

PascalABC.NET

Синтез и распознавание речи

Варианты установки

Данный модуль позволяет синтезировать и распознавать речь в системе программирования PascalABC.NET, при этом постоянное подключение к Интернету не требуется.

Основан на платформе *Microsoft Speech Platform 11.0*.

Для работы предусмотрены два варианта установки:

1. Установка из папки Installation (слайд 4) – в ней собраны все необходимые для работы файлы, более удобный вариант, однако рекомендуется лишь в том случае, если вы доверяете этому сайту и автору.
2. Установка соответствующих пакетов с сайта Microsoft (слайд 3) – описан процесс установки из официальных источников, приведены соответствующие ссылки.

Установка (с официального сайта)

Для работы синтезатора речи необходимо установить платформу Microsoft Speech Platform, а также соответствующие «голоса» и «распознаватели»:

1. Скачать и установить Runtime – <https://www.microsoft.com/en-us/download/details.aspx?id=27225>
2. Скачать дополнительные языки – рекомендуются английский (британский и американский), русский – <https://www.microsoft.com/en-us/download/details.aspx?id=27224>. По ссылке список языков и распознавателей, из них те, что имеют «TTS» в названии – голоса (text-to-speech), «SR» – распознаватели (speech recognizer). Рекомендуется скачать те голоса и распознаватели, в названии которых есть текст «en-GB», «en-US» и «ru-RU» – их надо устанавливать. После установки перезагрузить компьютер.
3. В папке с программой должна находиться библиотека Microsoft.Speech.dll, её можно найти отдельно, или же скачать и установить SDK (идёт в составе) – <https://www.microsoft.com/en-us/download/details.aspx?id=27226>.
4. После установки SDK необходимо найти библиотеку Microsoft.Speech.dll в папке, куда был установлен набор инструментов разработчика, по умолчанию это *c:\Program Files\Microsoft SDKs\Speech\v11.0\Assembly\Microsoft.Speech.dll*. Скопировать эту dll в папку с программами PascalABC.Net.

Установка (из архива)

Все необходимые файлы можно установить из папки Installation (делайте это только в том случае, если доверяете данному сайту):

1. Шаг 1: Установить Runtime – выберите в папке «*Step1 - Runtime*» соответствующую версию (x86 – для 32-битной ОС, x64 – для 64-битной).
2. Шаг 2: Установите языки – рекомендуются английский (британский и американский), русский. В папке «*Step 2 - Languages*» собраны основные языки и распознаватели, из них те, что имеют «TTS» в названии – голоса (text-to-speech), «SR» – распознаватели (speech recognizer). После установки перезагрузить компьютер.
3. В папку с программой на Паскале скопируйте библиотеку Microsoft.Speech.dll из папки «*Step3 - DLL*».

Проверка установки

Проверить корректность установки можно запуском простой программы на PascalABC.NET:

```
uses  
SpeechABC;  
begin  
    SpeechInfo;  
end.
```

Вывод (пример):

Для корректной работы убедитесь, что у вас в системе установлено устройство воспроизведения звука и микрофон.

Установлены русскоговорящие голоса:

Microsoft Server Speech Text to Speech Voice (ru-RU, Elena)

Установлены англоговорящие голоса:

Microsoft Server Speech Text to Speech Voice (en-US, Helen)

Microsoft Server Speech Text to Speech Voice (en-US, ZiraPro)

Microsoft Server Speech Text to Speech Voice (en-GB, Hazel)

Установлены модули распознавания русского языка:

Microsoft Server Speech Recognition Language - TELE (ru-RU)

Установлены модули распознавания английского языка:

Microsoft Server Speech Recognition Language - TELE (en-GB)

Microsoft Server Speech Recognition Language - TELE (en-US)

Голоса и распознаватели обнаружены, модуль готов к работе.

Синтез речи

Для синтеза речи используется устройство воспроизведения звука по умолчанию.

Воспроизвести звуковую фразу, представленную строкой, можно с помощью процедуры *Say(string)*. При этом автоматически определяется язык (предусмотрено определение русского или английского), для воспроизведения будет использован первый по порядку голос из соответствующего списка установленных голосов. Определение языка выполняется по преобладанию символов русского или английского алфавита (используются первые 100 символов строки).

Вызов процедуры является блокирующим, то есть процедура не завершится, пока не будет произнесена фраза.

Простейшая «говорилка»

В первом случае фраза будет воспроизведена русским голосом, во втором – первым по порядку из установленных английских или американских голосов.

Модуль не различает британский английский и американский английский (так как по английскому тексту этого сделать нельзя).

```
uses SpeechABC;  
begin  
  Say('Использование синтеза речи - это просто!');  
  Say('It is easy to use text-to-speech in Pascal ABC');  
end.
```

Объект для воспроизведения речи

Использование объекта класса `Speaker` позволяет явно указать язык для воспроизведения речи, узнать количество установленных голосов для соответствующего языка, а также изменять громкость воспроизведения (по умолчанию установлена 100% громкость).

```
uses SpeechABC;
begin
  var Arnold := new Speaker(Languages.English);
  Arnold.Volume := 100;
  for var i := 0 to Arnold.VoicesCount-1 do
    begin
      Arnold.SelectVoice(i);
      Arnold.Speak('Hasta la vista, baby!');
    end;
  end.
```


Синхронный и асинхронный синтез речи

Синтез речи можно выполнять в асинхронном режиме – с помощью метода `SpeakAsync(string)` класса `Speaker`. В этом случае фраза начинает проговариваться, но выполнение программы сразу продолжается – как будто фраза сказана «мгновенно». `SpeakAsync` – асинхронный метод, возвращающий управление до завершения процесса синтеза фразы.

Если метод `SpeakAsync` использован несколько раз подряд, то образуется очередь из фраз, которые будут сказаны друг за другом, при этом процесс выполнения программы продолжается.

Синтез фраз даже при асинхронном режиме является последовательным – они не накладываются друг на друга даже при использовании двух объектов класса `Speaker`.

Использование двух методов `SpeakAsync` подряд может приводить к неверному порядку произнесения фраз, в этом случае рекомендует использовать `Sleep` (оператор паузы) с небольшой задержкой.

Асинхронный синтез речи

Вывод текста «Я всё сказала!» будет выполнен ещё в процессе произнесения первой фразы. Вторая и третья фраза могут быть перепутаны местами.

```
uses SpeechABC;
begin
    // Создаём говорилку для русского языка
    var sp := Speaker.Create(Languages.Russian);
    // Вызываем проговаривание четырёх фраз
    sp.SpeakAsync('Проверка речи. Я умею говорить!');
    Sleep(10);
    sp.SpeakAsync('Вторая фраза');
    sp.SpeakAsync('Третья фраза');
    // Вывод текста – эта фраза будет напечатана ещё до того, как закончится
    // произнесение первой фразы
    Writeln('Я всё сказала!');
end.
```

Распознавание речи

Распознавание речи выполняется объектом класса Recognizer. Порядок работы и ограничения:

1. Распознавание выполняется только для заранее указанных фраз, произвольный текст объект не распознаёт.
2. При распознавании указывается язык, поэтому распознавание смешанных фраз невозможно – например, нельзя одновременно распознавать русскую и английскую речь (но можно попробовать транслитерацию).
3. Для анализа речи используется устройство записи звука по умолчанию, выбрать другое в данный момент нельзя. Как правило, это микрофон веб-камеры (встроенной или внешней).
4. В данном модуле используется асинхронный режим распознавания. Это означает, что после старта распознавания выполнение программы продолжается, а при распознавании речи будет вызван обработчик. Поэтому в простейших программах, где используется только распознавание, надо позаботиться о том, чтобы программа не завершалась после запуска распознавания.

Создание и настройка объекта Recognizer

Для работы
необходимо создать
объект класса
Recognizer, указать
набор вариантов –
массив строк, а
также определить
обработчик события
распознавания

```
uses SpeechABC;

/// Простейший обработчик распознавания
procedure Response(phrase : string);
begin
    Writeln(phrase);
end;

begin
    // Объявляем массив строк для распознавания
    var Phrases := new string[] ('Привет', 'Зелёный', 'Музыка', 'Кино', 'Спорт')

    // Создаём объект для распознавания с указанием фраз вариантов
    var Ear := new Recognizer(Phrases);
    // Указываем обработчик
    Ear.OnRecognized := Response;
    // Запускаем распознавание
    Ear.Start;

    // В обычном режиме распознавание асинхронное, поэтому чтобы программа
    // не завершилась – бесконечный цикл
    while true do;
end.
```