

第9章 VoiceXML的输入和输出命令

本章将要介绍通过文本语音转换（TTS）工具将信息输出给用户和播放声音文件时使用的VoiceXML元素，本章将要详细讲述如何通过语音识别和使用称为双音多频（DTMF）系统接收电话按键信息等方法来收集来自用户的输入信息。本章还要介绍如何妥善处理与输入相关的出错条件。本章详细说明了每个元素的属性和属性可以使用的值。我们将要使用本章介绍的各种标记来进一步建立语音商务站点。

9.1 VoiceXML中的输出方法

在VoiceXML中，有两种方法可以用来将信息输出给用户。第一种方法是通过文本语音转换（TTS）工具进行输出。另一种方法是通过播放预先录制的声音文件。在上一章中建立的语音商务应用程序中，“Our store allow you to conveniently purchase Voice Application Development with Voice XML books and t-shirts”和“How many books do you want？”这两个TTS输出均封装在block标记中，请参见程序清单9-1。

程序清单9-1 WelcomeGoto.vxml

```
<?xml version="1.0"?>
<!--WelcomeGoto.vxml-->
<!--This document demonstrates the goto command -->
<vxml version="1.0">
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright" content="Sams Publishing 2001"/>
  <meta name="description"
    content="Welcome page for Sams online bookstore to access
➡call (xxx)-xxx-xxxx"/>
  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <form id="welcome">
    <block>
      Welcome to Sams Publishing's online bookstore.
      <goto nextitem="inventory" />
    </block>

    <block name="inventory">
      Our store allows you to conveniently purchase Voice Application
      Development with Voice XML books and t-shirts.
      <goto next="#how_many" />
    </block>
  </form>
```

```
<form id="how_many">
  <block>
    How many books do you want?
    <goto next="CalculateCostSub2.vxml" />
  </block>
</form>
</vxml>
```

block元素也可以用来封装audio元素。audio元素用于设定预先录制的声音文件，以便向用户播放。我们将在本章后面部分中的“audio元素”小节中介绍它。VoiceXML 的输出可以包含声音文件与TTS的混合信息。

同时使用TTS信息和预先录制的声音文件时好处很多。使用 TTS时，

- 开发人员能够非常容易地修改输出信息。
- 可以用脚本语言XSL、ASP和JSP等生成输出信息，因此它是动态的。
- 不需要增加录制声音文件的时间，就可以迅速对应用程序原型进行编码。
- 如果不需要为专业人员支付录制声音文件的费用，那么就可以降低应用程序的开发费用。

如果使用声音文件，那么：

- 用户对应用程序就会有一个更好的感觉，因为几乎在所有情况下，用户都希望接收声音文件的输出，而不希望接收TTS信息。
- 开发人员不必花时间给输出添加语言标记，以便使它听起来“恰好正确”

在本章的下面各节和下面各章中的示例代码中，凡是增加的文本输出都是使用 TTS给出的，因为这使得代码比较容易阅读。

9.1.1 prompt元素

prompt元素是封装TTS输出和 audio元素时使用的另一个元素。当 TTS 信息不用prompt元素封装时，便可推定prompt元素使用了默认属性。

在两种情况下，必须用 prompt元素来封装输出。首先，如果输出包含 TTS 和语音标记元素，就必须使用prompt元素。语音标记元素用于控制向用户说出 TTS 信息时的各项特性，如速度，音量和发音等。（详细说明请参见本章后面部分中的“语音标记”这一节的内容。）第二种情况是，当输出信息需要bargain、cond、count或timeout等属性时，就必须使用prompt元素：

- bargain——如果bargain属性的值是 true，用户可以使用语音应答或 DTMF 来中断输出。如果bargain的值是 false，用户将不能中断输出，输出将一直播放下去，直到结束。
- cond——cond属性使用的是布尔值，用于确定究竟是向用户输出文本还是声音文件。它的默认值是true。
- count——当初次访问菜单或窗体项时，它的初始提示计数器的值被初始化为 1。只有当count的值等于提示计数器的值，或者小于提示计数器的值但是拥有菜单或窗体项中的任何提示的最大值时，才输出提示。当一个提示被选定并且排队等候播放之后，提示计数器的值便递增1，当退出菜单或窗体项时，提示计数器便清零。
- timeout——这是提示播放后，产生 noinput事件之前，平台等待用户作出应答的时间。

timeout的默认值是各种平台根据自己的情况设定的。

在本章的“field元素”这一节中，你可以找到 bargain和 timeout属性的例子。在本章的“事件”一节中可以看到 count属性的例子。程序清单 9-2中的示例代码使用带有 cond属性的 prompt元素，建立一个新窗体，以便通知用户说他采购的商品已经添加给他的购物车。

程序清单 9-2 CalculateCostPrompt.vxml

```
<?xml version="1.0"?>
<!--CalculateCostPrompt.vxml-->
<!--Demonstrates the prompt element -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright"
    content="Sams Publishing 2001"/>

  <meta name="description"
    content="Calculate the cost of a book order,
      to access call (xxx)-xxx-xxxx"/>
  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date"
    content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <var name="order_prompt" />
  <var name="number_of_books" expr="200" />

  <form id="calculate_price">
    <var name="order_text"
      expr="'The total price of the order is '"/>
    <subdialog name="discount"
      src="./CalculateDiscount.vxml#calculate_discount">
      <param name="quantity" expr="number_of_books" />
      <param name="unit_price" expr="24.95" />
      <param name="isCoupon" expr="true" />
      <filled>
        <assign name="order_prompt"
          expr="order_text + discount.price" />
        <assign name="application.total_order_cost"
          expr="application.total_order_cost + discount.price"/>
        <assign name="application.total_books"
          expr="application.total_books + number_of_books" />
        <goto next="#output_total" />
      </filled>
    </subdialog>
  </form>

  <form id="output_total">
    <block>
      Your
      <prompt cond="number_of_books == 1">book has</prompt>
      <prompt cond="number_of_books != 1">books have</prompt>
```

```

        been added to your shopping cart.
    </block>
</form>
</vxml>

```

当output-total 窗体被访问时，首先向用户读出文字“ Your ”，接着下面的指令

```
<prompt cond="number_of_books == 1">book has</prompt>
```

被执行。由于number_of_book 的值等于200，cond属性的值计算为false，因此提示不读出。接着，下面这条指令

```
<prompt cond="number_of_books > 1">books have</prompt>
```

被执行。由于number_of_books的值大于1，因此向用户读出“ books have ”(图书已经)。最后输出的是“ been add to you shopping cart ”(被添加到你的购物车中)。

9.1.2 语音标记

语音标记是赋予 sayas、break、div、emp和 pros 等元素的名字。这些元素总是被嵌入 prompt元素中，语音标记可用于调节如何向用户播放 TTS信息的声音。它采用的是 Java API语音标记语言技术标准（0.5版，1997年8月 28日推出）。详细信息请访问网址 <http://java.sun.com/products/java-media/speech/forDevelopers/JSLM>。

有些VoiceXML 平台只支持一部分语音标记元素和属性，原因是平台使用的 TTS 服务器受到硬件和软件的限制。如果某个平台不支持某个语音标记元素，它就应该忽略该元素，并且按正常方式输出TTS信息。

程序清单9-3中的示例应用程序将语音标记元素纳入了你的语音商务应用程序。使用户能够更容易理解TTS输出。这个示例应用程序显示了语音标记元素是如何嵌套的。

程序清单9-3 WelcomeSM.vxml

```

<?xml version="1.0"?>
<!--WelcomeSM.vxml-->
<!--This document demonstrates the speech markup-->
<vxml version="1.0" application="BookStoreRoot">
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright" content="Sams Publishing 2001"/>

  <meta name="description"
    content="Welcome page for Sams online bookstore,
      to access call (xxx)-xxx-xxxx"/>

  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <form id="welcome">
    <block>
      <prompt>
        <pros vol="+20%">

```

```

<div type="paragraph">
  <emp level="strong">Welcome</emp>
  to Sams Publishing's online bookstore.

  Our store allows you to conveniently purchase Voice
  Application Development with Voice XML books and
  t-shirts. You can also visit us on the web at
  <break msec="100" />
  <sayas sub="W W W dot voice genie dot com ">
    www.voicegenie.com
  </sayas>
</div>
</pros>
</prompt>
<goto next="#how_many" />
</block>
</form>

<form id="how_many">
  <block>
    How many books do you want?
    <goto next="CalculateCostPrompt.vxml" />
  </block>
</form>
</vxml>

```

由于TTS的播放过程很长，因此用户很难收听和完全理解传递给他们的所有信息。语音标记提供了一种方法，通过插入一些停顿，调整某些单词的发音，并且改变说话的音量和速度，使得很长的TTS播放过程看上去很像是自然的对话。不过没有一种固定的方法可以用来选择语音标记元素的正确值及其属性值。开发人员常常采取试用不同的语音标记值和了解它的错误值的方法，找出“最佳”的输出。

1. sayas元素

sayas元素用于设定单词或词组应该如何发音。它配有下列属性：

- phon——phon属性使开发人员能够在应用程序这一级上控制某些单词应该如何发音。Phon使用国际音标（IPA）的Unicode字符集来定义一系列的发音。Java API语音标记技术标准称，“IPA字符由 \u0250至\u02AF的代码、\u02B0至\u02FF的修饰符、\u0300至\u036F的读音符号、某些拉丁文和希腊文、以及 \u0000至\u017F这个范围的符号字符来代表。Unicode IPA 支持特性的详细说明请参见 Unicode标准第2.0版（Unicode集团编制，Addison-Wesley Developers出版社1996年出版）。该技术标准还提供了例子，在这个例子中，单词 phonetitecs的发音使用了一系列的 unicode声音：

```

<sayas phone="\u0066\u006F\u028A\u006E\u025B\u0074\u026A\u006B\u0073">
  phonetitecs
</sayas>

```

- sub——sub属性用于定义说出的替代文字而不是实际包含的文字。
- class——class属性的有效值是 phone，date，digit，literal，currency，number 和time。

class属性用于通知平台，封装的文本应该用特定的格式来输出。

与sayas元素相同，value元素也有一个 class属性。如果要了解 class属性的举例，请参见本章的“value元素”一节的内容。

在WelcomeSM.vxml 程序中，语句

```
<sayas sub="W W W dot voice genie dot com ">www.voicegenie.com</sayas>
```

将VoiceXML平台设定为字符串“www.voicegenie.com”应该用Web地址的约定来发音。

2. break元素

break 元素用于设定语音输出的一个停顿。本元素配有下列属性：

- msec——本属性是指停顿的毫秒数。
- size——size属性用于描述停顿的持续时间。它的值可以是 none，small，medium和large。none，small，medium和 large的持续时间要根据不同的平台而定。

无论是msec和size属性，都可以用于 break元素，但是两者不能同时使用。如果这两个属性都没有设定，则假设 size=“medium”。

在WelcomeSM.vxml示例程序中，break元素<break msec=“100”>位于提示中的web地址前：

```
<prompt>
  <pros vol="+20%">
    <div type="paragraph">
      <emp level="strong">Welcome</emp>
      to Sams Publishing's online bookstore.Our store allows you to
      conveniently purchase Voice Application Development with
      Voice XML books and t-shirts. You can also visit us on the web
      at <break msec="100" />

      <sayas sub="W W W dot voice genie dot com ">www.voicegenie.com
    </div>
  </pros>
</prompt>
```

在程序段落的这个位置上应该有一个停顿，因为它将使用户有机会准备写下 web地址。虽然TTS引擎能够在句号和逗号位置上自动停顿，但是当阅读很长的信息时，有时必须设置一些停顿，使用户有额外的时间明智地处理信息。

3. div元素

div元素用于确定任何封装的 TTS将以句子或段落形式作为平台的输出。它配有下面这个属性：

- type——type属性用于确定平台将如何输出 TTS信息。Type属性的值可以是 sentence（句子），也可以是 paragraph（段落）。

在WelcomeSM.vxml示例程序中，<div type=“paragraph”>...</div> 被封装在一个提示中：

```
<prompt>
  <pros vol="+20%">
```

```

<div type="paragraph">
  <emp level="strong">Welcome</emp>
  to Sams Publishing's online bookstore.Our store allows you to
  conveniently purchase Voice Application Development with Voice
  XML books and t-shirts. You can also visit us on the web at
  <break msec="100" />
  <sayas sub="W W W dot voice genie dot com ">
    www.voicegenie.com
  </sayas>
</div>
</pros>
</prompt>

```

`<div type=" paragraph " >...</div>`这个指令告诉平台说，封装的文本是个段落，它应该正确地读出。

4. emp元素

emp元素用于设定封装的 TTS信息说出时应该使用什么样的强调等级。它配有下面这个属性：

- level——level属性用于设定单词的强调等级。它的值可以是 strong，moderate，none和 reduced。默认值是 moderate。

在WelcomeSM.vxml示例程序中，`<emp level=" strong " > Welcome </emp>`用于指明单词“welcome”应该用特定的强调语气来发音。以便强调说明该应用程序希望用户感到欢迎他使用该应用程序。

5. pros元素

pros元素用于设定说出封装的 TTS信息时的速度，音量，音调和音调范围。它配有下列属性：

- rate——rate属性用于设定说出 TTS信息时所用的速度。
- vol——vol属性用于设定说出 TTS信息时所用的音量。
- pitch——pitch属性用于设定说出 TTS信息时所用的音调。
- range——range属性用于说出 TTS 信息时所用的音调范围。

在WelcomeSM.vxml 示例程序中，`<pros vol=" +20% " > ...</pros>`封装在一个指示中：

```

<prompt>
  <pros vol="+20%">
    <div type="paragraph">
      <emp level="strong">Welcome</emp>
      to Sams Publishing's online bookstore.Our store allows you to
      conveniently purchase Voice Application Development with Voice
      XML books and t-shirts. You can also visit us on the web at
      <break msec="100" />
      <sayas sub="W W W dot voice genie dot com ">
        www.voicegenie.com
      </sayas>
    </div>
  </pros>
</prompt>

```

<pros vol = “ +20% ” > ...</pros>这个指令告诉平台说，封装的文字播放时的音量应该比前面的音量水平高20%。在你的示例程序中，前面的音量水平是平台的默认音量水平。

9.1.3 value元素

value元素用于输出变量的值和表达式的计算结果。变量既可以对应于录制的声音文件，也可以对应于某种变量的类型，比如 undefined, null, boolean, string, number或 object。当输出的值是个录制的声音文件时，该声音文件将被播放给用户。否则，变量或表达式的计算结果便通过TTS 输出给用户。

Value元素配有下列属性：

- expr——expr属性是计算的表达式，它的计算结果将被输出给用户。
- class——class属性的有效值是 phone、date、digits、literal、currency、number和time。class属性用于将封装的文本输出时采用的特定格式通知平台。
- mode——mode属性的值有两个，即默认值 tts（文语转换）和recorded。如果输出的值是一个录制的声音文件，mode 应该是recorded，否则，mode应是 tts。
- recSrc——recSrc是当 mode是 recorded时要连接的声音文件的 URL。

9.1.4 程序举例

下面让我们再次访问示例程序 CalculateCostPrompt.vxml，当用户将他要购买的图书添加到他的购物车上去时，该程序将输出一个确认消息。在下面的程序清单 9-4中所示的程序中，你使用value 元素来扩展确认消息，使之包含添加给购物车的图书数量、这些图书的费用、购物车中的图书和T恤衫的合计数量，以及购物车中的所有商品的合计费用。

程序清单9-4 CalculateCostValue.vxml

```
<?xml version="1.0"?>
<!--Root document -->
<vxml version="1.0" >
    <meta name="author" content="John O'Reilly"/>
    <meta name="copyright"
          content="Sams Publishing 2001"/>

    <meta name="description"
          content="Root document for our book store,
                  to access call (xxx)-xxx-xxxx"/>
    <meta name="keywords" content="VoiceXML Sams Books"/>
    <meta name="maintainer" content="johnnyo@sams.net"/>
    <meta name="robots" content="index,follow"/>
    <meta http-equiv="Date"
          content="Thu, 7 Aug 2000 20:20:20 EST"/>
    <var name="total_order_cost" expr="0" />
    <var name="total_book_cost" expr="0" />
    <var name="total_tshirt_cost" expr="0" />
    <var name="total_books" expr="0" />
    <var name="total_shirts" expr="0" />
```

```

</vxml>

<?xml version="1.0"?>
<!--CalculateCostValue.vxml-->
<!--Demonstrates the value element -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright"
    content="Sams Publishing 2001"/>

  <meta name="description"
    content="Calculate the cost of a book order,
      to access call (xxx)-xxx-xxxx"/>
  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date"
    content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <meta name="maintainer" content="johnoreilly@lucent.com"/>
  <meta name="application" content="Book Store"/>
  <var name="added_cost" />
  <var name="added_books" expr="200" />

  <form id="calculate_price">
    <block>
      <assign name="application.total_books"
        expr="application.total_books + added_books" />
    </block>
    <subdialog name="discount"
      src="./CalculateDiscount.vxml#calculate_discount">
      <param name="quantity" expr="application.total_books" />
      <param name="unit_price" expr="24.95" />
      <param name="isCoupon" expr="false" />
      <filled>
        <assign name="application.total_book_cost"
          expr="Number(discount.price)" />

        <!-- Keep track of the cost of the books added -->
        <var name="old_cost"
          expr="application.total_order_cost" />
        <!-- Cost of the books added = new total - old total. -->
        <assign name="application.total_order_cost"
          expr="application.total_book_cost
            + application.total_tshirt_cost" />
        <assign name="added_cost"
          expr="application.total_order_cost - old_cost" />
        <goto next="#output_total" />
      </filled>
    </subdialog>
  </form>

  <form id="output_total">
    <block>

```

```

    You have added <value expr="String(Math.round(added_books))" />

    <prompt cond="added_books == 1">book</prompt>
    <prompt cond="added_books != 1">books</prompt>

    to your shopping cart. The cost of the
    <value expr="String(Math.round(added_books))" /> added
    <prompt cond="added_books == 1">book</prompt>
    <prompt cond="added_books != 1">books</prompt>

    is <value expr="added_cost" class="currency" />. Your
    shopping cart now contains
    <value expr="String(Math.round(application.total_books))" />

    <prompt cond="application.total_books == 1">book</prompt>
    <prompt cond="application.total_books != 1">books</prompt>

    and
    <value expr="String(Math.round(application.total_shirts))" />

    <prompt cond="application.total_shirts == 1">t-shirt</prompt>
    <prompt cond="application.total_shirts != 1">t-shirts</prompt>

    The total cost for your shopping cart is
    <value expr="application.total_order_cost" class="currency" />
  </block>
</form>
</vxml>

```

在这个示例程序中，变量 `total_book_cost` 和 `total_tshirt_cost` 被添加给文档 `BooksStoreRoot.vxml`。`total_book_cost` 是购物车中目前所有图书的费用。`total_tshirt_cost` 是购物车中目前所有T恤衫的费用。

`value` 元素用于输出变量 `added_books`、`added_cost`、`total_books`、`total_costs` 和 `total_order_cost` 的内容。`value` 元素被嵌入 `block` 元素中，并且混杂在 TTS 字符串中。`value` 元素也可以封装在 `prompt` 元素之中，并且如你在“`audio` 元素”这一节的示例程序中看到的那样，`value` 元素也可以混杂在预先录制的声音文件中，你可能已经注意到 `order_text` 和 `order_prompt` 变量已经从下面这个示例程序中删除了：

```

<var name="order_prompt" />

<form id="calculate_price">
  <var name="order_text"
    expr="'The total price of the order is '" />
  <subdialog name="discount"
    src="./CalculateDiscount.vxml#calculate_discount">
    <param name="quantity" value="200" />
    <param name="unit_price" expr="24.95" />
    <param name="isCoupon" expr="true" />
    <filled>
      <assign name="order_prompt"

```

```
expr="order_text + discount.price" />
```

这个程序中的图书费用与 order_text 中包含的文本是分开输出的。通过分开输出图书的费用，你能够利用 value 元素中的 class 属性的各项特性。通过将 currency 值赋予 class 属性，你就可以将变量 total_order_cost 和 added_cost 作为货币值而不是十进制数字来输出。

在输出 added_books、total_books 和 total_shirts 时使用的 value 语句中，每个变量都有下面的类型代码所封装

```
String(Math.round(<variable>))
```

这个封装文本包含 ECMAScript 函数 Math.round () 和 string ()。执行这两个函数的结果是可以确保被它封装的变量将用一个整数值来说出。例如，在示例程序 CalculateCostValue.vxml 中，当下面的指令

```
<value expr="total_books" />
```

执行时，变量 total_books 将被说成是 “ two-hundred point zero ”。当下面的指令

```
<value expr="String(Math.round(<variable>))" />
```

执行时，变量 total_books 就被说成是 “ two-hundred ”。之所以需要使用中这些函数，目的是确保变量 added_books，total_books 和 total_shirts 能够正确输出，因为这些变量均属于 Number 类型的变量。因此 VoiceXML 平台可能将这些变量的值作为十进制数字输出。包括 String () 和 Math.round () 在内的 ECMAScript 函数将在第 11 章中进一步介绍。

为了输出购买图书的合计费用，你必须更改 filled 元素中封装的逻辑，以便计算图书的合计费用和购物车中的商品的合计费用。现在购物车中所有图书的费用将由子对话框 discount 来计算。由于子对话框返回的变量类型是 object，因此子对话框 discount 的 namelist 返回的变量 price 是变量类型 object。下面的语句

```
. <assign name="application.total_book_cost" expr="Number(discount.price)" />
```

使用 price 的变量类型 object，再将它转换成变量类型 Number，然后将变量类型 Number 赋予 total_book_cost。

当变量 total_book_cost 与 total_tshirt_cost 相加时，它必须属于变量类型 Number，这样才能得出购物中商品的新费用的正确结果。如果下面语句中的变量 total_book_cost 的类型是 Object

```
<assign name="application.total_order_cost" expr="application.total_book_cost + application.total_tshirt_cost" />
```

那么当命令执行时，变量 total_book_cost 和 total_tshirt_cost 将被并置，而不是相加，得出的结果将是 29940，而不是 2994。最后，存放在变量 old_cost 中的加上图书费用前购物车中的商品合计费用，将从存放在变量 total_order_cost 中的购物车商品的当前合计费用中减去。结果得出添加给购物车的图书的费用，它存放在变量 added_cost 中。

本示例程序中的所有 value 元素都没有设定 mode 的值，也就是说 mode 使用的是它的默认值 tts。对于本示例程序中的每个 value 元素来说，value 元素的 expr 属性的计算结果将是使用 TTS 的输出。在第 10 章的 “ record 元素 ” 这一节中，可以看到引用录制的声音文件并且将 mode 设置为 recorded 的变量的例子。

9.1.5 audio元素

录制的声音文件和 TTS是VoiceXML中将信息输出给用户的两种方法。audio元素用于设定向用户播放预先录制的声音文件。不同的 VoiceXML平台支持不同的声音文件格式。最常用的声音文件类型是.wav文件，不过你必须查看 VoiceXML平台的文档，才能了解它支持的声音文件格式的列表。

audio元素配有下列属性：

- src——src属性是向用户播放的声音文件的 URL。
- caching——caching属性可以使用fast和safe这两个值中的一个。如果 caching设置为它的默认值fast，那么如果有一个声音文件的高速缓存拷贝，并且它没有过期，那么就使用该拷贝。如果 caching设置为safe，那么不管平台以前是否将声音文件放入高速缓存，都要从 URL那里下载该声音文件。Caching的值通常设置为 fast，即预先录制的声音文件，因为当应用程序设计完成并且部署后，该文件不会经常变更。
- fetchtimeout——本属性用于设定出现提取声音文件超时之前，平台为检索声音文件而等待的时间间隔。如果出现了提取声音文件的超时，并且 audio元素封装的文本像本节的示例程序中的情况一样，那么该文本将输出给用户。如果出现了提取声音文件的超时，并且 audio元素封装了另一个 audio元素，那么平台接下来将设法下载被封装的 audio元素中引用的预录制的声音文件。例如，如果在试图下载下面的语句中的声音文件 fife.wav 时出现了提取声音文件的超时

```
<audio src="file.wav" fetchtimeout="3s">
  <audio src="fetchtimeoutfile.wav"></audio>
</audio>
```

那么该平台接下来将设法下载文件 fetchtimeoutfile.wav。如果 audio元素没有封装任何文字或 audio元素

```
<audio src="file.wav" fetchtimeout="3s" />
```

那么该平台将发出一个出错事件。Fetchtimeout的默认值是由不同的平台自己设定的。

- fetchhint——本属性用于确定平台是否应该在实际需要前预先提取声音文件。如果 fetchhint属性的值是 safe，那么只有当需要时才下载声音文件。如果 fetchhint属性的值是默认值 prefetch，那么平台将在上载网页时下载声音文件。如果 fetchhint 的值是stream，那么平台将不等到声音文件下载完毕，便开始播放该文件。

9.1.6 使用举例

程序清单9-5中的程序用预先录制的声音文件取代了不是通过 value 元素生成的所有TTS。

程序清单9-5 welcomeAudio.vxml

```
<?xml version="1.0"?>
<!--WelcomeAudio.vxml-->
<!--This document demonstrates the audio command -->
<vxml version="1.0" application="BookStoreRoot.vxml">
```

```

<meta name="author" content="John O'Reilly"/>
<meta name="copyright" content="Sams Publishing 2001"/>

<meta name="description"
      content="Welcome page for Sams online bookstore,
              to access call (xxx)-xxx-xxxx"/>

<meta name="keywords" content="VoiceXML Sams Books"/>
<meta name="maintainer" content="johnnyo@sams.net"/>
<meta name="robots" content="index,follow"/>
<meta http-equiv="Date" content="Thu, 7 Aug 2000 20:20:20 EST"/>
<form id="welcome">
  <block>
    <prompt>
      <audio src="welcome.wav" caching="fast" fetchtimeout="4s"
              fetchhint="prefetch">
        <pros vol="+20%">
          <div type="paragraph">
            <emp level="strong">Welcome</emp>
            to Sams Publishing's online bookstore.
            Our store allows you to conveniently purchase
            Voice Application Development with Voice XML
            books and t-shirts. You can also visit us on the
            web at <break msec="100" />
            <sayas sub="W W W dot voice genie dot com ">
              www.voicegenie.com
            </sayas>
          </div>
        </pros>
      </audio>
    </prompt>
    <goto next="CalculateCostAudio.vxml" fetchaudio="fetchaudio.wav" />
  </block>
</form>
</vxml>

<?xml version="1.0"?>
<!--CalculateCostAudio.vxml-->
<!--Demonstrates the mixed audio files and text-to-speech element -->
<vxml version="1.0" application="BookStoreRoot.vxml" >
  <meta name="author" content="John O'Reilly"/>
  <meta name="copyright"
        content="Sams Publishing 2001"/>

  <meta name="description"
        content="Calculate the cost of a book order,
                to access call (xxx)-xxx-xxxx"/>
  <meta name="keywords" content="VoiceXML Sams Books"/>
  <meta name="maintainer" content="johnnyo@sams.net"/>
  <meta name="robots" content="index,follow"/>
  <meta http-equiv="Date"
        content="Thu, 7 Aug 2000 20:20:20 EST"/>
  <meta name="maintainer" content="johnoreilly@lucent.com"/>

```

```

<meta name="application" content="Book Store"/>
<var name="added_cost" />
<var name="added_books" expr="200" />

<form id="calculate_price">
  <block>
    <assign name="application.total_books"
      expr="application.total_books + added_books" />
  </block>
  <subdialog name="discount"
    src="./CalculateDiscount.vxml#calculate_discount"
    fetchaudio="fetchaudio.wav">
    <param name="quantity" expr="application.total_books" />
    <param name="unit_price" expr="24.95" />
    <param name="isCoupon" expr="false" />
    <filled>
      <assign name="application.total_book_cost"
        expr="Number(discount.price)" />

      <!-- Keep track of the cost of the books added -->
      <var name="old_cost"
        expr="application.total_order_cost" />
      <!-- Cost of the books added = new total - old total. -->
      <assign name="application.total_order_cost"
        expr="application.total_book_cost +
          application.total_tshirt_cost"/>
      <assign name="added_cost"
        expr="application.total_order_cost - old_cost" />
      <goto next="#output_total" />
    </filled>
  </subdialog>
</form>

<form id="output_total">
  <block>
    <audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
    <audio src="added.wav" fetchtimeout="2s"> added </audio>
    <value expr="String(Math.round(added_books))" />

    <prompt cond="added_books == 1">
      <audio src="book.wav" fetchtimeout="2s">book</audio>
    </prompt>
    <prompt cond="added_books != 1">
      <audio src="books.wav" fetchtimeout="2s">books</audio>
    </prompt>

    <audio src="to.wav" fetchtimeout="2s">to </audio>
    <audio src="yourcart.wav" fetchtimeout="2s">
      your shopping cart.
    </audio>
    <audio src="thecost.wav" fetchtimeout="2s">
      The cost of the

```

```

</audio>
<value expr="String(Math.round(added_books))"/>
<audio src="added.wav" fetchtimeout="2s">added</audio>

<prompt cond="added_books == 1">
  <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="added_books != 1">
  <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="is.wav" fetchtimeout="2s">is </audio>
<value expr="added_cost" class="currency"/>.
<audio src="cartcontains.wav" fetchtimeout="2s">
  Your shopping cart now contains
</audio>
<value expr="String(Math.round(application.total_books))" />

<prompt cond="application.total_books == 1">
  <audio src="book.wav" fetchtimeout="2s">book</audio>
</prompt>
<prompt cond="application.total_books != 1">
  <audio src="books.wav" fetchtimeout="2s">books</audio>
</prompt>

<audio src="and.wav" fetchtimeout="2s"> and </audio>
<value expr="String(Math.round(application.total_shirts))" />

<prompt cond="application.total_shirts == 1">
  <audio src="tshirt.wav" fetchtimeout="2s">t-shirt</audio>
</prompt>
<prompt cond="application.total_shirts != 1">
  <audio src="tshirts.wav" fetchtimeout="2s">t-shirts</audio>
</prompt>

<audio src="totalcost.wav" fetchtimeout="2s">
  The total cost of the items in your shopping cart is
</audio>
<value expr="application.total_order_cost" class="currency"/>
</block>
</form>
</vxml>

```

在文档 WelcomeAudio.vxml 中，下面这个指令：

```

<audio src="welcome.wav" caching="safe" fetchtimeout="2s" fetchhint="prefetch">
  <pros vol="+20%">
    <div type="paragraph">
      <emp level="strong">Welcome</emp>
      to Sam's Publishing's online bookstore.
      Our store allows you to conveniently purchase Voice
      Application Development with Voice XML books and t-shirts.
    </div>
  </pros>
</audio>

```

```

        You can also visit us on the web at <break msec="100" />
        <sayas sub="W W W dot voice genie dot com ">
            www.voicegenie.com
        </sayas>
    </div>
</pros>
</audio>

```

用于通过 audio 元素的 src 属性来设定一个要播放的预先录制的声音文件，名字为 welcome.wav。在这个示例代码中，welcome.wave 驻留在与 welcomeAudio.vxml 相同的目录中，但是你可以非常容易地将 welcome.wav 放在 Web 服务器上，并且设定它的 URL。

当初次遇到 src 等于 welcome.wav 的 audio 元素时，便从 Web 服务器那里下载该声音文件，然后由 VoiceXML 平台播放。通过 fetchhint 属性赋予的 prefetch 值，你就可以让 VoiceXML 平台在执行 audio 元素之前下载声音文件 welcome.wav。预先提取声音文件可以使声音文件更快地下载。

声音文件下载时，voiceXML 平台也会将 welcome.wav 文件的一个拷贝放在平台上，当声音文件放入高速缓存后，再次要求播放 welcome.wav 文件并且 caching 属性是默认值 fast 时，如果 caching 属性还没有过期，平台就不必再次下载 welcome.wav 文件，而是只需要播放存储在平台上的这个文件拷贝就行了。

welcome.wav 取代的文本被封装在 audio 元素中，如果不能在 fetchtimeout 属性规定的 2 秒钟内下载声音文件，或者你在不支持 .wav 文件格式的平台上运行该文档，或者你因为任何其他原因而下载或播发声音文件，那么将播放 audio 元素中封装的文本。VoiceXML 允许在 audio 元素中封装语言标记、value 元素、声音文件元素和混合声音文件，以及 TTS。

在示例程序 CalculateCostAudio.vxml 中，audio 元素将以下面的形式相继出现：

```

<audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
<audio src="added.wav" fetchtimeout="2s"> added </audio>

```

当 audio 元素相继出现时，它们将在一个声音文件之后立即播放另一个声音文件，如果录制正确的话，人们就会感到这两个声音文件录制成了一个声音文件一样。

声音文件也可以与 TTS 信息混合在一起。当下面的命令：

```

<audio src="youhave.wav" fetchtimeout="2s"> You have </audio>
<audio src="added.wav" fetchtimeout="2s"> added </audio>
<value expr="String(Math.round(added_books))" />

```

执行时，首先向用户播放 “ You have added ” (你已经增加)，然后通过 TTS 向用户说出 “ two hundred ” (200)。

audio 元素既可以是 block 元素的子元素，也可以的 prompt 元素的子元素。在 CalculateCostAudio.vxml 程序文件中，下面的 audio 元素：

```

<audio src="youhave.wav" fetchtimeout="2s"> You have </audio>

```

是 block 元素的子元素。在同一个程序文件中，下面的 audio 元素：

```

<audio src="book.wav" fetchtimeout="2s">book</audio>

```

是 prompt 元素的子元素。

程序文件 WelcomeAudio.vxml 和 CalculateCostAudio.vxml 都提供了使用 fetchaudio 属性的例子。在 WelcomeAudio.vxml 中，下面的指令：

```
<goto next="CalculateCostAudio.vxml" fetchaudio="fetchaudio.wav" />
```

用于设定当下载 CalculateCostAudio.vxml 文件时，VoiceXML 平台将播放 fetchaudio.wav 文件。而在 CalculateCostAudio.vxml 文件中，下面的指令：

```
<subdialog name="discount" src="/CalculateDiscount.vxml#calculate_discount"  
fetchaudio="fetchaudio.wav">
```

用于设定下载 CalculateDiscount.vxml 文件时，VoiceXML 平台将播放 fetchaudio.wav 文件，如果你一面等待文件下载，一面又播放声音文件，就可以减少用户必须经过的“停播时间”量。

9.2 接收用户的输入

到现在为止，本章介绍的内容均与向用户输出数据有关。下面各节内容将集中介绍如何接收用户输入的信息。输入信息可以用两种方式来接收，一种是语音，一种是双音多频（DTMF）。输入信息既可以直接识别，也可以录音。直接识别的输入信息是通过语法来表述的，该语法设定了一组声音，用户可以发出这一组声音，也可以由用户按下 DTMF 按钮，以便要求进行某项操作，或者将信息提供给应用程序。

9.2.1 field 元素

field 元素用于接收来自用户的自动语音识别信息或 DTMF 信息。它配有下列属性：

- name——name 是用于存放来自用户的输入信息的变量。
- expr——expr 是 name 属性中声明的变量的初始值。expr 并不是个必不可少的属性，如果它被省略了，那么该变量的值将是 undefined。如果变量的值是 undefined 之外的某个值，那么封装在 field 元素中的指令将不予执行。
- cond——cond 是个表达式，它计算的结果可以是 true，也可以是 false。如果 cond 的值计算为 true，那么封装在 field 元素中的指令将被执行，否则便跳过 field 封装的指令。
- type——type 用于指明期望的用户输入信息的类型。内置的类型是 boolean、date、digits、currency、number、phone 和 time。type 属性是可有可无的。如果没有设定 type，field 变量便与 field 内声明的用户定义的语法相关联。
- boolean——当用户说“yes”或者按下 DTMF 键 1 时，便返回 true 值。当用户说“no”或者按下 DTMF 键 2 时，便返回 false 值。

Yes 和 no 的默认 DTMF 按键可以被取代，下面的代码将取代 yes 的默认 DTMF 值：

```
<field type="boolean?y=7" />
```

这个代码中 yes 的 DTMF 值被设置为 7，同样，no 的 DTMF 值可以设置为 8。请看下面的代码：

```
<field type="boolean?n=8" />
```

另外，yes 和 no 的 DTMF 值都可以进行更换，请看下面的代码：

```
<field type="boolean?y=7;n=8" />
```