

The WireLess TelNet voiXtreme voice system

Table of Contents

Description.....	3
TTS System.....	3
ASR System.....	3
Predefined actions.....	4
User and noise calibration.....	4
Voice Triggers.....	4
Conditions.....	5
TTS.....	5
ASR.....	5
Input Modes.....	7
Single Voice input.....	7
Single Voice input algorithm:.....	8
Single + Repeat Voice input.....	9
Single + repeat Voice input algorithm:.....	10
Validate Voice Input (input digits).....	11
Validate Voice input algorithm (digits):.....	12
Validate Voice Input (input cancel).....	13
Validate Voice input algorithm (cancel):.....	14
TTS Only modes.....	15
Read screen.....	15
Print Escape commands.....	15
Configuration File.....	16
Global definitions about the voice engine.....	16
TTS (Text-to-speech) section.....	16
Text delimiters on display	16
User messages.....	17
Keys to control voice engine by user.....	17
Special character translator options.....	18
ASR (Automatic Speech Recognition) definitions section.....	18
Tuning.....	18
Custom grammars.....	19
Keywords.....	19
Keyboard keys by keywords.....	21
ASR engine calibration.....	21
Voice Triggers.....	23
Voice trigger conditions section.....	23
Cursor position condition.....	24
Voice trigger TTS actions section.....	24

Voice trigger ASR actions section.....	25
Sample configuration file.....	29
Screen and Trigger samples.....	29
Bin location screen.....	29
Quantity at location screen.....	30

Description

The TelNet voice system allows the PDA to say texts (TTS) and to perform ASR inputs using a “standard” telnet application (VT / 5250).

TTS System

Text data to be said (by TTS), may be supplied by:

- TnVT print commands (***ESC 5i / ESC 4i***)
- Text displayed on screen surrounded by beginning and end tags
- “Voice Triggers” containing constant and variable texts

The TTS only system is intended to be used as “voice guide”, the user only hears the system and do not perform voice recognition actions.

It is not recommended to not mix the different voice modes.

ASR System

Voice input from user. The ASR system recognizes the user's voice and converts it into text data using “grammars”. Data input is processed by the local application, and then sent to host following programming configuration.

Grammars are recognition algorithms designed to produce text inputs.

Available grammars are:

- “Digits”: accepts decimal digits from “0” to “9”. More than one digit numbers are composed by repetition of digits. Digits should be preceded by a prefix and followed by a suffix to improve recognition.
- "Controls": accepts “Accept” and “Cancel” orders. *Accept* and *Cancel* keywords are configurable and processed by the voice library, and host transmission is performed following input algorithm.
- “Functions”: accepts up to 20 keywords, that should be associated with a function key in the configuration file.
- “Custom”: 10 custom grammars allowing to independent activation of groups of keywords.

Keywords change according to the language selected for the ASR

Each time a word is recognized by the ASR engine, a “Reliability Level” is returned indicating the score reached by the voice locution to match one of the words in grammar. A high score (near 10.000) indicates that ASR engine consider the user locution corresponds to the returned word with high level of probability. A low score indicates a low level of probability, so there is a high risk of misheard, so in this case it is

recommended to reject the recognition and wait for user input again. The returned reliability level changes following application environment, so it is possible to configure this option in configuration file.

ASR engine processes user locutions in three steps:

- Waiting for audio signal
- Capturing
- Recognition

In the first state the system waits for some audio signal. If this signal is significant, it consider that user is speaking then starts the capture streaming. When audio signal is not more detected, or a timeout is reached, the audio captured frames are processed by recognizer. A configurable option (`AsrThreshold`) allows to tune the audio level to switch from one state to another.

When activated, the voice system will be in play mode (recognizing defined input words) or in pause mode (recognizing only resume words).

Predefined actions

Some keywords are processed internally by the voice library to perform some usual functions:

- Repeat command, to repeat to user the last voice announce.
- Information command, to say to user useful information about the current trigger.
- Volume up/down command, to set up / down the TTS volume.
- Speed up/down command, to set up/down the TTS speed
- Speech-pause command, allowing to pause the ASR recognition engine to avoid invalid input.

These actions are performed internally, it is just needed to fill some configurable fields to adapt to current application.

User and noise calibration

The ASR engine automatically adapts the recognition to the current user and noise environment. It is recommended to reset the terminal to change the current user adaptation. An ASR calibrate procedure is run at emulator start-up allowing the user to say some words to calibrate the ASR engine to current user and current noise environment. A configuration file allows to set critical words to perform calibration and validation of good recognition before working with the system.

Voice Triggers

Voice triggers are defined in three sections:

- *Conditions*: match needed to perform an event
- *TTS*: allows to give the user a voice instruction
- *ASR*: allows to get data from user

One or more triggers could be defined to perform voice events. Voice events can be TTS only, ASR only and TTS + ASR.

Conditions

“Voice triggers” will start the voice input process. Their execution is determined by the presence of specific conditions on the emulated screen, such as:

- Text found / not found at a specific place of the screen
- Cursor position
- It is possible to match two screen areas

Up to five conditions about screen display may be defined.

TTS

In the Voice Trigger, a TTS “voice announcement” may be defined to be said before the ASR recognition process starts. This announcement may be a mix of constant text (defined inside the configuration file) and variable text (read from the emulated screen). Variable text will be defined by beginning position and length: three numeric values separated by commas (row and column position in screen and text length) delimited by curly brackets: {r, c, l}. For example, {10, 7, 14} defines a text located at row 10, column 7, with 14 characters length.

Some processing may be done on text to say.

- Say or spell (<SPL> <SAY>)
- Uppercase or lowercase (U, L)
- Alpha or numeric trim(T, N)
- Suppress leftmost or rightmost characters (L, R)
- Take only leftmost or rightmost characters (l, r).

Up to three independent sentences are available. It is possible to conditionally say or not these sentences following the precedent trigger executed.

ASR

For the ASR recognition you should define the ASR grammar to be used and the ASR algorithm.

Standard voice grammars, "Digits", "Controls" and “Functions”, are included with the voiXtreme libraries. Custom grammars will be added. The grammars are activated independently in each trigger.

The ASR algorithm could be:

- *Single* (Confirm=No): The engine recognizes the user input, then sends it to host.
- *Single with repetition* (Repeat=Yes): The engine recognizes the user input, repeats it to user by TTS, then sends it to host.
- *Validate* (Confirm=Yes): The engine recognizes the user input, repeats it and waits for a command said by user (“*accept*” / “*cancel*”):
 - If *cancel*, the application restarts the trigger from the beginning (voice announcement).
 - If *accept*, the application sends user input to host.
- *Validate with repetition* (Repeat=Yes): Same as *Validate* but the validation command is echoed to user.

See below the different input modes.

Some options allows to validate user input such as min and max digits.

A format option allows to reformat output strings to combine user input (represented by {\$}) and constants or variable (read from screen) data.

A check option allows to mandatory match user input with constant or variable data to allow data output. It will be useful to validate input in transactions when the user response must match some known value (i/e location, product code, etc).

Informational options allow to help the user in working flow to know what is expected to say.

Input Modes

The following images describe the different input modes.

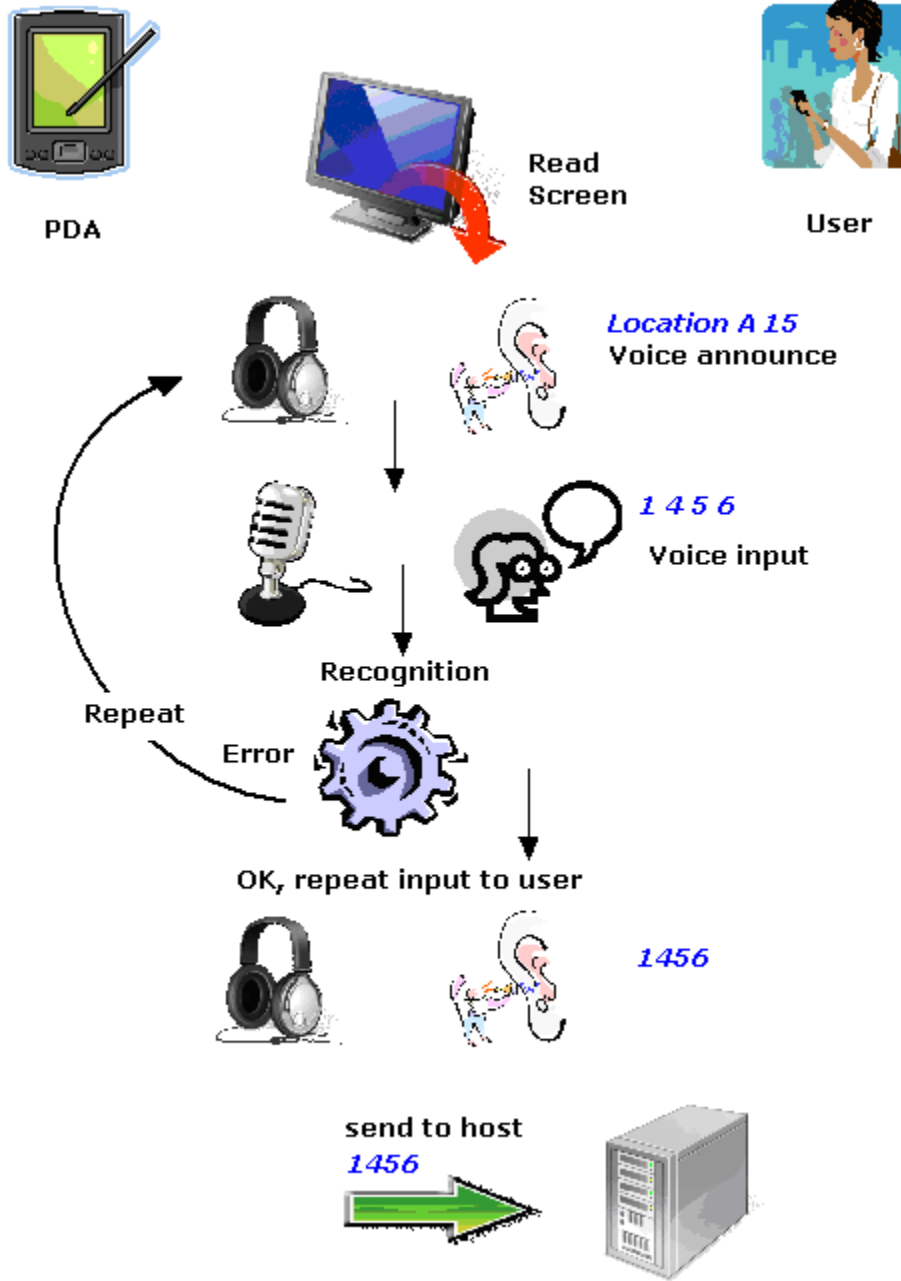
Single Voice input



Single Voice input algorithm:

- 1/ A defined trigger matches screen data and cursor position.
- 2/ The TTS section of the trigger is processed to say the voice announcement to the user.
- 3/ Waits the end of TTS announcement.
- 4/ The ASR section of trigger is processed to perform the ASR input, using the grammar indicated in the section (on this case, "Digits").
- 5/ End of recognition:
 - Repeat, goes to (2).
 - Error (no recognition, low reliability level, min / max) goes to (2).
 - Cancel, sends the [\[VOICETRIGGER_ASR_XX\]](#)/Cancel string to Host, then exit.
 - Digits, sends the input followed by [\[VOICETRIGGER_ASR_XX\]](#)/Terminator string to Host, then exit.

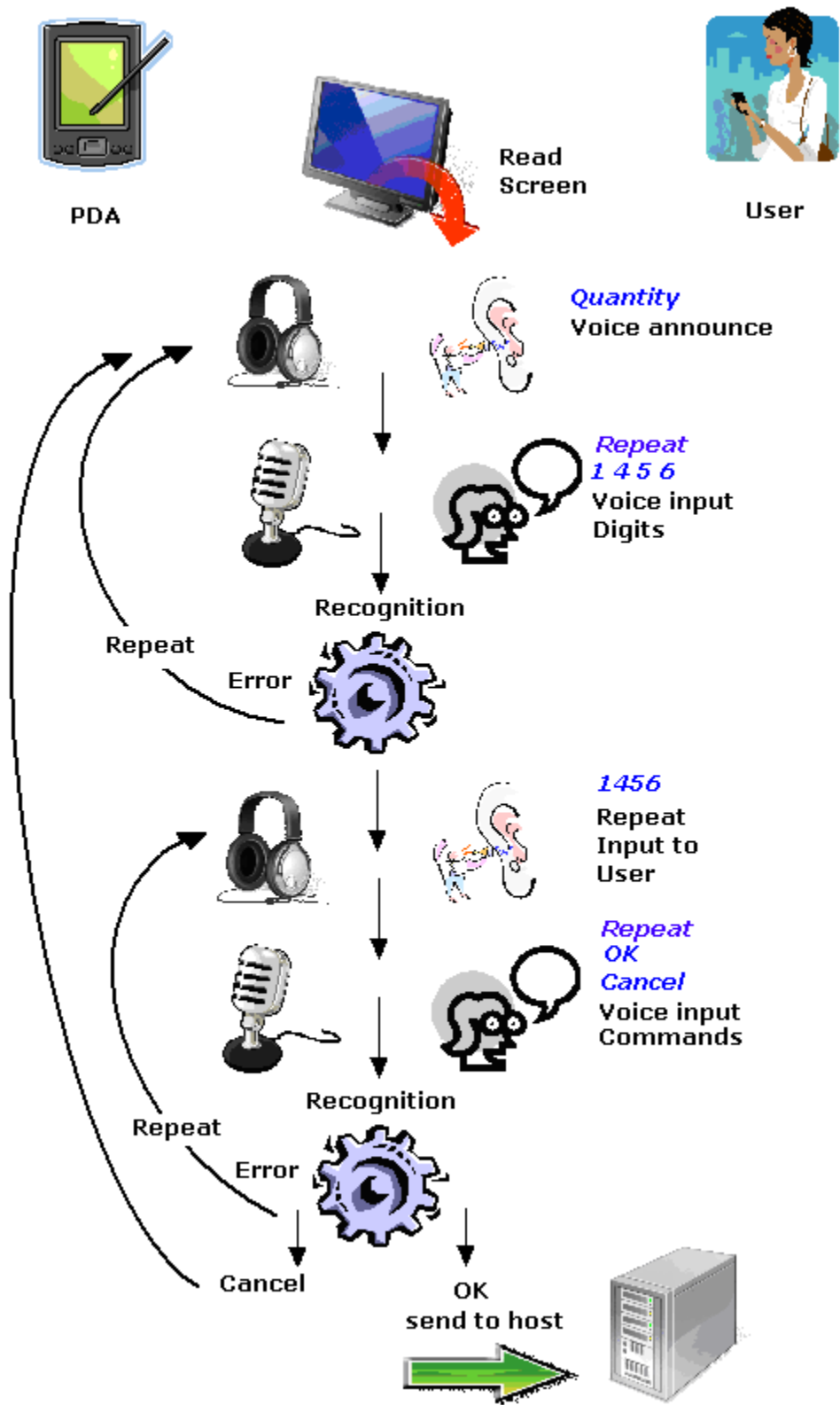
Single + Repeat Voice input



Single + repeat Voice input algorithm:

- 1/ A defined trigger matches screen data and cursor position.
- 2/ The TTS section of the trigger is processed to say the voice announcement to user.
- 3/ Waits the end of TTS announcement.
- 4/ The ASR section of the trigger is processed to perform the ASR input, using the grammar indicated in the section (on this case, "Digits").
- 5/ End of recognition:
 - Repeat, goes to (2).
 - Error (no recognition, low reliability level, min / max) goes to (2).
 - Cancel, says "Cancel" to user, sends the [\[VOICETRIGGER_ASR_XX\]](#)/Cancel string to Host, then exit.
 - Digits, says the input to user, sends the input followed by [\[VOICETRIGGER_ASR_XX\]](#)/Terminator string to Host, then exit.

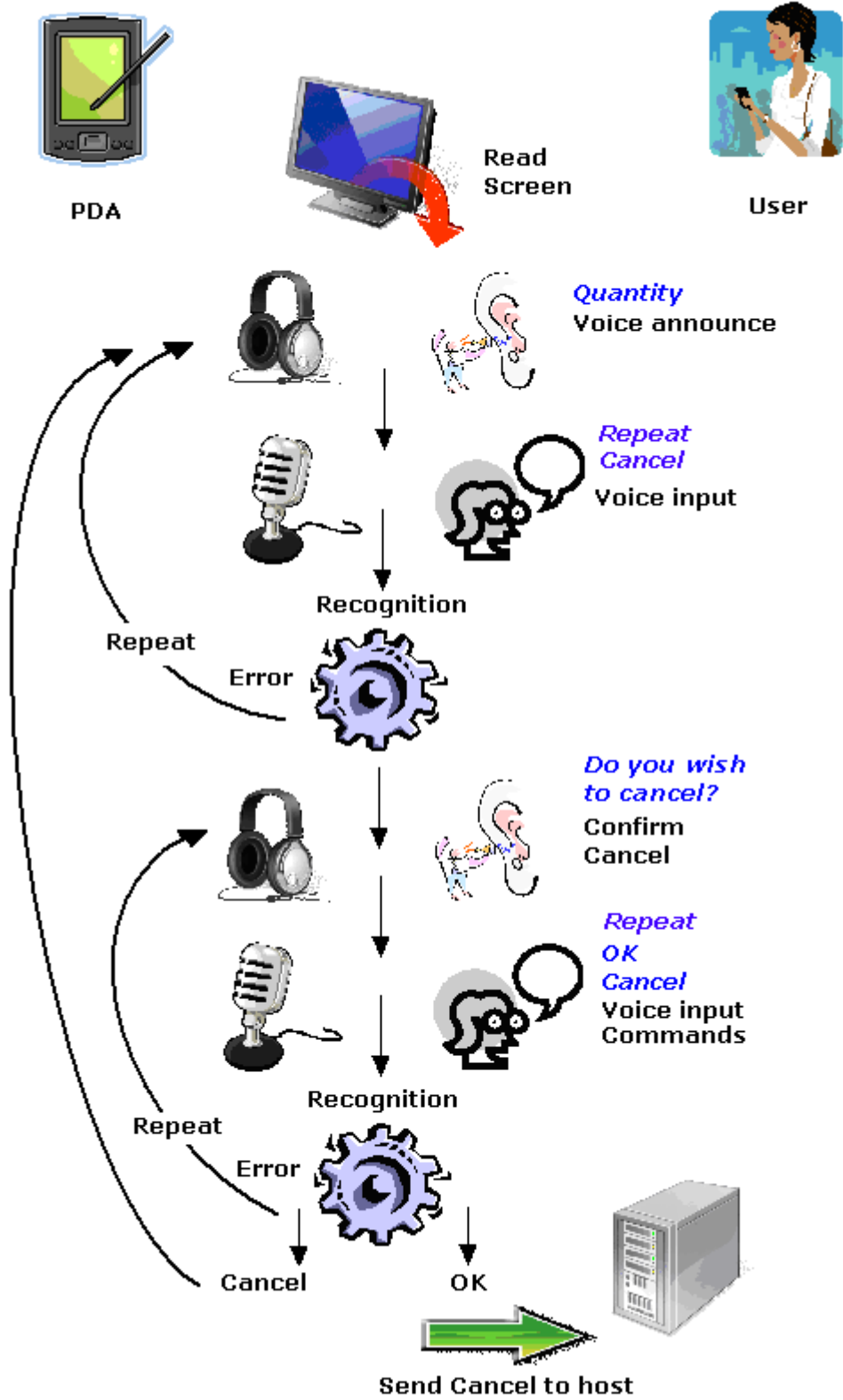
Validate Voice Input (input digits)



Validate Voice input algorithm (digits):

- 1/ A defined trigger matches screen data and cursor position.
- 2/ The TTS section of the trigger is processed to say the voice announcement to user.
- 3/ Waits the end of TTS announcement.
- 4/ The ASR section of the trigger is processed to perform the ASR input, using the grammar indicated in the section (on this case, "Digits").
- 5/ End of recognition:
 - Repeat, goes to (2).
 - Error (no recognition, low reliability level, min / max) goes to (2).
 - **Cancel, goes to (9).**
 - Digits, goes to (6).
- 6/ Says the input to user by TTS.
- 7/ Performs the ASR recognition, using the "Commands" grammar.
- 8/ End of recognition:
 - Repeat, goes to (6).
 - Error (no recognition, low reliability level, min / max) goes to (6).
 - Cancel, goes to (6).
 - OK, says "OK" to user, sends the input followed by [\[VOICETRIGGER_ASR_XX\]](#)/Terminator string to Host, then exit.
- 9/ **Says the CancelConfirmation phrase to user by TTS.**
- 10/ **Perform the ASR recognition, using the "Commands" grammar.**
- 11/ End of recognition:
 - Repeat, goes to (9).
 - Error (no recognition, low reliability level, min / max) goes to (9).
 - **Cancel, says "Cancel" to user, goes to (2).**
 - **OK, says "OK" to user, sends the [\[VOICETRIGGER_ASR_XX\]](#)/Cancel string to Host, then exits.**

Validate Voice Input (input cancel)



Validate Voice input algorithm (cancel):

- 1/ A defined trigger matches screen data and cursor position.
- 2/ The TTS section of the trigger is processed to say the voice announcement to user.
- 3/ Waits the end of TTS announcement.
- 4/ The ASR section of the trigger is processed to perform the ASR input, using the grammar indicated in the section (on this case, "Digits2).
- 5/ End of recognition:
 - Repeat, goes to (2).
 - Error (no recognition, low reliability level, min / max) goes to (2).
 - Cancel, goes to (9).
 - Digits, goes to (6).
- 6/ Says the input to user by TTS.
- 7/ Performs the ASR recognition, using the "Commands" grammar.
- 8/ End of recognition:
 - Repeat, goes to (6).
 - Error (no recognition, low reliability level, min / max) goes to (6).
 - Cancel, goes to (6).
 - OK, says "OK" to user, sends the input followed by [\[VOICETRIGGER_ASR_XX\]](#)/Terminator string to Host, then exit.
- 9/ Says the CancelConfirmation phrase to user by TTS.
- 10/ Performs the ASR recognition, using the "Commands" grammar..
- 11/ End of recognition:
 - Repeat, goes to (9).
 - Error (no recognition, low reliability level, min / max) goes to (9).
 - Cancel, says "Cancel" to user, goes to (2).
 - OK, says "OK" to user, sends the [\[VOICETRIGGER_ASR_XX\]](#)/Cancel string to Host, then exits.

TTS Only modes

These modes are intended to be used in a “TTS” only applications. Users will heard the terminal, but input data by scanner or keyboard (no ASR recognition).

Do not use these modes in the same application with voice triggers.

Read screen

Read Screen sample

In this sample, a part of the screen (delimited by SAY: / :SAY) is read to user by TTS.

```
** Order Picking OPL-LOC
Order Number : 1259-AA0
Type: Mandatory
Bin Location : A-L-008
Bin-Loc ID: █
SAY: Go to ALPHA LIMA 8 :SAY
```

The text delimited by the tags “SAY: - :SAY” is read out loud to user as: “*go to alpha lima eight*”.

Print Escape commands

This sample shows a text sent to terminal inside a DEC-VT print command.

The escape character (0xc1B) is noted by <ESC>.

```
<ESC>[5iGO TO ALPHA LIMA 8<ESC>[4i
```

“*go to alpha lima eight*” will be heard.

Configuration File

Here below there's the description of the Voice configuration file.

The configuration file follows a “dot-ini” format with sections (delimited by brackets), keys (keywords inside the section, next to "=" sign) and values (text next to "=" sign). Sections and keys are in special font: **SECTION**. Values are in **green**.

Global definitions about the voice engine

TTS (Text-to-speech) section

[TTS_DEF]

This section groups the global Text-To-Speech (TTS) settings, and enables to process the screen and the printer for voice actions.

VoiceSpeed=70

Default speed of TTS engine (0 to 100). User may modify this speed.

VoiceVolume=70

Default volume of TTS engine (0 to 100). User may modify this speed.

TtsPrinterOn=Yes

Enables the TnVT printer commands (Esc 5 I) to be captured by TTS engine instead of sending them to printer, so any text sent to printer will be read out loud to user by TTS. This mode is intended for use in TTS only (voice guide) applications. Do not use this mode together with triggers.

TtsScreenOn=Yes

Allows the PDA to say by TTS some texts displayed on screen.

The application scans the display data searching beginning and end tags (values defined on configuration file), and once found the tags, TTS reads aloud the delimited text.

This mode is intended for use in TTS only (voice guide) applications. Do not use this mode together with triggers.

Text delimiters on display

TtsScrSayTagBeg=SAY:

Beginning pattern of the text to look for on emulated terminal display to be read out loud by TTS.

TtsScrSayTagEnd=:SAY

End pattern of the text to look for on emulated terminal display to be read out loud by TTS.

TtsScrSplTagBeg=SPL:

Beginning pattern of the text to look for on emulated terminal display to be **spelled** by TTS.

TtsScrSplTagEnd=:SPL

End pattern of the text to look for on emulated terminal display to be **spelled** by TTS.

User messages

Some messages that will be said to user at specific conditions.

MsgWelcome=Welcome

Message to say at start-up.

MsgConnecting=Connecting...

Message to say when connecting.

MsgConnected=Connected

Message to say when TelNet connects to host.

MsgDisconnect=End of connection

Message to say when TelNet connection ends.

Keys to control voice engine by user**KeyRepeat=2070**

Key to repeat last text speech.

KeyVolUp=2071**KeyVolDn=2072**

Keys to change the volume.

KeySpeedUp=2073**KeySpeedDn=2074**

Keys to change the TTS speed.

Special character translator options

TranslatorUse=No

TranslatorName=application\Wt\T\Fnt_PC850.cfg

Character translator used for special character codifications. Allows customizing character-by-character translations.

ASR (Automatic Speech Recognition) definitions section

Tuning

[ASR_DEF_TUNING]

This section groups the ASR engine settings to tune the ASR engine.

ReliabilityLvl=4500

Minimum reliability level (for digits) necessary to accept a word recognition. When a word is recognized, ASR engine returns a “reliability level” which is the confidence level that the engine assigns to the recognition. A low reliability level denotes a confusing recognition that will be rejected.

ReliabilityPause=6000

Minimum reliability level (in pause mode) necessary to accept word recognition.

ReliabilityCmd=5500

Minimum reliability level for commands (functions and controls) necessary to accept a keyword recognition (controls or functions).

ReliabilityExp=300

Minimum reliability level for expected data (see expected in trigger ASR section) necessary to accept a word recognition.

AsrThreshold=50

Audio signal threshold (0 to 100) to detect a voice speech. Near to 0 needs to speak very loud. Near to 100 needs to speak whispering.

In a noisy environment it is needed to set this value in a low level (20) to avoid background noise to start recognition engine. Of course the user must speak very loud. In quiet environments it is possible to set a high value (60) to improve user working comfort. The values may change following hardware (PDA, headset).

SyncTs=Yes

Not implemented feature, set to Yes.

StartBeep=\windows\voixtreme.wav

When the ASR engine is ready to hear the user (the voice announcement is finished) a beep is performed to advise operator to speak. This beep will be configured to adapt to user preferences and noise environment.

StartBeepPause=\windows\voixtreme.wav

In pause mode, a different beep may be performed..

LogDo=Yes

A log file may log the user voice activity, such as:

- Time-stamp
- Current trigger used
- Voice announcement
- ASR result (word and reliability level).

This file will be automatically deleted if it reaches a size of 1Mb.

LogFile=\WTnAsrLog.log

The log file location and name.

Custom grammars

[ASR_DEF_GRAMMARS]

This section groups the names of the customized grammars (up to 10).

The grammars names are not significant for application, that just must match the name of the grammar in BNF file. It allows to independently activate groups of words (usually keywords) depending on application requirements. It is useful to activate in one trigger only the keywords that are useful in the specific transaction.

CustomGrammar0=<name>

Name of the grammar referenced by the input mode parameter [VOICETRIGGER_ASR](#) /
Grammars=0

CustomGrammar1=<name>

Name of the grammar referenced by the input mode parameter [VOICETRIGGER_ASR](#) /
Grammars=1

...

CustomGrammar9=<name>

Name of the grammar referenced by the input mode parameter [VOICETRIGGER_ASR](#) /
Grammars=9

...

Keywords

[ASR_DEF_KEYWORDS]

This section groups the usage of some keywords. The keywords must be defined in the grammar to be recognized. Upon a keyword is recognized, the action performed by this keyword is defined in this table.

If a keyword is defined in the grammar but is not defined in this table it will be consider as data and sent "as is" as user input.

If a keyword is defined in this section, but is not present in the grammar it will never be recognized.

To be recognized, a keyword must be in an active grammar.

ResRepeat=Repeat

This is the keyword of the "suspend" grammar the user should say to ask for a repetition (voice announcement before ASR) during ASR recognition.

ResAccept=Ok

This is the keyword of the "controls" grammar the user should say to accept an action.

ResCancel=Cancel

This is the keyword of the "controls" grammar the user should say to cancel an action.

ResPause=speech-pause

This keyword of the "suspend" grammar allows to switch to pause state from active state.

ResResume=speech-resume

This keyword of the "resume" grammar allows to switch from pause state to active state.

ResInfo=information

This keyword allows to heard the configured information sentence.

SpeedUp=speed-up

This keyword allows set up by user the TTS speed (0 to 100).

SpeedDn=speed-down

This keyword allows set down by user the TTS speed (0 to 100).

VolumeUp=volume-up

This keyword allows set up by user the TTS volume (0 to 100).

VolumeDn=volume-down

This keyword allows set down by user the TTS volume (0 to 100).

AsrSuffix=ready

This is the suffix to be said to user (by TTS) in voice repeat sequence.

KeywordFn01=January|first

This keyword of the “functions” grammar, allows to generate the F1 function key. The sequence sent to host depends of the terminal emulation selected.

It is possible to set one or more keywords (separated by a “pipe”) to generate a function key. In case of multiple words for one key, it is recommended to set in different grammars to activate them separately. This feature is useful to handle different actions on the same key depending on context. I/e in one trigger the F1 key means to skip a picking location, in other trigger it will mean to cancel an order, in this case it is possible to set “skip|cancel” and activate one of them depending on context.

Up to 20 keywords can be set to perform from F1 to F20 function keys.

PauseInfo=In pause. Say speech-resume to continue.

When the voice system is in pause mode, this sentence will be said to user in case of “information” (ResInfo) keyword recognition.

Keyboard keys by keywords**[ASR_DEF_KBD_KEYWORDS]**

This section groups the keywords that will perform some specific keyboard actions (other than function keys) needed on application by a recognized keyword.

When the keyword is recognized, the associated keyboard scan-code is sent to keyboard processor. This feature allows to perform actions intended to be performed by user on keyboard (i/e ctrl-x to close the telnet session).

Count=X

The total of couples Scc and Kwd.

Scc00=0018

The scan-code 00 to be generated when the associated keyword is recognized.

Kwd00=disconnect

The keyword to generate the associated scan-code. It is mandatory needed to this keyword to be present in any grammar to be recognized.

ASR engine calibration**[ASR_DEF_CALIBRATE]**

This section groups the options for the ASR calibrate procedure. The Asr calibrate allows the ASR engine to adapt to current user and noise environment conditions.

The data calibrate file (AsrCalibrate.txt) is a configurable file containing a list of testing words to show/say to user and recognize.

This file has any number of text lines (separated by CR), and each line has four fields separated by semicolon (;).

The fields of each line are:

<type>;<Result>;<Display text>;<Voice announce>

- * The Type is C for commands and D for digits
- * The 'result' string to obtain (must be exactly the same than the grammars)
- * The 'display text' to be shown in the dialog box.
- * The 'voice announce' to be said to the user.

TargetReliability=5000

This is the target reliability level to apply to validate a recognized word.

TargetOk=5

This is the number of validated words the user is challenged to perform to terminate the calibrate procedure.

A valid word is a good response + the reliability level greater than **TargetReliability**.

SequenceMode=0

The mode to play the customized calibrate file.

0=True random mode with repetition

1=Random mode without repeating the same word

2=Sequential

3= By user selection (combo box)

SkipOnEr=Yes

Yes= If the word is not recognized, goto next word.

No= Repeat current word until it is recognized (regardless the reliability level reached).

Grammars=DCF0123456789

The grammars to activate in the calibration procedure.

AnnounceOk=Valid

The voice announce to user when the word is valid (good response and good reliability level).

AnnounceNER=Recognized

The voice announce to user when the word is a good response but a not enough reliability level.

AnnounceER=Invalid

The voice announce to user when the word is a not a good response.

Welcome=Voice calibrate procedure.

The phrase said when the procedure starts.

Goodbye=End of calibration.

The phrase said at ending the procedure.

Voice Triggers

Voice triggers are several sections that allow to perform voice actions depending on conditions to match on the emulated screen. Each trigger is composed of 3 sections:

- The conditions section (conditions to be satisfied in order to run the trigger actions): [VOICETRIGGER_XX].
- The TTS section (TTS actions to do): [VOICETRIGGER_TTS_XX]
- The ASR section (ASR actions to do): [VOICETRIGGER_ASR_XX]

One (at least) or more triggers can be defined.

Trigger 1 ...

[VOICETRIGGER_01]
[VOICETRIGGER_TTS_01]
[VOICETRIGGER_ASR_01]

Trigger 2 ...

[VOICETRIGGER_02]
[VOICETRIGGER_TTS_02]
[VOICETRIGGER_ASR_02]

Trigger 3 ...

[VOICETRIGGER_03]
[VOICETRIGGER_TTS_03]
[VOICETRIGGER_ASR_03]

...

Voice trigger conditions section

The conditions to match on the emulated screen to run the trigger should be defined on this section.

There are two kinds of conditions: cursor position, on one hand, and three text conditions, on the other. Text conditions are excluding conditions (Logical AND), i.e. all the text conditions must be true to run the trigger.

[VOICETRIGGER_XX]

On=Yes

Enables/ disables this trigger.

Up to 5 text matches on the emulated screen

Type_1=A=

Type of match on screen: A Alphabetic, N Numeric.

- Equal (A=, N=)
- Not equal (A!=, N!=)
- Greater (A>, A<)
- Lesser (A<, N<)
- Greater or equal (A>=, N>=)
- Lesser or equal (A<=, N<=)

Row_1=4

Row position where to look for a text.

Row=0 and Col=0 means 'unused condition'.

Col_1=4

Column position where to look for a text.

Row=0 and Col=0 means 'unused condition'.

Match_1=Show System Summary

Text to look for at Row_1 / Col_1 position. To consider ending spaces, text should be delimited by quotes "".

If no text is supplied, the condition is not evaluated (TRUE).

To compare two areas in screen set here an {r,c,l} tag to compare the text denoted by {r,c,l} against the text located at Row_1, Col_1 position.

Cursor position condition

CursorAtRow=8

Cursor row position to evaluate. 0 (zero) means in any row.

CursorAtCol=20

Cursor column position to evaluate. 0 (zero) means in any column of the row.

Row=0 and Column=0, means anywhere in the screen.

Voice trigger TTS actions section

[VOICETRIGGER_TTS_XX]

These are the Text-To-Speech actions to do if the screen conditions match the voice trigger conditions. Constant text (supplied in the trigger) and variable text (extracted from emulated screen) will be said to user.

DoTts=Yes

Enables / disables TTS actions in this trigger.

TtsSay01=any constant text {r,c,l} another constant text {r,c,l} final text.

Combination of constant text (normal text in key) and variable text (delimited by curly brackets {}).

Variable text is indicated by three numeric values denoting the starting point of variable text by row and column, and its length “{r,c,l}”.

All the text is read out loud, except for those parts surrounded by special spell tags.

Special text tags <SPL><SAY> delimit text to spell, i.e.the text “location

<SPL>AB8<SAY> to go” will be read as “location *y be eight to go*”.

The last value (l) in the {r,c,l} tag may be preceded by a letter to indicate special operations:

- {r,c,Tl} means alphabetic spaces trim (no spaces)
- {r,c,Nl} means numeric trim (no leading zeroes)

The {r,c,l} tag can have an additional operator to handle some options {r,c,l,o}

- {r,c,l,Lo} means Left suppression of “o” characters. I/e {3,5,N20, L2) will take at row 3 and column 5 twenty characters, then suppress leading zeroes by numeric trim, then suppress leftmost 2 characters.
- {r,c,l,Ro} means Left suppression of “o” characters. I/e {3,5,T20, L2) will take at row 3 and column 5 twenty characters, then suppress spaces by trim, then suppress rightmost 2 characters.

The starting tag <X> allows to conditionally say the line depending on the previous trigger execution:

- <=> The same trigger. I/e: “TtsSay01=<=>Loop on the same trigger” will be said to user only if the previous trigger was the current one.
- <!> Not the same trigger. I/e: “TtsSay01=<=>New trigger” will be said to user only if the previous trigger was another than current.
- <=3> The trigger 3
- <!5> Not the trigger 5.

TtsSay02= any constant text {r,c,l} another constant text {r,c,l} final text.

Several TTS actions are allowed.

Voice trigger ASR actions section

[VOICETRIGGER_ASR_XX]

These are the ASR actions to perform in this trigger:

DoAsr=Yes

Enables / disables ASR for this trigger.

Grammars=DCF

Grammars to use in this trigger. A grammar is a set of rules and words that ASR engine use for speech recognition. The grammars are defined in cd_xx.bnf file that can be customized. Standard ASR engine includes three grammars:

- “D” for **digits** (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).
- “C” for **commands** (Cancel / Accept)
- “F” for **functions** (function keys keywords)
- “0” to “9” for **custom** grammars.

Other grammars (suspend / resume) are internally managed by ASR engine.

LenMin=0

Minimum voice input length expected. Inputs shorter than this will be rejected, and a new input will be requested.

LenMax=32

Maximum voice input length expected. Inputs bigger than this will be rejected, and a new input will be requested.

Confirm=Yes

Uses the *Confirm* or the *Simple* voice input algorithm to perform ASR input action. See [Input Modes](#) paragraph for an explanation of ASR algorithms.

ConfirmDigits=%s is correct?

Question that will be asked to user “Confirm” mode. The “%s” tag is replaced by the input value.

Repeat=Yes

Activates Repeat algorithm to perform ASR input action. See [Input Modes](#) paragraph for an explanation of ASR algorithms.

SayOneTime=Yes

The voice announce is said before any ASR recognition. The if ASR recognition is not valid, the voice announce will be automatically repeated (every 10 seconds approximately) or not (set SayOneTime=Yes).

CancelDo=Yes

Allows user to cancel a data input, and sends cancel sequence to host.

Cancel=\1B\41

Sequence sent to host when the user cancels the input without any data.

CancelConfirmation=Do you wish to cancel?

Question that will be asked to user if a Cancel command is said in “Confirm” mode.

Check=Text or {r,c,l,o}

The user input may be checked against this value to be validated. If not equal, the input is rejected and the ASR is started again. If equal, the input is validated, and sent to host. This is suitable to validate user input that must match any data on screen (I/e 3 last digits of a product code).

The check is defined by a 4 operators tag {r,c,l,o}

r= Row, c=Column, l=Length, o=Match operation.

The operation can be Mn (Match n digits) or Xn (Checksum match on n digits)

- {4,7,N15,M3} will take 15 characters of text at row 4, column 7, perform a numeric trim, then match the last 3 characters on user input.
- {4,7,T15,M3} will take 15 characters of text at row 4, column 7, perform a trim, then match the last 3 characters on user input.
- {4,7,15,X3} will take 15 characters of text at row 4, column 7, perform a check-sum, then match the last 3 characters on user input. The check-sum algorithm is specific to voiXtreme, an utility is available to calculate this algorithm. This method will be suitable to generate pseudo-random verification digits (i/e on locations codes).

Format=Text or {r,c,l} {\$}

The output text can be formatted to add fixed or variable text to user input.

- The {\$} tag represents the user input.
- Some {r,c,l} tags can add text form the display
- Constant text can be added.

Example: Format=9999999{\$},00{\09}{4,3,5} will prefix user input by constant text “9999999”, then follow it by constant text “,00” then a tabulation char, and finish by 5 characters text located at row 4 column 3. I/e if user input is “123” the data sent to host will be “ 9999999123,00<TAB>ABCDE”.

Terminator=\0D

Terminator string that will be sent to host after the ASR formatted data (added as trailing) to validate input (in this case, Enter).

Information=informational text to user requesting help

The sentence to say to user if the **ResInfo** keyword is recognized. This sentence is intended to help the user about actions to do on this trigger in case of help (i/e “go to specified location then say the check digits”, or “take the announced quantity then validate it).

CponfirmInfo=informational text to user requesting help

The sentence to say to user if the **ResInfo** keyword is recognized. This information will be said during confirmation algorithm. This sentence is intended to help the user about validation of data (i/e “say accept to validate or cancel to say another quantity”)

Sample configuration file

Screen and Trigger samples

This sample explains, on one hand, the trigger configuration necessary to do a voice input for a Bin-Location picking operation and, on the other, the trigger configuration to perform a quantity input in two screens.

Bin location screen

```
** Order Picking OPL-LOC
User : 0234 John JUAREZ
Order Number : 1259-AA0
Line : 05 / 23
Type: Mandatory
Bin Location : AB- 15
Bin-Loc ID: 
```

In this case, a simple algorithm is used as the two digit location ID known by system, then the input is sent without confirmation and the host sends a new screen (if match) or sends the same (if not match).

```
PDA "go to location ey be fifteen please"
User "count five seven ok"
PDA "fifty seven"
PDA to host "57<ENTER>"
```

```
[VOICETRIGGER_01]
On=Yes
Type_1=A=
Row_1=1
Col_1=1
Match_1=** Order Picking OPL-LOC
Row_2=0
Col_2=0
Match_2=
CursorAtRow=7
CursorAtCol=0
```

```
[VOICETRIGGER_TTS_01]
DoTts=Yes
```

```
TtsSay01=Go to
TtsSay02=Location <SPL>{6,19,2}<SAY>{6,22,3}
TtsSay03=Please
```

```
[VOICETRIGGER_ASR_01]
DoAsr=Yes
Grammar=DC
LenMin=2
LenMax=2
Confirm=No
Repeat=Yes
SayOneTime=Yes
CancelDo=Yes
CancelConfirmation=Do you wish to cancel?
Cancel=\1B
Format={$}
Terminator=\0D
Information=Say bin location code.
```

Quantity at location screen

```
** Order Picking OPL-QTY
User : 0234 John JUAREZ
Order Number : 1259-AA0
Line : 05 / 23
Type: Mandatory
Bin Location : AB- 15
Product : 1238767653490
Quantity: 
```

In this case a validate algorithm is used to confirm input quantity.

PDA “product four hundred ninety, give quantity”

User “count one four four ok”

PDA “one hundred and forty for”

User “accept”

PDA to host “144<ENTER>”

```
[VOICETRIGGER_02]
On=Yes
Row_1=1
Col_1=1
Match_1>** Order Picking OPL-QTY
Row_2=0
Col_2=0
Match_2=
CursorAtRow=8
CursorAtCol=0
```

```
[VOICETRIGGER_TTS_02]
```

```
DoTts=Yes
TtsSay01=Product {7,23,3}
TtsSay02=Give quantity

[VOICETRIGGER_ASR_02]
DoAsr=Yes
Grammar=DC
LenMin=1
LenMax=6
Confirm=Yes
Repeat=No
CancelDo=Yes
CancelConfirmation=Do you wish to cancel?
Cancel=\1B
Format={$}
Terminator=\0D
Information=Say the product quantity to pick.
ConfirmInfo=Confirm the quantity by accept, or say cancel.
```