IBM Pervasive Computing
January 2003

Developing Multimodal Applications using XHTML+Voice

Contents

3 ***Structure of an XHTML+Voice Application***

3 ***Namespace Declaration***

4 ***Visual Part***

5 ***Voice Part***

6 ***External Speech Grammar Files***

7 ***Phonetic Representations***

7 ***Processing Part***

8 ***Flow of Events***

9 ***Pizza Order Application***

14 ***For More Information***

Multimodal applications use multiple methods of communication between a user and device.

Executive Summary

On the Internet, people use browsers to visit Web sites, access documents from networks, and fill out forms. With this growing capability to retrieve information, communications between users and their devices is receiving more attention. As devices become smaller, other means of input -- in addition to keyboard or tap screen -- are becoming necessary. Small handheld devices, including cell phones and PDA's, now contain sufficient processing power to handle multiple tasks. On some devices it is difficult to perform these tasks using only keyboard, stylus, or handwriting recognition. This has lead to a new application technology called *multimodal*, the use of multiple methods of communication between the user and a device. These methods include keypad, touch or tap screen, handwriting recognition, and voice recognition.

This paper illustrates the basic structure and contents of an XHTML+Voice multimodal application, describing its fundamental building blocks. It is intended for those who are familiar with XHTML, VoiceXML, and HTML.

Each of the building blocks are described and coding samples are provided. A multimodal implementation of a hypothetical Pizza Order Form application is presented as an example.

The Structure of an XHTML+Voice Application

A basic XHTML+Voice multimodal application consists of a *Namespace Declaration*, *Visual Part*, *Voice Part*, and a *Processing Part*. *Figure 1* illustrates these components and their relationship to each other.

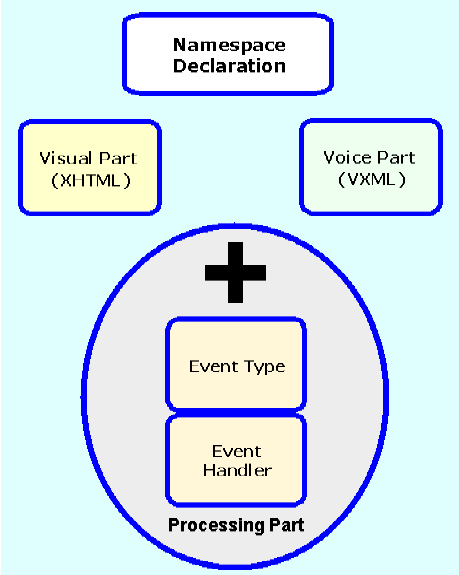

Figure 1 -- Components of an XHTML+Voice

Namespace Declaration

The *Namespace Declaration* for a typical XHTML+Voice application is written in XHTML, with additional declarations for VoiceXML, and XML-events. *Figure 2* is an example of the namespace declaration for an XHTML+Voice application.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML+Voice 1.0/EN" "xhtml+voice.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:ev="http://www.w3.org/2001/xml-events"
xmlns:vxml="http://www.w3.org/2001/vxml" xml:lang="en_US" >
```

Figure 2 -- Namespace declaration

Visual Part

The *Visual Part* of an XHTML+Voice application is XHTML code that is used to display the various form elements to the device's screen, if available. This can be ordinary XHTML code and may include check boxes and other form items that are found in a typical form. For example, *Figure 3* displays the pizza size

```
  <b>Size:</b><br/>
  <input type="radio" name="size" id="sizeSmall" ev:event="focus"
ev:handler="#voice_size"/>
      Small 12&quot;
  <input type="radio" name="size" id="sizeMedium" ev:event="focus"
ev:handler="#voice_size"/>
      Medium 16&quot;
  <input type="radio" name="size" id="sizeLarge" ev:event="focus"
ev:handler="#voice_size"/>
      Large 22&quot;
```

Figure 3 -- Visual part of a multimodal application

choices and their appropriate radio buttons. *Figure 4* illustrates a typical form using XHTML+Voice.

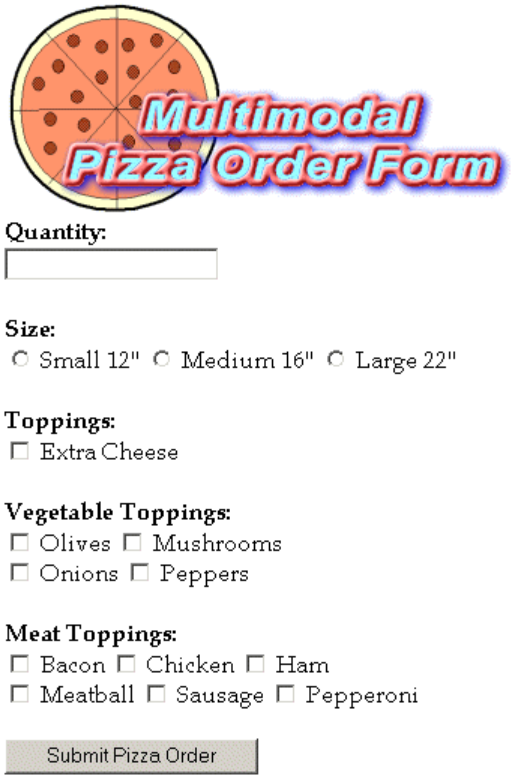

Figure 4 -- Input form for sample application

Voice Part

The *Voice Part* of an application is the section of code that is used to prompt the user for a desired field within a form. This VoiceXML code utilizes an external *grammar* to define the possible field choices. If there are many choices, or a combination of choices is required, the external grammar can be used to handle the valid combinations.

For example, to select the vegetable toppings for a pizza, there are multiple ways to say the selections. The VoiceXML code in *Figure 5* is used with the *vegtoppings.jsgf* grammar file to prompt the user to select vegetable toppings for the pizza. To add additional vegetable topping choices, modify the *vegtoppings.jsgf* file.

size.jsgf

```
grammar pizza_size;
public <size> = small|medium|large ;
```

Figure 5 -- Voice part of a multimodal application

```
<vxml:form id="voice_toppings_vegetable">
    <vxml:field name="vtoppingsvegetable">
      <vxml:prompt>
        What vegetable toppings would you like?
      </vxml:prompt>
      <vxml:grammar src="vegtoppings.jsgf"/>
      <vxml:catch event="help nomatch noinput">
        For example, say mushrooms and peppers.
      </vxml:catch>
      <vxml:filled>
        <vxml:assign name="varVegetableToppings" expr="vtoppingsvegetable"/>
      </vxml:filled>
    </vxml:field>
  </vxml:form>
```

Figure 6 -- How to specify an external grammer

External Speech Grammar Files

External speech grammar files are used to define specific selections. For example, the vegetable toppings for the form are in the *vegtoppings.jsgf* file and the meat toppings are in the *meattoppings.jsgf* file. These grammar files are shown in *Figures 7, 8 and 9*.

vegtoppings.jsgf

```
grammar vegitable_toppings;
<veggies> = olives | mushrooms | onions | peppers;
public <toppings> =
    <NULL> {
              this.$value="";
    }
    (<veggies> [and]{
              this.$value = this.$value + " " + $veggies;
    })+;
```

Figure 7 -- Grammars for vegetable toppings

onetotwenty.jsgf

```
grammar one_twenty;
public <onetotwenty> =  1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20 ;
```

Figure 8 -- Grammars for number choice of 1-20

meattoppings.jsgf

```
grammar meat_toppings;
<meats> = bacon | chicken | ham | meatball | sausage | pepperoni;
public <toppings> =
   <NULL> {
               this.$value="";
   }
   (<meats> [and]{
               this.$value = this.$value + " " + $meats;
   })+;
```

Figure 9 -- Grammars for meat toppings

Phonetic Representations

The application also needs the phonetic pronunciations for the words in the grammar. The *pizza.pbs* file contains the words and their phonetic representations.

```
pizza.pbs
small          S M AO L
medium   M IY D IY AH M
large          L AA R JH
yes      Y EH S
no       N OW
and      AX N DD
olives   AA L IH V Z
mushroomsM AH SH R UW M Z
onions   AH N Y AH N Z
peppers  P EH P ER Z
bacon          B EY K AH N
chicken  CH IH K AH N
ham      HH AE M
meatball M IY T B AO L
sausage  S AO S AH JH
sausage  S AO S IH JH
pepperoniP EH P ER OW N IY
```

Processing Part

The *Processing Part* of the application contains the code that is used to perform the needed instructions for each of the various events. This section also contains the event handlers. For example, the quantity of pizzas ordered requires the event handlers shown in *Figure 10*.

```
<b>Quantity:</b><br/>
<input type="text" id="pizzaQuantity" ev:event="focus" ev:handler="#voice_quantity"/>
<ev:listener ev:event="vxmldone" ev:handler="#handleVoiceQuantityDone"
ev:observer="pizzaQuantity" ev:propagate="stop"/>
```

Figure 10 -- Processing part

Now that the basic components are present (visual, voice, and processing), *Figure 11* illustrates the logic and code used to enable the pizza quantity field.

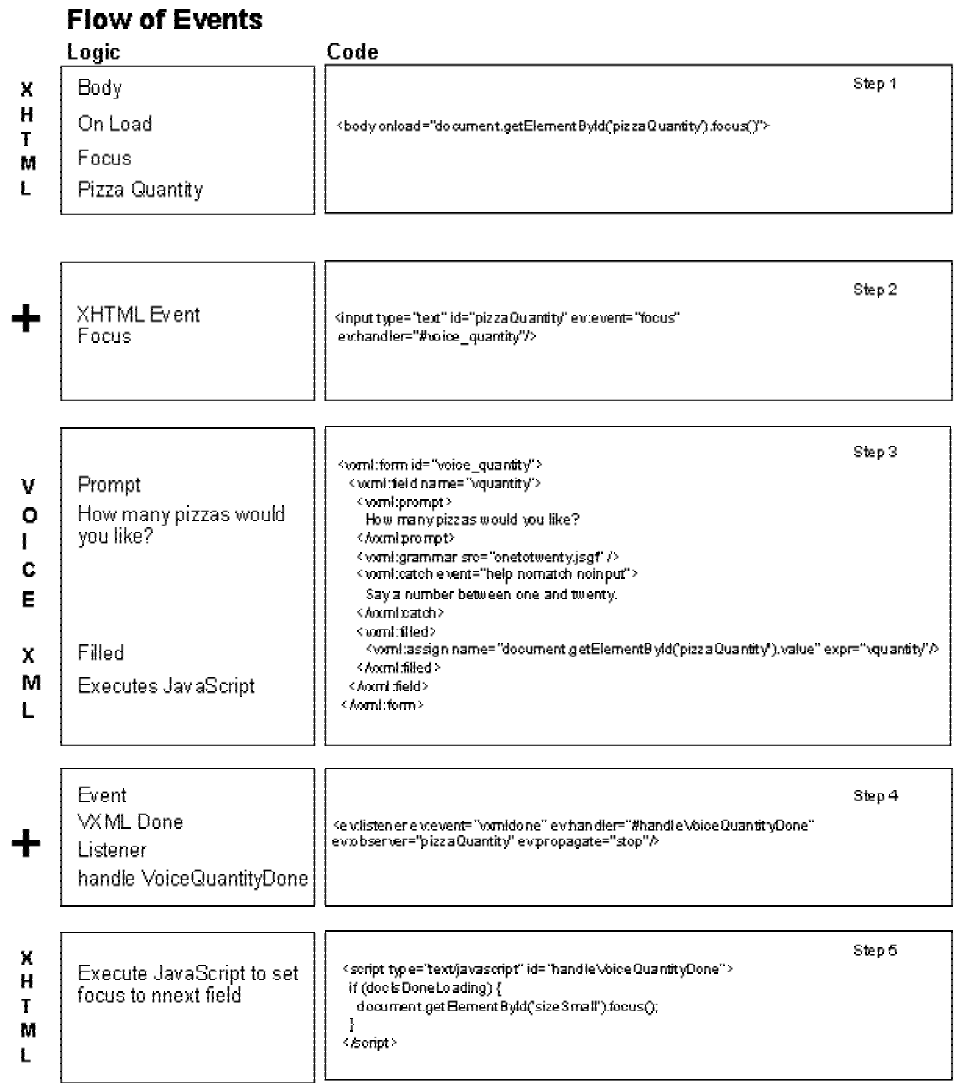

Figure 11 -- Code for enabling the quantity field

Pizza Order Form: an Example of an XHTML+Voice Application

The following is the entire code example of the pizza order form.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML+Voice 1.0/EN" "xhtml+voice.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:ev="http://www.w3.org/2001/xml-events"
xmlns:vxml="http://www.w3.org/2001/vxml" xml:lang="en_US" >
<!--
***************** Copyright Notice **************************
pizza.mxml

(C) COPYRIGHT International Business Machines Corp. 2001,2002

This program may be used, executed, copied, modified and
distributed without royalty for the purpose of developing,
using, marketing, or distributing.
*************************************************************
-->
 <head>
   <title>Pizza</title>

   <script type="text/javascript">
     var docIsDoneLoading = false;
     var varPizzaSize = undefined;
     var varExtraCheese = undefined;
     var varVegetableToppings = undefined;
     var varMeatToppings = undefined;
   </script>

   <script type="text/javascript" id="handleVoiceQuantityDone">
     if (docIsDoneLoading) {
       document.getElementById('sizeSmall').focus();
     }
   </script>

   <script type="text/javascript" id="handleVoicePizzaSizeDone">
     if (docIsDoneLoading) {
       if (varPizzaSize != undefined) {
         document.getElementById('sizeSmall').checked = false;
         document.getElementById('sizeMedium').checked = false;
         document.getElementById('sizeLarge').checked = false;

       if (varPizzaSize == "small")
         document.getElementById('sizeSmall').checked = true;
   else if (varPizzaSize == "medium")
         document.getElementById('sizeMedium').checked = true;
       else if (varPizzaSize == "large")
         document.getElementById('sizeLarge').checked = true;

       varPizzaSize = undefined;
       document.getElementById('toppingsExtraCheese').focus(); }
     }
   </script>

   <script type="text/javascript" id="handleVoiceToppingsExtraDone">
     if (docIsDoneLoading) {
       if (varExtraCheese != undefined) {
         if (varExtraCheese == "true")
           document.getElementById('toppingsExtraCheese').checked = true;
         else
           document.getElementById('toppingsExtraCheese').checked = false;

         varExtraCheese = undefined;
         document.getElementById('vegetableOlives').focus();
       }
     }
   </script>

   <script type="text/javascript" id="handleVoiceToppingsVegetableDone">
     if (docIsDoneLoading) {
       if (varVegetableToppings != undefined) {
         document.getElementById('vegetableOlives').checked = false;
         document.getElementById('vegetableMushrooms').checked = false;
```

```
                document.getElementById('vegetableOnions').checked = false;
                document.getElementById('vegetablePeppers').checked = false;

                if (varVegetableToppings.search(/olives/i) != -1)
                  document.getElementById('vegetableOlives').checked = true;
                if (varVegetableToppings.search(/mushrooms/i) != -1)
                  document.getElementById('vegetableMushrooms').checked = true;
                if (varVegetableToppings.search(/onions/i) != -1)
                  document.getElementById('vegetableOnions').checked = true;
                if (varVegetableToppings.search(/peppers/i) != -1)
                  document.getElementById('vegetablePeppers').checked = true;

                varVegetableToppings = undefined;
                document.getElementById('meatBacon').focus();
              }
            }
          </script>

          <script type="text/javascript" id="handleVoiceToppingsMeatDone">
            if (docIsDoneLoading) {
              if (varMeatToppings != undefined) {
                document.getElementById('meatBacon').checked = false;
                document.getElementById('meatChicken').checked = false;
                document.getElementById('meatHam').checked = false;
                document.getElementById('meatMeatball').checked = false;
                document.getElementById('meatSausage').checked = false;
                document.getElementById('meatPepperoni').checked = false;

                if (varMeatToppings.search(/bacon/i) != -1)
                  document.getElementById('meatBacon').checked = true;
                if (varMeatToppings.search(/chicken/i) != -1)
                  document.getElementById('meatChicken').checked = true;
                if (varMeatToppings.search(/ham/i) != -1)
                  document.getElementById('meatHam').checked = true;
                if (varMeatToppings.search(/meatball/i) != -1)
                  document.getElementById('meatMeatball').checked = true;
                if (varMeatToppings.search(/sausage/i) != -1)
                  document.getElementById('meatSausage').checked = true;
                if (varMeatToppings.search(/pepperoni/i) != -1)
                  document.getElementById('meatPepperoni').checked = true;

                varMeatToppings = undefined;
                document.getElementById('submitButton').focus();
              }
            }
          </script>

          <script type="text/javascript">
            function displayPizzaOrder() {
              var order = "Your pizza order is: " +
        document.getElementById('pizzaQuantity').value;
              var totToppings = countToppings();
              var numToppings = totToppings;

              if (document.getElementById('sizeSmall').checked)
                order += " small";
              else if (document.getElementById('sizeMedium').checked)
                order += " medium";
              else if (document.getElementById('sizeLarge').checked)
                order += " large";

              if (document.getElementById('pizzaQuantity').value > 1)
                order += " pizzas";
              else
                order += " pizza";

              if (numToppings > 0) {

                order += " with";

                if (document.getElementById('toppingsExtraCheese').checked) {
                  order += " extra cheese";
                  numToppings--;
                  order += appendComma( totToppings, numToppings );
                }

                if (document.getElementById('vegetableOlives').checked) {
                  order += " olives";
                  numToppings--;
                  order += appendComma( totToppings, numToppings );
                }
```

```
      if (document.getElementById('vegetableMushrooms').checked) {
        order += " mushrooms";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('vegetableOnions').checked) {
        order += " onions";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('vegetablePeppers').checked) {
        order += " peppers";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }

      if (document.getElementById('meatBacon').checked) {
        order += " bacon";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('meatChicken').checked) {
        order += " chicken";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('meatHam').checked) {
        order += " ham";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('meatMeatball').checked) {
        order += " meatball";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('meatSausage').checked) {
        order += " sausage";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
      if (document.getElementById('meatPepperoni').checked) {
        order += " pepperoni";
        numToppings--;
        order += appendComma( totToppings, numToppings );
      }
    }

    order += ".";

    alert( order );
  }

  function countToppings() {
    var numToppings = 0;

    if (document.getElementById('toppingsExtraCheese').checked)
      numToppings++;

    if (document.getElementById('vegetableOlives').checked)
      numToppings++;
    if (document.getElementById('vegetableMushrooms').checked)
      numToppings++;
    if (document.getElementById('vegetableOnions').checked)
      numToppings++;
    if (document.getElementById('vegetablePeppers').checked)
      numToppings++;

    if (document.getElementById('meatBacon').checked)
      numToppings++;
    if (document.getElementById('meatChicken').checked)
      numToppings++;
    if (document.getElementById('meatHam').checked)
      numToppings++;
    if (document.getElementById('meatMeatball').checked)
      numToppings++;
    if (document.getElementById('meatSausage').checked)
      numToppings++;
    if (document.getElementById('meatPepperoni').checked)
      numToppings++;
```

```
      return numToppings;
    }

    function appendComma( toppingsTotal, toppingsCount ) {
      var commaStr = "";

      if (toppingsCount > 1) {
        commaStr = ",";
      } else if (toppingsCount > 0) {
        if (toppingsTotal > 2)
          commaStr = ",";
        commaStr = " and";
      }

      return commaStr;
    }
  </script>

  <script type="text/javascript">docIsDoneLoading = true;</script>

  <vxml:form id="voice_quantity">
    <vxml:field name="vquantity">
      <vxml:prompt>How many pizzas would you like?</vxml:prompt>
      <vxml:grammar src="onetotwenty.jsgf" />
      <vxml:catch event="help nomatch noinput">Say a number between one
and twenty.</vxml:catch>
      <vxml:filled>
        <vxml:assign
name="document.getElementById('pizzaQuantity').value" expr="vquantity" />
      </vxml:filled>
    </vxml:field>
  </vxml:form>

  <vxml:form id="voice_size">
    <vxml:field name="vsize">
      <vxml:prompt>What size of pizza would you like?</vxml:prompt>
      <vxml:grammar src="size.jsgf" />
      <vxml:catch event="help nomatch noinput">Say small, medium, or
large.</vxml:catch>
      <vxml:filled>
        <vxml:assign name="varPizzaSize" expr="vsize" />
      </vxml:filled>
    </vxml:field>
  </vxml:form>

  <vxml:form id="voice_toppings_extra">
    <vxml:field name="vtoppingsextra" type="boolean">
      <vxml:prompt>Would you like extra cheese?</vxml:prompt>
      <vxml:catch event="help nomatch noinput">Say yes or
no.</vxml:catch>
      <vxml:filled>
        <vxml:assign name="varExtraCheese" expr="vtoppingsextra" />
      </vxml:filled>
    </vxml:field>
  </vxml:form>

  <vxml:form id="voice_toppings_vegetable">
    <vxml:field name="vtoppingsvegetable">
      <vxml:prompt>What vegetable toppings would you like?</vxml:prompt>
      <vxml:grammar src="vegtoppings.jsgf" />
      <vxml:catch event="help nomatch noinput">For example, say mushrooms
and peppers.</vxml:catch>
      <vxml:filled>
        <vxml:assign name="varVegetableToppings"
expr="vtoppingsvegetable" />
      </vxml:filled>
    </vxml:field>
  </vxml:form>

  <vxml:form id="voice_toppings_meat">
    <vxml:field name="vtoppingsmeat">
      <vxml:prompt>What meat toppings would you like?</vxml:prompt>
      <vxml:grammar src="meattoppings.jsgf" />
      <vxml:catch event="help nomatch noinput">For example, say sausage
and pepperoni.</vxml:catch>
      <vxml:filled>
        <vxml:assign name="varMeatToppings" expr="vtoppingsmeat" />
      </vxml:filled>
    </vxml:field>
  </vxml:form>
 </head>
```

```
 <body onload="document.getElementById('pizzaQuantity').focus()">

   <form onsubmit="displayPizzaOrder()">
     <img src="mmpizza.gif" border="0" alt=""/><br/>

     <b>Quantity:</b><br/>
     <input type="text" id="pizzaQuantity" ev:event="focus"
ev:handler="#voice_quantity"/>
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceQuantityDone" ev:observer="pizzaQuantity"
ev:propagate="stop" />
     <br/><br/>

     <b>Size:</b><br/>
     <input type="radio" name="size" id="sizeSmall" ev:event="focus"
ev:handler="#voice_size"/>
       Small 12&amp;quot;
     <input type="radio" name="size" id="sizeMedium" ev:event="focus"
ev:handler="#voice_size"/>
       Medium 16&amp;quot;
     <input type="radio" name="size" id="sizeLarge" ev:event="focus"
ev:handler="#voice_size"/>
       Large 22&amp;quot;
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoicePizzaSizeDone" ev:observer="sizeSmall"
ev:propagate="stop" />
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoicePizzaSizeDone" ev:observer="sizeMedium"
ev:propagate="stop" />
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoicePizzaSizeDone" ev:observer="sizeLarge"
ev:propagate="stop" />
     <br/><br/>

     <b>Toppings:</b><br/>
     <input type="checkbox" id="toppingsExtraCheese" ev:event="focus"
ev:handler="#voice_toppings_extra"/>
       Extra Cheese
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsExtraDone"
ev:observer="toppingsExtraCheese" ev:propagate="stop" />
     <br/><br/>

     <b>Vegetable Toppings:</b><br/>
     <input type="checkbox" id="vegetableOlives" ev:event="focus"
ev:handler="#voice_toppings_vegetable"/>
       Olives
     <input type="checkbox" id="vegetableMushrooms" ev:event="focus"
ev:handler="#voice_toppings_vegetable"/>
       Mushrooms
     <br/>
     <input type="checkbox" id="vegetableOnions" ev:event="focus"
ev:handler="#voice_toppings_vegetable"/>
       Onions
     <input type="checkbox" id="vegetablePeppers" ev:event="focus"
ev:handler="#voice_toppings_vegetable"/>
       Peppers
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsVegetableDone"
ev:observer="vegetableOlives" ev:propagate="stop" />
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsVegetableDone"
ev:observer="vegetableMushrooms" ev:propagate="stop" />
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsVegetableDone"
ev:observer="vegetableOnions" ev:propagate="stop" />
     <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsVegetableDone"
ev:observer="vegetablePeppers" ev:propagate="stop" />
     <br/><br/>

     <b>Meat Toppings:</b><br/>
     <input type="checkbox" id="meatBacon" ev:event="focus"
ev:handler="#voice_toppings_meat"/>
       Bacon
     <input type="checkbox" id="meatChicken" ev:event="focus"
ev:handler="#voice_toppings_meat"/>
       Chicken
     <input type="checkbox" id="meatHam" ev:event="focus"
ev:handler="#voice_toppings_meat"/>
       Ham
```

```
      <br/>
      <input type="checkbox" id="meatMeatball" ev:event="focus"
ev:handler="#voice_toppings_meat"/>
        Meatball
      <input type="checkbox" id="meatSausage" ev:event="focus"
ev:handler="#voice_toppings_meat"/>
        Sausage
      <input type="checkbox" id="meatPepperoni" ev:event="focus"
ev:handler="#voice_toppings_meat"/>
        Pepperoni
      <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsMeatDone" ev:observer="meatBacon"
ev:propagate="stop" />
      <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsMeatDone" ev:observer="meatChicken"
ev:propagate="stop" />
      <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsMeatDone" ev:observer="meatHam"
ev:propagate="stop" />
      <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsMeatDone" ev:observer="meatMeatball"
ev:propagate="stop" />
      <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsMeatDone" ev:observer="meatSausage"
ev:propagate="stop" />
      <ev:listener ev:event="vxmldone"
ev:handler="#handleVoiceToppingsMeatDone" ev:observer="meatPepperoni"
ev:propagate="stop" />
      <br/><br/>

      <input type="submit" name="submit" id="submitButton" value="Submit
Pizza Order" />
    </form>
  </body>
</html>
```

For more information

For additional information and a full description of the programming interface used with XHTML+Voice , visit the http://www.w3.org/TR/xhtml+voice/ Web site.

Visit http://www.w3.org/TR/xhtml+voice/ for additional information on events and event handlers.

To learn more information about IBM Pervasive computing software vist **ibm.com/**pvc. Requests for technical information about IBM products can also be made to your IBM reseller or IBM marketing representative.

IBM Corporation
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
01-03