.NET Framework

DeepSpeech Class

class

Concrete implementation of DeepSpeechClient.Interfaces.IDeepSpeech.

Public Functions

```
DeepSpeechClient.DeepSpeech.DeepSpeech(string aModelPath)
```

Initializes a new instance of DeepSpeech class and creates a new acoustic model.

Parameters

- `aModelPath`: The path to the frozen model graph.

Exceptions

- `ArgumentException`: Thrown when the native binary failed to create the model.

```
unsafe uint DeepSpeechClient.DeepSpeech.GetModelBeamWidth()
```

Get beam width value used by the model. If SetModelBeamWidth was not called before, will return the default value loaded from the model file.

Return

Beam width value used by the model.

```
unsafe void DeepSpeechClient.DeepSpeech.SetModelBeamWidth(uint aBeamWidth)
```

Set beam width value used by the model.

Parameters

- `aBeamWidth`: The beam width used by the decoder. A larger beam width value generates better results at the cost of decoding time.

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.DeepSpeech.AddHotWord(string aWord, float aBoost)
```

Add a hot-word.

Parameters

- `aWord`: Some word
- `aBoost`: Some boost

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.DeepSpeech.EraseHotWord(string aWord)
```

Erase entry for a hot-word.

Parameters

- `aWord`: Some word

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.DeepSpeech.ClearHotWords()
```

Clear all hot-words.

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe int DeepSpeechClient.DeepSpeech.GetModelSampleRate()
```

Return the sample rate expected by the model.

Return

Sample rate.

```
unsafe void DeepSpeechClient.DeepSpeech.Dispose()
```

Frees associated resources and destroys models objects.

```
unsafe void DeepSpeechClient.DeepSpeech.EnableExternalScorer(string
aScorerPath)
```

Enable decoding using an external scorer.

Parameters

- `aScorerPath`: The path to the external scorer file.

Exceptions

- `ArgumentException`: Thrown when the native binary failed to enable decoding with an external scorer.
- `FileNotFoundException`: Thrown when cannot find the scorer file.

```
unsafe void DeepSpeechClient.DeepSpeech.DisableExternalScorer()
```

Disable decoding using an external scorer.

Exceptions

- `ArgumentException`: Thrown when an external scorer is not enabled.

```
unsafe void DeepSpeechClient.DeepSpeech.SetScorerAlphaBeta(float aAlpha, float aBeta)
```

Set hyperparameters alpha and beta of the external scorer.

Parameters

- `aAlpha`: The alpha hyperparameter of the decoder. Language model weight.
- `aBeta`: The beta hyperparameter of the decoder. Word insertion weight.

Exceptions

- `ArgumentException`: Thrown when an external scorer is not enabled.

```
unsafe void DeepSpeechClient.DeepSpeech.FeedAudioContent(DeepSpeechStream stream, short [] aBuffer, uint aBufferSize)
```

Feeds audio samples to an ongoing streaming inference.

Parameters

- `stream`: Instance of the stream to feed the data.
- `aBuffer`: An array of 16-bit, mono raw audio samples at the appropriate sample rate (matching what the model was trained on).

```
unsafe string DeepSpeechClient.DeepSpeech.FinishStream(DeepSpeechStream stream)
```

Closes the ongoing streaming inference, returns the STT result over the whole audio signal.

Return

The STT result.

Parameters

- `stream`: Instance of the stream to finish.

```
unsafe Metadata DeepSpeechClient.DeepSpeech.FinishStreamWithMetadata(DeepSpeechStream stream, uint aNumResults)
```

Closes the ongoing streaming inference, returns the STT result over the whole audio signal, including metadata.

Return

The extended metadata result.

Parameters

- `stream`: Instance of the stream to finish.
- `aNumResults`: Maximum number of candidate transcripts to return. Returned list might be smaller than this.

```
unsafe string DeepSpeechClient.DeepSpeech.IntermediateDecode(DeepSpeechStream stream)
```

Computes the intermediate decoding of an ongoing streaming inference.

Return

The STT intermediate result.

Parameters

- `stream`: Instance of the stream to decode.

```
unsafe Metadata DeepSpeechClient.DeepSpeech.IntermediateDecodeWithMetadata(DeepSpeechStream stream, uint aNumResults)
```

Computes the intermediate decoding of an ongoing streaming inference, including metadata.

Return

The STT intermediate result.

Parameters

- `stream`: Instance of the stream to decode.
- `aNumResults`: Maximum number of candidate transcripts to return. Returned list might be smaller than this.

```
unsafe string DeepSpeechClient.DeepSpeech.Version()
```

Return version of this library. The returned version is a semantic version (SemVer 2.0.0).

```
unsafe DeepSpeechStream DeepSpeechClient.DeepSpeech.CreateStream()
```

Creates a new streaming inference state.

```
unsafe void DeepSpeechClient.DeepSpeech.FreeStream(DeepSpeechStream stream)
```

Destroy a streaming state without decoding the computed logits. This can be used if you no longer need the result of an ongoing streaming inference and don't want to perform a costly decode operation.

```
unsafe string DeepSpeechClient.DeepSpeech.SpeechToText(short [] aBuffer, uint aBufferSize)
```

Use the DeepSpeech model to perform Speech-To-Text.

Return

The STT result. Returns NULL on error.

Parameters

- `aBuffer`: A 16-bit, mono raw audio signal at the appropriate sample rate (matching what the model was trained on).
- `aBufferSize`: The number of samples in the audio signal.

```
unsafe Metadata DeepSpeechClient.DeepSpeech.SpeechToTextWithMetadata(short [] aBuffer, uint aBufferSize, uint aNumResults)
```

Use the DeepSpeech model to perform Speech-To-Text, return results including metadata.

Return

The extended metadata. Returns NULL on error.

Parameters

- `aBuffer`: A 16-bit, mono raw audio signal at the appropriate sample rate (matching what the model was trained on).
- `aBufferSize`: The number of samples in the audio signal.
- `aNumResults`: Maximum number of candidate transcripts to return. Returned list might be smaller than this.

DeepSpeechStream Class

class **DeepSpeechStream** : *public* **IDisposable**

Wrapper of the pointer used for the decoding stream.

Public Functions

```
unsafe DeepSpeechClient.Models.DeepSpeechStream.DeepSpeechStream(IntPtr ** streamingStatePP)
```

Initializes a new instance of DeepSpeechStream.

Parameters

- `streamingStatePP`: Native pointer of the native stream.

ErrorCodes

See also the main definition including descriptions for each error in Error codes.

enum **`DeepSpeechClient::Enums::ErrorCodes`**

Error codes from the native DeepSpeech binary.

Values:

```
DS_ERR_OK = 0x0000
DS_ERR_NO_MODEL = 0x1000
DS_ERR_INVALID_ALPHABET = 0x2000
DS_ERR_INVALID_SHAPE = 0x2001
DS_ERR_INVALID_SCORER = 0x2002
DS_ERR_MODEL_INCOMPATIBLE = 0x2003
DS_ERR_SCORER_NOT_ENABLED = 0x2004
DS_ERR_FAIL_INIT_MMAP = 0x3000
DS_ERR_FAIL_INIT_SESS = 0x3001
DS_ERR_FAIL_INTERPRETER = 0x3002
DS_ERR_FAIL_RUN_SESS = 0x3003
DS_ERR_FAIL_CREATE_STREAM = 0x3004
DS_ERR_FAIL_READ_PROTOBUF = 0x3005
DS_ERR_FAIL_CREATE_SESS = 0x3006
DS_ERR_FAIL_INSERT_HOTWORD = 0x3008
DS_ERR_FAIL_CLEAR_HOTWORD = 0x3009
DS_ERR_FAIL_ERASE_HOTWORD = 0x3010
```

Metadata

class **`Metadata`**

Stores the entire CTC output as an array of character metadata objects.

Property

property **DeepSpeechClient::Models::Metadata::Transcripts**

List of candidate transcripts.

CandidateTranscript

class **CandidateTranscript**

Stores the entire CTC output as an array of character metadata objects.

Property

property **DeepSpeechClient::Models::CandidateTranscript::Confidence**

Approximated confidence value for this transcription.

property **DeepSpeechClient::Models::CandidateTranscript::Tokens**

List of metada tokens containing text, timestep, and time offset.

TokenMetadata

class **TokenMetadata**

Stores each individual character, along with its timing information.

Public Members

string DeepSpeechClient.Models.TokenMetadata.Text

Char of the current timestep.

int DeepSpeechClient.Models.TokenMetadata.Timestep

Position of the character in units of 20ms.

float DeepSpeechClient.Models.TokenMetadata.StartTime

Position of the character in seconds.

DeepSpeech Interface

interface **IDeepSpeech**

Client interface for DeepSpeech

Subclassed by DeepSpeechClient.DeepSpeech

Public Functions

```
unsafe string DeepSpeechClient.Interfaces.IDeepSpeech.Version()
```

Return version of this library. The returned version is a semantic version (SemVer 2.0.0).

```
unsafe int DeepSpeechClient.Interfaces.IDeepSpeech.GetModelSampleRate()
```

Return the sample rate expected by the model.

Return

Sample rate.

```
unsafe uint DeepSpeechClient.Interfaces.IDeepSpeech.GetModelBeamWidth()
```

Get beam width value used by the model. If SetModelBeamWidth was not called before, will return the default value loaded from the model file.

Return

Beam width value used by the model.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.SetModelBeamWidth(uint aBeamWidth)
```

Set beam width value used by the model.

Parameters

- `aBeamWidth`: The beam width used by the decoder. A larger beam width value generates better results at the cost of decoding time.

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.EnableExternalScorer(string aScorerPath)
```

Enable decoding using an external scorer.

Parameters

- `aScorerPath`: The path to the external scorer file.

Exceptions

- `ArgumentException`: Thrown when the native binary failed to enable decoding with an external scorer.

- `FileNotFoundException`: Thrown when cannot find the scorer file.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.AddHotWord(string aWord, float aBoost)
```

Add a hot-word.

Parameters

- `aWord`: Some word
- `aBoost`: Some boost

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.EraseHotWord(string aWord)
```

Erase entry for a hot-word.

Parameters

- `aWord`: Some word

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.ClearHotWords()
```

Clear all hot-words.

Exceptions

- `ArgumentException`: Thrown on failure.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.DisableExternalScorer()
```

Disable decoding using an external scorer.

Exceptions

- `ArgumentException`: Thrown when an external scorer is not enabled.

```
unsafe void DeepSpeechClient.Interfaces.IDeepSpeech.SetScorerAlphaBeta(float aAlpha, float aBeta)
```

Set hyperparameters alpha and beta of the external scorer.

Parameters

- `aAlpha`: The alpha hyperparameter of the decoder. Language model weight.
- `aBeta`: The beta hyperparameter of the decoder. Word insertion weight.

Exceptions

- `ArgumentException`: Thrown when an external scorer is not enabled.

```
unsafe string
DeepSpeechClient.Interfaces.IDeepSpeech.SpeechToText(short [] aBuffer,
uint aBufferSize)
```

Use the DeepSpeech model to perform Speech-To-Text.

Return

The STT result. Returns NULL on error.

Parameters

- `aBuffer`: A 16-bit, mono raw audio signal at the appropriate sample rate (matching what the model was trained on).
- `aBufferSize`: The number of samples in the audio signal.

```
unsafe Metadata
DeepSpeechClient.Interfaces.IDeepSpeech.SpeechToTextWithMetadata(short
[] aBuffer, uint aBufferSize, uint aNumResults)
```

Use the DeepSpeech model to perform Speech-To-Text, return results including metadata.

Return

The extended metadata. Returns NULL on error.

Parameters

- `aBuffer`: A 16-bit, mono raw audio signal at the appropriate sample rate (matching what the model was trained on).
- `aBufferSize`: The number of samples in the audio signal.
- `aNumResults`: Maximum number of candidate transcripts to return. Returned list might be smaller than this.

```
unsafe void
DeepSpeechClient.Interfaces.IDeepSpeech.FreeStream(DeepSpeechStream
stream)
```

Destroy a streaming state without decoding the computed logits. This can be used if you no longer need the result of an ongoing streaming inference and don't want to perform a costly decode operation.

```
unsafe DeepSpeechStream
DeepSpeechClient.Interfaces.IDeepSpeech.CreateStream()
```

Creates a new streaming inference state.

```
unsafe void
DeepSpeechClient.Interfaces.IDeepSpeech.FeedAudioContent(DeepSpeechStre
am stream, short [] aBuffer, uint aBufferSize)
```

Feeds audio samples to an ongoing streaming inference.

Parameters

- `stream`: Instance of the stream to feed the data.
- `aBuffer`: An array of 16-bit, mono raw audio samples at the appropriate sample rate (matching what the model was trained on).

```
unsafe string
DeepSpeechClient.Interfaces.IDeepSpeech.IntermediateDecode(DeepSpeechSt
ream stream)
```

Computes the intermediate decoding of an ongoing streaming inference.

Return

The STT intermediate result.

Parameters

- `stream`: Instance of the stream to decode.

```
unsafe Metadata
DeepSpeechClient.Interfaces.IDeepSpeech.IntermediateDecodeWithMetadata(
DeepSpeechStream stream, uint aNumResults)
```

Computes the intermediate decoding of an ongoing streaming inference, including metadata.

Return

The extended metadata result.

Parameters

- `stream`: Instance of the stream to decode.
- `aNumResults`: Maximum number of candidate transcripts to return. Returned list might be smaller than this.

```
unsafe string
DeepSpeechClient.Interfaces.IDeepSpeech.FinishStream(DeepSpeechStream
stream)
```

Closes the ongoing streaming inference, returns the STT result over the whole audio signal.

Return

The STT result.

> **Parameters**
>
> - `stream`: Instance of the stream to finish.

```
unsafe Metadata
DeepSpeechClient.Interfaces.IDeepSpeech.FinishStreamWithMetadata(DeepSp
eechStream stream, uint aNumResults)
```

Closes the ongoing streaming inference, returns the STT result over the whole audio signal, including metadata.

> **Return**
>
> The extended metadata result.

> **Parameters**
>
> - `stream`: Instance of the stream to finish.
> - `aNumResults`: Maximum number of candidate transcripts to return. Returned list might be smaller than this.

.NET API Usage example

Examples are from *native_client/dotnet/DeepSpeechConsole/Program.cs*.

Creating a model instance and loading model

```
57                using (IDeepSpeech sttClient = new DeepSpeech(model ??
58"output_graph.pbmm"))
                  {
```

Performing inference

```
 93                       if (extended)
 94                       {
 95                           Metadata metaResult =
 96sttClient.SpeechToTextWithMetadata(waveBuffer.ShortBuffer,
 97                                   Convert.ToUInt32(waveBuffer.MaxSize / 2), 1);
 98                           speechResult = MetadataToString(metaResult.Transcripts[0]);
 99                       }
100                       else
101                       {
102                           speechResult = sttClient.SpeechToText(waveBuffer.ShortBuffer,
103                               Convert.ToUInt32(waveBuffer.MaxSize / 2));
                          }
```