

Настройка голосового диалога системы MBG.GOLAS

1. Настойка заголовка диалога голосового терминала

Файл с заголовком диалога называется voice_config_header.xml, находится в папке tpbv\bin терминала. Данный файл содержит информацию о подключениях, доступных терминалу, системных фразах и командах.

В файле можно описать следующие данные:

<server>192.168.0.1</server> - ip адрес или dns имя основного сервера

<port>5007</port> - порт основного сервера

<start_state>login</start_state> - имя шага диалога, с которого он начинается при запуске голосового клиента

<exit_state>exit</exit_state> - имя шага, на который диалог попадает по команде «exit»

<accept_word>okay</accept_word> - команда, которой надо подтверждать ввод данных

<voice_logging>on</voice_logging> - передавать(on) или не передавать(off) на голосовой сервер лог терминала

Лог передаётся на сервер, указанный в параметре voice_server(см. ниже) в виде строки формата

```
voicelog|%speaker%|%macaddr%|%ipaddr%|%time%|%text%|%login%
```

где

%speaker% - субъект, осуществивший генерацию логируемого текста(Н-человек, S-сканер ШК, С-терминал)

%macaddr% - мак адрес терминала

%ipaddr% - ip адрес терминала

%time% - время терминала

%text% - текст

%login% - логин, под которым работали на терминале в момент генерации текста

<voice_server>192.168.0.1</voice_server> - ip адрес или dns имя голосового сервера(может совпадать с основным)

<voice_port>4388</voice_port> - порт голосового сервера(может совпадать с основным)

<start_frase>Назови логин</start_frase> - Фраза, которую терминал скажет после запуска

<exit_frase>Скажи закончить работу ещё раз</exit_frase> - Фраза для подтверждения команды «exit»

<reboot_frase>Скажи система перезапуск ещё раз</reboot_frase> - Фраза для подтверждения команды «reboot»

<serialkey_path>tpbv\bin\serialkey.txt</serialkey_path> - путь к файлу с серийным номером терминала

<message_log>>true</message_log> - писать(true) или не писать(false) лог на диск терминала в файл tpbv\bin\messages.log. При каждом запуске терминала данный файл очищается, оставляя в нём только последние 20 тысяч строк

<enable_scanner>scan</enable_scanner> - Что надо сказать, чтоб включить сканер. Если пусто, сканер будет автоматически включаться на шагах, где он разрешён. Данная команда поддерживается только для сканеров, у которых есть возможность инициализировать триггер с внешнего устройства.

<scanner_delay>3000</scanner_delay> - Сколько миллисекунд будет работать сканер после включения или последнего сканирования. Данная команда поддерживается только для сканеров, у которых есть возможность завершать работу сканирующей головки с внешнего устройства. Если заполнен параметр delay_after_scan, то это значение будет определять только время работы ДО первого сканирования.

<delay_after_scan>2000</delay_after_scan> - Сколько миллисекунд будет работать сканер после последнего сканирования. Данная команда поддерживается только для сканеров, у которых есть возможность завершать работу сканирующей головки с внешнего устройства.

<scanner_name>WT63</scanner_name> - Имя подключенного Bluetooth сканера.

2. Настройка голосового диалога

Файл диалога называется voice_config.xml, находится в папке trbv\bin терминала. Данный файл содержит информацию о голосовом диалоге, которые будут осуществлять взаимодействие терминала и человека. С точки зрения архитектуры, голосовой клиент представляет из себя машину состояний, которая переходит с шага на шаг диалога в зависимости от условий, описанных внутри шагов.

В файле можно описать следующие данные:

<help> - Расшифровка слов из тэга <valid_reply> для команды "help". Голосовой терминал оперирует командами, написанными латиницей с определением их произношения на русском языке в файле грамматики. Данный список требуется, чтобы расшифровать эти команды для пользователя, если он хочет услышать список доступных команд.

<word> - расшифровка одной команды

<name>ready</name> - внутреннее имя команды

<value>готов</value> - произношение этой команды согласно грамматики

<states> - список шагов диалога

<login> - Имя шага

<request>Скажи логин</request> - запрос, что говорит система при команде «repeat»

<valid_reply>number|cancel</valid_reply> - Команды, которые будут обрабатываться в этом шаге. Все остальные команды будут игнорироваться. Если в данном поле указать null, то, при попадании на этот шаг, система не будет ожидать никакого ответа от пользователя, а сразу начнёт обработку. Если заполнить этот параметр значением any, то система примет любую команду

<set_before_send> - список присвоений переменных ДО отправки запроса на сервер. Помимо переменных, которые можно определить в данном разделе, в любой момент доступны следующие системные переменные:

%SERIALKEY% - серийный номер терминала

%BTSCANNER_SERIALKEY% - серийный номер подключенного Bluetooth сканера ШК

%SCANNER% - как была введена последняя команда, 0 – голосом, 1 – сканером ШК

%MAC_ADDRESS% - мак адрес терминала

%IP_ADDRESS% - ip адрес терминала

%TIMESTAMP% - текущая дата-время терминала в формате dd-mm-yyyy hh24:mi:ss

%TIMESTAMP_MS% - текущая дата-время терминала в формате dd-mm-yyyy hh24:mi:ss:ms

%TIMESTAMP_METRO% - текущая дата-время терминала в формате dd-mm-yy hh24:mi:ss

<set_after_send> - список присвоений переменных ПОСЛЕ отправки запроса на сервер.

<set> - присвоение переменной

<name>%CONFIG%</name> - имя переменной, таблицы или поля таблицы

<type>string</type> - тип переменной, допустимы string, number, table, setfield

<value>'10.09.2021'</value> - присваиваемое значение.

В данное поле можно вводить выражения, которые будут вычисляться перед присваиванием. Для выражений допустимо использовать различные функции и операторы в формате .NET Expression. Подробное описание формата можно найти здесь:

[DataColumn.Expression Свойство \(System.Data\) | Microsoft Docs](#)

Также доступны дополнительные функции:

INT(<число>) – округление числа до ближайшего целого

ROUND(<число>) – отбросить дробную часть числа

UCASE(<строка>) – привести строку в верхний регистр

LCASE(<строка>) – привести строку в нижний регистр

Для переменной типа string значение выражения должно представлять строку, для number – число. Для типа table – допустимы значения

next – перейти на следующую строку. Если дошли до конца таблицы, то выполнение next проведёт к появлению во всех полях таблицы значения null

prev – перейти на предыдущую строку

first – перейти на первую строку

addline – добавить новую строку

delete – очистить таблицу, удалить все строки

<sendstring>user_login|%login%|%messageid%|%MAC_ADDRESS%|%shift%</sendstring> - Шаблон запроса, отправляемого на сервер. Имена внутри знаков % подменяются на значения из буфера переменных. Имя запроса должно стоять самым первым и отделяться от параметров символом |. Данное имя будет использовано в дальнейшем как имя таблицы, где будет храниться ответ от сервера

<receivestring>%err_code%|%username%|%sessionid%</receivestring> - Шаблон ответа от сервера. Имена, указанные внутри знаков % будут сохранены в буфер переменных. Ответ от сервера всегда сохраняется как таблица, даже если там только одна строка. Переменные в данном случае являются полями таблицы. После получения ответа в переменные попадает значение из первой строки таблицы. Также создаётся переменная, имя которой совпадает с именем таблицы, в ней хранится текущий номер обрабатываемой строки. Строки нумеруются, начиная с 0.

При переходе между строк таблицы с помощью next, prev, first(см. выше описание **<value>**), в переменные-поля автоматически попадают значения из строки, на которую перешли. Если необходимо изменить значение поля таблицы, следует использовать такое присвоение:

```
<set>
```

```
  <name>GetPalletIDs.pal_status</name>
```

```
  <type>setfield</type>
```

```
  <value>9</value>
```

```
</set>
```

Где в <name> указано <имя таблицы>.<имя поля>. Если просто поменять значение поля через присваивание переменной, то при переходе между строк таблицы старое значение пропадёт.

<clear_voice_buffer>false</clear_voice_buffer> - Нужно ли(true/false) очищать буфер переменных при переходе в этот шаг

<repeat>false</repeat> - Нужно ли(true/false) повторять то, что сказал пользователь и ждать подтверждения. Если нет тега <reply_to_repeat>,повторять будет все команды

<reply_to_repeat>close|skip|empty|problem|cancel</reply_to_repeat> - список команд, которые пользователь должен подтверждать. Подтверждение выглядит следующим образом:

Пользователь: <команда>

Терминал: Сказал <команда>

Пользователь: <команда подтверждения (см. <accept_word>)> или <отмена>

<stringname>login</stringname> - Под каким именем переменной сохранится то, что сказал пользователь. Имена чувствительны к регистру букв

<scanner>true</scanner> - Допустимо ли на этом шаге использовать сканер ШК(true/false)

<scanner_status>on</scanner_status> - Подключиться(on) или отключиться(off) от сканера Voxter Scan. Данный параметр не работает на сканеры сторонних производителей.

<server>voice</server> - сервер, на который будет отправлено сообщение. По умолчанию – на голосовой сервер, для значения voice – на голосовой.

<encoding>ASCII</encoding> - кодировка сообщения. Допустимы значения UTF-8(по умолчанию) или ASCII

<async>true</async> - Отправка запроса в синхронном(false), по умолчанию, или в асинхронном(true) режиме. При асинхронном запросе терминал не сможет получить от сервера никаких данных и в **receivestring** должен быть указан шаблон 0.

<load_config> - загрузка новых файлов конфигурации после обработки данного шага. Файлы должны находится в папке bin

<header_file>voice_config_header_pbl.xml</header_file> - имя файла с заголовком

<config_file>voice_config_pbl.xml</config_file> - имя файла с диалогом

<to_state>pbl_start_point</to_state> - начальный шаг в загружаемом диалоге

<nextstate> - Список шагов, на которые можно перейти из данного шага. Переход производится, если выполнится условие, описанное внутри правила перехода. Правила проверяются по очереди с первого, если условие будет выполнено, то произойдёт переход и нижеописанные правила проверяться не будет.

<state> - правило перехода на другой шаг

<name>say_shift</name> - имя нового шага

<set_subgrammar>1</set_subgrammar> - номер подграмматики, которая будет активирована, если будет выполнен переход на новых шаг. По умолчанию останется прежняя подграмматики. После запуска голосового клиента активна подграмматика #1.

<condition> - список условий. Для перехода должны быть выполнены все условия в списке

<expression> - условие. Если этот тэг оставить незаполненным, то будет выполнен безусловный переход

<operand1>%login%</operand1> - первый операнд

<operation>EQ</operation> - операция сравнения, допустимы значения: EQ =; NE <>; GE >=; GT >; LE <=; LT <; Для строк можно использовать только EQ и NE, остальное только для чисел

<operand2>cancel</operand2> - второй операнд

<answer>скажи номер смены</answer> - То, что скажет система при переходе в новое состояние, если условие перехода выполнено. Здесь можно просто написать текст, тогда он будет воспроизведён через систему синтеза речи, установленную на устройстве. Также можно с помощью функции \$FILE{<имя файла>} воспроизвести какой-либо файл из папки tpbv/audiofiles. Помимо этого, есть возможно использование функция для проговаривания чисел с помощью записанных звуковых файлов:

\$NUMB{<число>} – произнести число в десятичной форме, например: 123- сто двадцать три

\$NUMS{<число>} – произнести число поциферно, например: 123 – один два три

Для возможности использования функций NUMB и NUMS, в папке tpbv/audiofiles должны быть звуковые файлы для всех цифр и десятичных названий круглых чисел:

0, 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, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1000000