The “decimal point” voice input allows to perform a numeric input using a decimal point as separator in a single utterance. That allows to say “eleven dot three” for an input as “11.3”.

The integer part of the input (previous the dot) decimal part of the input can be uttered as a “natural” number from 0 to 999. I/e: “one” for 1.xxx, “seventeen” for 17.xxx, and “two hundred and twenty seven” for 227.xxx. The decimal part is mandatory and should be announced even if it is zero. I/e for 0.5 it is illegal to say “dot five” and should be announced “zero dot five”.

The decimal part is optional and can have several ways to be announced. The decimal point input should be defined to work with one, two or three decimal places. So it is needed to define if the “decimal places” announces will be:

- “x.9”
- “x.99”
- “x.999”

all along the current application. There are three predefined <point> grammar rules in the “.bnf” file to recognize the corresponding input format. This file should be modified to allow one of the tree different inputs.

The decimal part is optional, so if it is zero, only the integer part should be announced.

One decimal place

Number	**Utterance**	**Result**
12.5	“twelve dot five”	12.5
12.0	“twelve dot zero”	(illegal)
12.0	“twelve”	12.0

Two decimal places

Number	**Utterance**	**Result**
12.50	“twelve dot fifty”	12.50
12.50	“twelve dot five”	(illegal)
12.00	“twelve dot zero zero”	(illegal)
12.00	“twelve”	12.00
12.01	“twelve dot zero one”	12.01

Tree decimal places

Number	**Utterance**	**Result**
12.500	“twelve dot five hundreds”	12.500
12.500	“twelve dot five”	(illegal)
12.000	“twelve dot zero zero zero”	(illegal)
12.000	“twelve”	12.000
12.015	“twelve dot zero fifteen”	12.015
12.007	“twelve dot zero zero seven”	12.007

12.103	“twelve dot one hundred and tree”	12.103

It is possible to configure checks of the several parts of the decimal number, the number of the output digits.
The “expected” option holds the check options as “Min1-Max1,Min2-Max2,n,=,” where:

- Min1, the minimum value of the integer part.
- Max1, the maximum value of the integer part.
- Min2, the minimum value of the decimal part.
- Max2, the maximum value of the decimal part.
- n, the number of output digits of the decimal part (Independently of the input mode).
- =x, the separator character as output (I/e “=,” will replace the decimal dot by a comma.

I/e two decimal places in the English way: “Expected=000-499,00-99,5,=.” Will check an integer part from 0 to 499, a decimal part from 00 to 99, then output is formatted with five decimal positions. 71.09 said as “seventy-one dot zero nine” will generate an output as “71.09000”.

I/e two decimal places in French way: “Expected=0-99,0-99,6,=,” Will check an integer part from 0 to 99, a decimal part from 00 to 99, then output is formatted with six decimal positions. 71.09 said as “soixante onze virgule zéro neuf” will generate an output as “71,090000”.

Grammars

A new grammar <point> denoted by the “P” letter will be added to the voice library to activate this option. Three predefined <point> grammars will be available on the ‘.bnf’ file and will be commented / uncommented to run only one of them.

```
// 999.999 decimal point input
<point>: <hundreds> | <hundreds> . 00 <digits> | <hundreds> . 0 <digex> | <hundreds>
. <century> <tens> ;
// 999.99 decimal point input
//<point>: <hundreds> | <hundreds> . <digex> ;
// 999.9 decimal point input
//<point>: <hundreds> | <hundreds> . <digits> ;
```

Decimal point separator
A special pronounce was added for languages using the comma separator I/e:

```
!pronounce . PRONAS "komma";
```

The MANDATORY decimal point separator in the grammar is the dot “.”, to change the output result you can use the “Expected” field.

(Wikipedia) History of the decimal separator