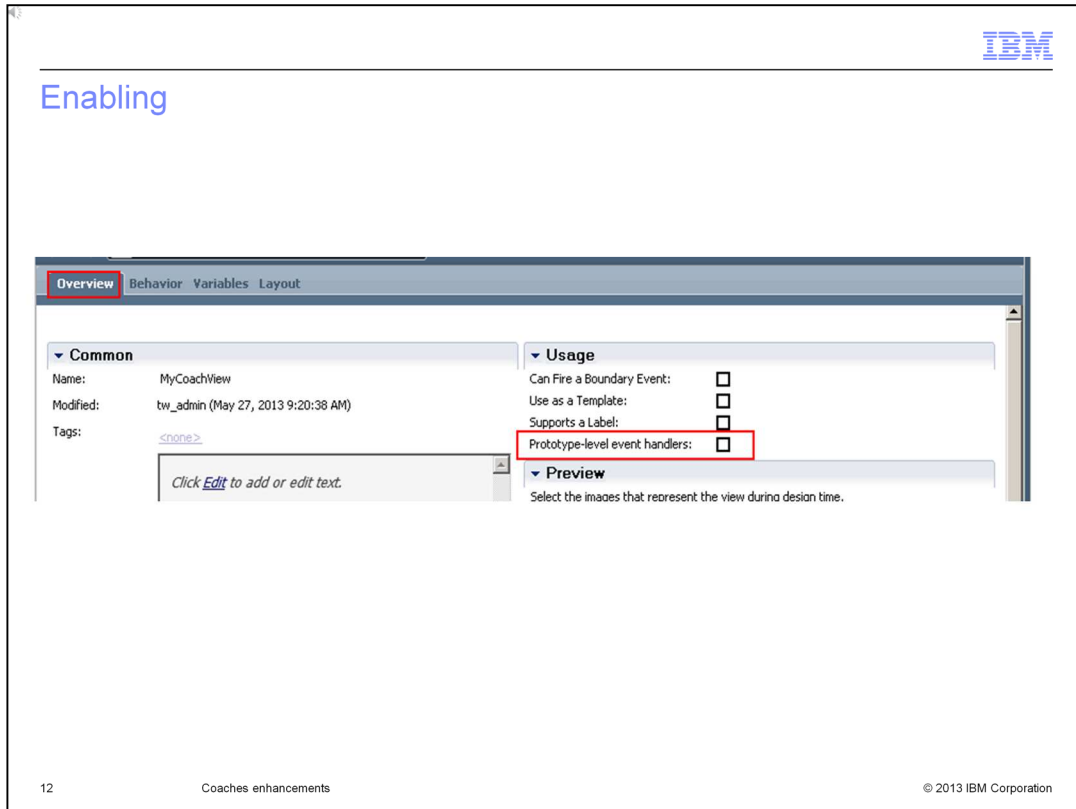
This section describes different prototype-level event handlers within coaches.

Overview

- New in V8.5.0.0 is the ability to declare lifecycle event handlers (load, view, change, unload, validate and collaborate) at the prototype level, instead of instance level
- In previous releases, all lifecycle event handlers are declared at the instance level
 - If multiple instances of the same coach view are used, the same handlers are “instantiated” multiple times, causing heavier memory footprint and potential performance issues
- This feature can be enabled by selecting the “Prototype-level event handlers” check box on the overview tab of the coach view editor
- If this feature is enabled, JavaScript closure technique can no longer be used to access any in-line script

Feature disabled	Feature enabled
Inline JavaScript: <code>var myVariable = null;</code>	Inline JavaScript: <code>this.myVariable = null;</code>
Lifecycle event handlers: <code>myVariable = “newVal”;</code>	Lifecycle event handlers: <code>this.myVariable = “newVal”;</code>

Previously, if multiple instances of the same Coach View existed on the same page, then the event handler functions were copied to each instance. This causes memory and potential performance issues. Now, it is possible to set the event handlers to be attached to the prototype instead of the instance object. This allows reuse of the code across many instances.



To enable prototype-level event handlers, first, on the Overview page of the Coach view editor, select the check box. Second, ensure that code uses object member access instead of closure scope.

IBM

Section

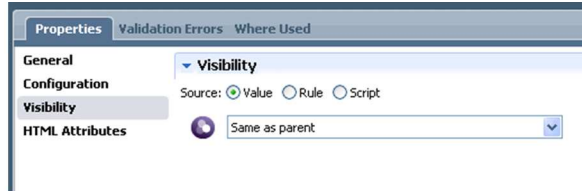
Visibility

13 Coaches enhancements © 2013 IBM Corporation

This section describes visibility, as part of the coaches enhancement.

Introduction

- Visibility in Coaches



- Note the two additional styles for defining visibility – Rule and Script

The visibility of items in coaches is more flexible in V8.5. It now allows rules and scripts to determine visibility at runtime. The visibility properties are now on a separate tab in the Properties window, to accommodate the new styles for specifying visibility.

Rules (1 of 2)

- Rules can be based on variable states
- Rules can be based on the user who is viewing/editing the Coach and his or her membership of a team
- Rules are ordered; first match is applied; if nothing matches then “Otherwise” is applied
 - A logical “OR” can be created by having the same “set to” values with different variables
 - Rules can be reordered
- Rules can include many AND clauses

Visibility

Source: Value Rule Script

Set to: **Editable** When: `privCust.lastName` Condition: Equal to Value: "Taylor"

And: `privCust.firstName` Condition: Equal to Value: "Grant"

Set to: **Read only** When user is: **Member** Team: **All Users**

Otherwise: **Same as parent**

Add rule for:

15

Coaches enhancements

© 2013 IBM Corporation

If you use rules to specify visibility, then the visibility is determined by the state of variables or by who is viewing the task at runtime. Rules can be combined and ordered to create simple “AND” and “OR” statements.

Rules (2 of 2)

- The condition (operator) for each rule is tailored to the variable "type"
 - For example, if the variable is a Boolean, then "less than" does not make sense and will not appear in the drop down
- The "Value" (operand) adapts to the variable type
 - For example, Date Time widget to enable convenient Date selection

▼ Visibility

Source: Value Rule Script

Set to: When: Condition: Value:

And: Condition: Value:

Otherwise:

Add rule for:

May, 2013						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8
Today: 5/28/2013						

– Input validation for Integer, Decimal, Boolean

The Properties page for rules ensures that the values you enter match the type of the variable chosen. The operand adapts to the data, ensuring ease of authoring.

Script (1 of 2)

- Use the Select button to choose variables that the script will use
 - The variables are “watched” for changes and the script runs when changes occur
 - If script area is empty when variables are selected, then a skeleton script is generated to help the user
- Script is client-side only



17

Coaches enhancements

© 2013 IBM Corporation

The second new way of specifying visibility is by using client-side JavaScript. The variables on which the script is operating are chosen and the script is called when any of the values change. A skeleton script is generated after you choose the variables. Scripting provides the most flexible means of specifying visibility.

Script (2 of 2)

- A tooltip is available to describe the parameters of the script function

```
function(context_event_local) (this == the coach object)  
1 // context - data from context.bpm.system, context.bpm.team.member, or context.bpm.team.manager. Member and manager are lists of strings  
2 // identifying the team to which the current user belongs to or is a manager of. An empty list is the default.  
3 if event - data from an initialize or change event. The change event is like the one for the change() event handler with two additional properties:  
4     type: "change"  
5 }     path: the fully qualified name of the variable that changed such as "local.employee.phoneNumber[2].areaCode"  
6 ret local - data from information stored in any Coach variable. For example, to get a variable value, use a call like local.get("employee").get("phoneNumber").get(0).get("type")
```

The Properties page for the script style of visibility provides help through a tooltip. It helps you understand the parameters that are passed into the script code.



Section

Summary

19

Coaches enhancements

© 2013 IBM Corporation

The next slide provides a summary of this presentation on the coaches enhancements in IBM Business Process Manager V8.5.

Summary

- You can use JavaScript code completion to quickly learn the coach API and to speed up authoring of code for your coach views
- Substitution variables can be used in Custom HTML items to create more dynamic user interfaces using the values of data present in the runtime context
- You can reduce memory consumption and improve performance by leveraging prototype-level event handlers on coach views
- The visibility of items in coaches can be described using client-side JavaScript and simple rules

You learned about the new JavaScript code-completion capabilities and about using substitution variables in Custom HTML items. You also learned how to improve performance using prototype-level event handlers and how to specify the visibility of coach items, using both scripts and rules.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

1. Did you find this module useful?
2. Did it help you solve a problem or answer a question?
3. Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_BPMV85_Coaches.ppt

This module is also available in PDF format at: [../BPMV85_Coaches.pdf](..../BPMV85_Coaches.pdf)

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2013. All rights reserved.