



## Time

Plugin Name:	time
Display Name:	Time/Time Period
Class Name:	com.audium.sayitsmart.plugin-ins.AudiumSayItSmartTime

- [Description, on page 1](#)
- [Input Formats, on page 1](#)
- [Output Formats, on page 2](#)
- [Filesets, on page 3](#)
- [Audio Files, on page 4](#)
- [Examples, on page 5](#)

## Description

This Say It Smart type handles the playback of the time or a time period. Whether to play back the time or a time period is specified by an input format. The plug-in also supports the different components of the time separated by colons (:) and will require the use of this delimiter if any component of the time is expressed with one digit instead of two (for example, 1:09 AM can be expressed as 0109 or 1:9 where the colon is required if any component is not padded with 0s). The time arrives in 24-hour military format and time periods arrive in combinations of hours, minutes, and seconds. The time is read back in standard English fashion; the hour, the minute, and either “A.M.” or “P.M.”. Time periods are read back with each component followed by a qualifier (*hours*, *minutes*, or *seconds*). The plug-in will only read the time or time period if it is legitimate (the components are within the appropriate range).

This plug-in uses the Unified CVP Number Say it Smart plug-in to render each component of the time or time period. It uses the same audio files so recordings done to support Number can be leveraged to support Time.

## Input Formats

Name (Display Name)	Description
------------------------	-------------

time_hhmm (24Hr Time (HHMM))	This input format is used to specify the time. It must arrive in 24-hour format with the hours from 00 to 23 and the minute from 00 to 59.  The data can be handled in any of the following formats: <ul style="list-style-type: none"> <li>• hhmm</li> <li>• hh:mm</li> <li>• h:mm</li> <li>• hh:m</li> <li>• h:m</li> </ul>
period_hhmmss (Time Period (HHMMSS))	This input format is used to specify a time period including hours (from 00 to 99), minutes (from 00 to 59), and seconds (from 00 to 59).  The data can be handled in any of the following formats: <ul style="list-style-type: none"> <li>• hhmmss</li> <li>• hh:mm:ss</li> </ul>
period_hhmm (Time Period (HHMM))	This input format is used to specify a time period including hours (from 00 to 99) and minutes (from 00 to 59).  The data can be handled in any of the following formats: <ul style="list-style-type: none"> <li>• hhmm</li> <li>• hh:mm</li> </ul>
period_mmss (Time Period (MMSS))	This input format is used to specify a time period including minutes (from 00 to 99) and seconds (from 00 to 59).  The data can be handled in any of the following formats: <ul style="list-style-type: none"> <li>• mmss</li> <li>• mm:ss</li> </ul>

## Output Formats

Name (Display Name)	Input Format Depends On	Description
time (The Time)	time_hhmm	The time is read back with the hour (from 1 to 12) followed by the minute (from 0 to 59) followed by "A.M." or "P.M.". If the minute is zero, it will be omitted.

time_special_12 (The Time 12=Midnight/Noon)	time_hhmm	The time is read back exactly as above except that 00:00 is read as <i>midnight</i> and 12:00 is read as <i>noon</i> .
period (Time Period)	period_hhmmss period_hhmm period_mmss	The time period is read back with each component followed by the qualifier <i>hours</i> , <i>minutes</i> , or <i>seconds</i> . If one component is zero, it is omitted.

## Filesets

Name (Display Name)	Output Format Depends On	Description
standard_time (Standard Time)	time	This fileset involves fewer audio files to render the time but at the cost of sounding a bit robotic. This directly correlates to the Unified CVP Number Say it Smart plug-in's <i>standard</i> fileset.
enhanced_time (Enhanced Time)	time	This fileset involves more audio files to render a better sounding time. This directly correlates to the Unified CVP Number Say It Smart plug-in's <i>enhanced</i> fileset.
standard_special_12 (Standard Time + Noon/Midnight)	time_special_12	This fileset is exactly the same as <i>standard_time</i> except with two extra files; <i>noon</i> and <i>midnight</i> .
enhanced_special_12 (Enhanced Time + Noon/Midnight)	time_special_12	This fileset is exactly the same as <i>enhanced_time</i> except with two extra files; <i>noon</i> and <i>midnight</i> .
standard_period (Standard Time Period)	period	This fileset involves fewer audio files to render the time period but at the cost of sounding a bit robotic. This directly correlates to the Unified CVP Number Say it Smart plug-in's <i>standard</i> fileset.
enhanced_period (Enhanced Time Period)	period	This fileset involves more audio files to render a better sounding time period. This directly correlates to the Unified CVP Number Say It Smart plug-in's <i>enhanced</i> fileset.

# Audio Files



**Note** When reading back a time, zeros are replaced by *oh*. for example, 13:05 is read back as *one oh five P.M.* This is not the case for time periods.

## standard\_time

oh	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	30	40	50	am	pm				

## enhanced\_time

oh	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
am	pm								

## standard\_special\_12

oh	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	30	40	50	am	pm	noon	midnight		

## enhanced\_special\_12

oh	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59

am	pm	noon	midnight						
----	----	------	----------	--	--	--	--	--	--

**standard\_period**

oh	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	30	40	50	60	70	80	90		
hour	hours	minute	minutes	second	seconds				

**enhanced\_period**

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99
hour	hours	minute	minutes	second	second				

## Examples

## Example #1

Data:	20:43
Input Format:	time_hhmm
Output Format:	time
Fileset	standard_time
Playback:	“8” “40” “3” “pm”

## Example #2

Data:	20:43
-------	-------

Input Format:	time_hhmm
Output Format:	time
Fileset	enhanced_time
Playback:	“8” “43” “pm”

## Example #3

Data:	0000
Input Format:	time_hhmm
Output Format:	time_special_12
Fileset	standard_special_12
Playback:	“midnight”

## Example #4

Data:	02:00
Input Format:	time_hhmm
Output Format:	time_special_12
Fileset	enhanced_special_12
Playback:	“2” “am”

## Example #5

Data:	12:09
Input Format:	time_hhmm
Output Format:	time
Fileset	standard_time
Playback:	“12” “oh” “9” “pm”

## Example #6

Data:	810001
Input Format:	period_hhmmss
Output Format:	period
Fileset	standard_period
Playback:	“80” “1” “hours” “1” “second”

## Example #7

Data:	0001
Input Format:	period_hhmm
Output Format:	period
Fileset	standard_period
Playback:	"1" "minute"

## Example #8

Data:	99:59
Input Format:	period_mmss
Output Format:	period
Fileset	enhanced_period
Playback:	"99" "minutes" "59" "seconds"

