< field >

Description

The < field > element declares an input field in a form and prompts the user for values that match a grammar.

Syntax

```
<field
name="String"
expr="ECMAScript_Expression"
cond="ECMAScript_Expression"
type="boolean"|"number"|"digit"
slot="String"
modal="true" | "false"
saveutterance="true" | "false">
child elements
</field>
```

Attributes

name	The name attribute defines the name of the field item variable that holds the matched user input. This attribute is required.
expr	The expr attribute is the initial value of the field item variable. This field will be visited only if the expression evaluates to undefined. This attribute is optional and defaults to undefined.
cond	The cond attribute is a Boolean condition that must evaluate to true in order for this field to be visited. This attribute is optional and defaults to true.
type	The type attribute specifies a built-in grammar. This attribute is optional and can be used as an alternative to the <grammar> element.
slot	The slot attribute is the name of the grammar slot used to populate the field item variable for mixed initiative dialog. This attribute is optional and defaults to the variable name.
modal	The modal attribute is set to true if only the field's grammars are enabled. Otherwise, all active grammars are enabled. This attribute is optional and defaults to false.

Shadow Variables

The <field> shadow variable (name$) has the following properties after the field is filled in by user input. These properties are available in object application.lastresult$ as well: * name$.utterance – a raw string of words that were recognized. * name$.inputmode – a user input type, either dtmf or voice. * name$.interpretation – an interpretation for this result. * name$.confidence – a recognition confidence level. Floating point value between 0 to 1.0.

DTMF Built-in types

The following built-in types are supported for the DTMF input mode: boolean The grammar for affirmative and negative phrases. It returns “true” for yes and “false” for no. digits The grammar for a string of digits from spoken or DTMF input. It returns the string of digits. number The grammar for numbers. It returns a string of digits from 0 to 9, and may optionally include a decimal point (“.”) and/or a plus or minus sign.

DTMF Built-in type parameters

The following built-in type parameters can be parameterized with this syntax:

```
typename?parameter1=value1;parameter2=value2
```

For example, we can assign a DTMF sequence for a Boolean type:

```
<field name="mychoice" type="boolean?y=5;n=6">
```

boolean	y - the DTMF sequence for an affirmative answer. n - the DTMF sequence for a negative answer.
digits	length - exact number of digits.
number	length - exact number of numbers.

If there is a conflict among these parameters, an error.badfetch event is thrown.

ASR Built-in types

The built-in types supported depends on the ASR installed and configured. Refer to the ASR provider.

Parents

```
<form>
```

Children

```
<audio>, <catch>, <enumerate>, <filled>, <grammar>, <help>, <link>,
<noinput>, <nomatch>, <option>, <prompt>, <property>, <return>, <value>
```

Extensions

None.

Limitations/Restrictions

None.

Example Code

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
 <form id="testfield">
  <block>
   Welcome to the restaurant.
  </block>
<!-- field using built in grammar with digits type -->
  <field name="built_in_gram1" type="number">
   <prompt> How many people you have? </prompt>
   <filled>
    Here is a table for <value expr="built_in_gram1"/> people.
    Have a seat.
    <goto nextitem="built_in_gram2" />
</filled>
  </field>
<!-- filed using built in grammar with boolean type -->
  <field name="built_in_gram2" type="boolean">
   <prompt> Are you ready for ordering now ? </prompt>
   <filled>
    <if cond="built_in_gram2 == 'true'">
     <goto nextitem="optionlist" />
    <elseif cond="built_in_gram2 == 'false'"/>
     <goto nextitem="notready" />
    </if>
   </filled>
  </field>
<!-- filed using option list -->
  <field name="optionlist">
   <prompt>
    Which entree would you like, baked potato, french fries?
   </prompt>
   <option dtmf="1" value="baked potato">baked potato</option>
   <option dtmf="2" value="french fries">french fries</option>
   <filled>
    ok. <value expr="optionlist"/> as entree.
    <goto nextitem="explicit_gram1" />
```

```
   </filled>
  </field>
<!-- filled using explicit grammar -->
  <field name="explicit_gram1">
   <prompt>
    Here is the main dishes you could choose from,
    Vegetarain delight, Prime Rib, Baked clams
   </prompt>
   <grammar type="text/x-grammar-choice-dtmf" mode="dtmf">
    1 {Vegetarain delight} |
    2 {Prime Rib} |
    3 {Baked clams}
   </grammar>
   <filled>
    Good choice.
    <value expr="explicit_gram1"/>
    is my favorite as well.
    <goto nextitem="ordertaken" />
   </filled>
  </field>
  <block name="goodbye">
   good bye
   <disconnect/>
  </block>
  <block name="ordertaken">
   Please wait. Your meal will be ready shortly.
   <disconnect/>
  </block>
  <block name="notready">
   Ok. Take your time and call back later.
   <disconnect/>
  </block>
 </form>
</vxml>
```