



IBM Software Group

IBM WebSphere Application Server Feature Pack for Communications Enabled Applications

Two-way forms widget



@business on demand.

© 2009 IBM Corporation
Updated July 30, 2009

This presentation will discuss the IBM WebSphere® Application Server Feature Pack for Communications Enabled Applications two way forms widget.

Section

Two-way forms



This section will discuss the general concept of two-way forms.

Two-way forms

- Two-way forms allow two parties in a co-browsing session to both interact with a form in the Web page
- Typical usage
 - ▶ Customer service representative (CSR) and customer co-browsing a Web page
 - ▶ CSR helps customer complete a form in the Web page
 - ▶ Both parties can see the form and CSR typically fills in information provided by the customer
 - ▶ Customer can fill in sensitive data themselves (for example credit card information)

When two parties are in a co-browsing session, using the CEA feature pack collaboration dialog widget to share a Web browsing session, they can both view pages that contain forms. The two way form widget allows the parties in the co-browsing session to both input information into the same form in the viewed Web page.

A typical example of when this feature can be used, is the scenario where a customer service representative is helping a customer navigate a Web site. A customer is on the telephone with the customer service representative, and they are both viewing the same Web page in a co-browsing session. The customer service representative can help the customer to fill in information in an online form that they are both viewing.

Built in features of two-way forms allow restrictions to be places on form input elements. A customer service representative, for example, can be denied access to a credit card input box on the Web site form. allowing only the customer to enter the data into the credit card entry fields in the form. The characters of the credit card entry fields in the two-way form can be masked. The customer service representative can never see the numbers themselves if they are masked, only a masking value indicating that characters have been typed into the two-way form field.

Setting up a two-way form

- 1) Import CEA Dojo toolkit
 - ▶ `<script type="text/javascript" src="ceadojo/dojo/dojo.js"`
 - ▶ `djconfig="parseOnLoad: true, isDebug:`
 - ▶ `false"></script>`
- 2) Import the two-way form CSS definitions
 - ▶ `<style> @import`
 - ▶ `"<contextRoot>/ceadojo/cea/widget/TwoWayForm/TwoWayForm.`
 - ▶ `css"; </style>`
- 3) Create a 'Form' element with a ceadojoType two-way form attribute
 - ▶ `<form ceadojoType="cea.widget.TwoWayForm"`
 - ▶ `name="myTwoWayForm" id="collabID"`
 - ▶ `ceaCollabOnSubmit="return myAction()" />`



During initial setup of the CEA Dojo toolkit, the ceadojo directory should be imported into the Web content directory of your Web project from the `<was_home>/feature_packs/cea/javascript` directory. Then, a two-way form is added to a Web page in a few steps.

First, the CEA Dojo toolkit is imported on the page using the script tag. Secondly, a style tag should be added to the top of the Web page to import the two-way forms widget style sheet. Finally, a form element should be added to the Web page. The form element needs to have the CEA Dojo type `cea.widget.TwoWayForm`, and must have a name and an ID.

The `ceaCollabOnSubmit` attribute in this example hooks the form element into the submit action for the form.

Developers must use this attribute instead of the typical 'onSubmit' attribute of regular forms.

Two-way forms

- The CEA two-way form widget requires the use of Dojo libraries
 - ▶ CEA Dojo Javascript library
 - ▶ Dijit form widget library
- Any widget supporting two-way editing functionality must have an ID specified



The CEA two-way forms widget is included in the CEA Dojo toolkit provided with the IBM WebSphere Feature Pack for Communication Enabled Applications.

The CEA Dojo toolkit requires the use of the Dijit form widget library provided in the <was_home>/feature_packs/cea/javascript/ceadojo directory that ships in the WebSphere Feature Pack for Communication Enabled Applications.

Any widget that is used to support the two-way editing functionality must have an ID specified.

Two-way form field masking

- Field values can be masked for form input elements such as textBox
- A JavaScript method can be implemented to replace typed characters with a custom mask
 - ▶ `<input type="text" name="textName" id="textName" value="" size="30" dojoType="dijit.form.TextBox" ceaCollabFilter="mask" />`
 - ▶ `function mask(value) { return "XXXX"; }`



When two parties are co-browsing a Web page that contains forms and form data it can be important to mask the values from a form input field from one or both of the parties viewing the page. Two-way forms have the ability to mask the values of an input field to someone viewing the form in a co-browse session.

You can implement a small JavaScript method that will return the masking character of your choice whenever data is input into a two-way form field. The `ceaCollabFilter` attribute is added to the form input field definition to enable the masking.

Two-way form field change notification

- When any field in a two-way form changes, a notification alerts the user
- By default, the modified field is highlighted
- Default action is overridden by adding two attributes to the form input field
 - ▶ **ceaCollabHandleRemoveNotification**
 - ▶ **ceaCollabHandleShowNotification**



Another feature of the two-way forms widget is the ability to notify a user if the other party has entered information into a form field.

When co-browsing a Web page that has form data, it might not be obvious to the co-browsing party that the other co-browser has entered data into a form field. You can enable field change notification in a two-way form to highlight a form field that has been modified.

For example, the customer service representative that is helping a customer to navigate a form on a co-browsed Web page can see that the customer has entered his name at the top of a form, because the form field will become highlighted to alert them.

Two-way form field change notification

- The notification shown to you when a two-way form field is changed can be customized
- Example: instead of highlighting the field that changed, show a tooltip using JavaScript methods
 - ▶ `function removeFunc(widget) { dijit.hideTooltip (widget.domNode); }`
 - ▶ `function showFunc(widget) { dijit.showTooltip ("Value changed", widget.domNode); }`
 - ▶ `<input type="text" name="textName" id="textName" value="" size="30" dojoType="dijit.form.TextBox" ceaCollabHandleRemoveNotification="removeFunc" ceaCollabHandleShowNotification="showFunc" />`

11

Two way forms widget

© 2009 IBM Corporation

You can implement different functionality to customize your own form field change notification.

The styling on the highlighting of the fields is dictated by the `TwoWayForm.css` file provided with the CEA feature pack. The styling can be changed in the style sheet file to modify the field change notification behavior.

Also, methods can be added to your Web pages to handle the field change notification behavior, for example if you wanted to have a tool tip displayed when a two-way form field is changed instead of having it highlighted.

The `ceaCollabHandleRemoveNotification` and `ceaCollabHandleShowNotification` attributes need to be modified so their values refer to the JavaScript methods that you want to have called to handle the form field change notification.

Two-way form field validation

- Two-way form fields can have the input validated
- A change to the input field by one party will require another party to accept or decline the changes if validation is enabled
- A validation widget opens alongside an input field widget when validation is required
- To enable validation on a two-way form field, set the value of the "**ceaCollabValidation**" attribute on the input field widget in question to "**default**"
 - ▶ `<input type="text" name="textName" id="textName" value="" size="30" dojoType="dijit.form.TextBox" ceaCollabValidation="default" />`



There is a feature of two-way forms that allows users to validate the data input into form fields. For example if a customer service representative was helping a customer enter data into a two-way form, the customer might want to make sure that the information the customer service representative entered for them is correct.

When a party enters data into a two-way form that needs validation by the other co-browser, an icon opens beside the entry field in the two-way form. The user must click the icon to indicate that the information is valid. You can enable this form field validation by adding the `ceaCollabValidation` attribute to the input field in the two-way form definition.

Restricting access to two-way form input fields

- Ability to input values into fields in a two-way form can be restricted
- For example a customer service representative should not be able to type information into a credit card entry box in a two-way form
- The input field widget can set the **ceaCollabWriteAccess** attribute
- Two values for this attribute dictate who can edit the field
 - ▶ Reader (party without control when the form is launched)
 - ▶ Writer (party with control when the form is launched)
- ```
<input type="text" name="textName" id="textName" value="" size="30" ceadojoType="dijit.form.TextBox" ceacollabwriteaccess="reader" />
```

13

Two way forms widget

© 2009 IBM Corporation

Access to two-way form fields can be restricted as well. Two-way forms can have entry fields that should only have data input by the customer, or by the customer service representative. You can restrict the input access to one of the co-browsing parties by using the **ceaCollabWriteAccess** attribute on your input field in your two-way form. Using that attribute, you can specify if the reader, meaning the party with no control of the Web collaboration session, has access to enter data into the form. Alternatively, you can specify that the writer, the party with control of the Web collaboration session, should have restricted access to enter data into the two-way form field.

This feature can be used in conjunction with the **ceaCollabValidation** attribute to ensure that only **one** party can change a particular field in a two-way form.

## Section

# *Summary*



This section will summarize the two-way forms widget presentation.

## Summary

- Two-way forms
  - ▶ Forms can be created that both parties involved in a co-browsing session can access
  - ▶ Form fields can be validated by users and marked as valid
  - ▶ Form content can be masked from either user using either default masking or custom masking characters
  - ▶ Form fields can be restricted for writing access by only one user
  - ▶ Change notification notifies visually that a form field has changed



The two-way forms widget allows two parties in a co-browsing session to enter data into the same Web page form. The Web page form must be designated as a `ceaDojoType` two-way form, and field input and access can be restricted to the co-browsing parties.

Form entry data can be validated and can be masked from one user or the other using the validation and masking features. A visual notification to the parties in the co-browsing session that the other party has entered data into the form can be enabled.

## Feedback

### Your feedback is valuable

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

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

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_CEAFP\\_TwoWayFormsWidget.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_CEAFP_TwoWayFormsWidget.ppt)

This module is also available in PDF format at: [../CEAFP\\_TwoWayFormsWidget.pdf](http://.../CEAFP_TwoWayFormsWidget.pdf)



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



## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. 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.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

