

# Table of Contents

## NPCEngine

NonPlayerCharacterEditor

NPCEngineConfig

NPCEngineConfigEditor

NPCEngineException

ResultFuture<ReturnType>

ServiceConfigDescriptor

## NPCEngine.API

Classification

Control

PersonaDialogue

SemanticQuery

ServiceMetadata

ServiceStatus

SpeechToText

TextGeneration<ContextType>

TextToSpeech

## NPCEngine.Components

AbstractDialogueSystem

AbstractSpeechToText

Character

ChatLine

ColliderLocationTrigger

DialogueUI

DictationRecognizerSTT

FantasyChatbotContext

FantasyChatbotTextGeneration

Location

NonPlayerCharacter

NPCEngineManager

NPCEngineSTT

PlayerCharacter

## NPCEngine.RPC

APICommunicatorHTTPImpl  
APICommunicatorZMQImpl  
Request  
RequestDispatcherImpl  
RPCBase  
RPCRequestMessage<ParametersType>  
RPCResponseError  
RPCResponseMessage<ReturnType>  
ServerType  
NPCEngine.Utility  
AudioSourceQueue  
CoroutineUtility  
Singleton<T>

# Namespace NPCEngine

## Classes

[NonPlayerCharacterEditor](#)

[NPCEngineConfig](#)

[NPCEngineConfigEditor](#)

[NPCEngineException](#)

General NPC Engine exception.

[ResultFuture<ReturnType>](#)

Future result of RPC call.

[ServiceConfigDescriptor](#)

# Class NonPlayerCharacterEditor

Inheritance

System.Object

NonPlayerCharacterEditor

Namespace: **NPCEngine**

Assembly: cs.temp.dll.dll

Syntax

```
public class NonPlayerCharacterEditor : Editor
```

Methods

OnInspectorGUI()

Declaration

```
public override void OnInspectorGUI()
```

# Class NPCEngineConfig

Inheritance

System.Object  
[Singleton<NPCEngineConfig>](#)  
NPCEngineConfig

Inherited Members

[Singleton<NPCEngineConfig>.Instance](#)  
[Singleton<NPCEngineConfig>.GetInstance\(\)](#)

Namespace: **NPC Engine**  
Assembly: cs.temp.dll.dll

Syntax

```
public class NPCEngineConfig : Singleton<NPCEngineConfig>
```

## Fields

### connectToExistingServer

Declaration

```
public bool connectToExistingServer
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

### debugLogs

Declaration

```
public bool debugLogs
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

### modelsPath

Declaration

```
public string modelsPath
```

Field Value

TYPE	DESCRIPTION
System.String	

### nChunksSpeechGeneration

Declaration

```
public int nChunksSpeechGeneration
```

Field Value

TYPE	DESCRIPTION
System.Int32	

npcEnginePath

Declaration

```
public string npcEnginePath
```

Field Value

TYPE	DESCRIPTION
System.String	

serverAddress

Declaration

```
public string serverAddress
```

Field Value

TYPE	DESCRIPTION
System.String	

serverConsole

Declaration

```
public bool serverConsole
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

serverType

Declaration

```
public ServerType serverType
```

Field Value

TYPE	DESCRIPTION
ServerType	

services

Declaration

```
public List<ServiceConfigDescriptor> services
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< <a href="#">ServiceConfigDescriptor</a> >	

temperature

Declaration

public float temperature
--------------------------

Field Value

TYPE	DESCRIPTION
System.Single	

topK

Declaration

public int topK
-----------------

Field Value

TYPE	DESCRIPTION
System.Int32	

Properties

Port

Declaration

public string Port { get; }
-----------------------------

Property Value

TYPE	DESCRIPTION
System.String	

Methods

RefreshServices()

Declaration

public void RefreshServices()
-------------------------------

ToBeStarted(String)

Declaration

public bool ToBeStarted(string name)
--------------------------------------

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	

Returns

TYPE	DESCRIPTION
System.Boolean	



# Class NPCEngineConfigEditor

## Inheritance

System.Object

NPCEngineConfigEditor

Namespace: [NPCEngine](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public class NPCEngineConfigEditor : Editor
```

## Methods

OnInspectorGUI()

## Declaration

```
public override void OnInspectorGUI()
```

# Class NPCEngineException

General NPC Engine exception.

Inheritance

System.Object  
System.Exception  
NPCEngineException

Namespace: [NPCEngine](#)  
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class NPCEngineException : Exception, ISerializable, _Exception
```

Constructors

NPCEngineException()

Declaration

```
public NPCEngineException()
```

NPCEngineException(SerializationInfo, StreamingContext)

Declaration

```
protected NPCEngineException(SerializationInfo info, StreamingContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Runtime.Serialization.SerializationInfo	info	
System.Runtime.Serialization.StreamingContext	context	

NPCEngineException(String)

Declaration

```
public NPCEngineException(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

NPCEngineException(String, Exception)

Declaration

```
public NPCEngineException(string message, Exception inner)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	

TYPE	NAME	DESCRIPTION
System.Exception	inner	

# Class ResultFuture<ReturnType>

Future result of RPC call.

Inheritance

System.Object

ResultFuture<ReturnType>

Namespace: [NPCEngine](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ResultFuture<ReturnType>
```

Type Parameters

NAME	DESCRIPTION
ReturnType	

## Properties

### Error

Declaration

```
public NPCEngineException Error { get; }
```

Property Value

TYPE	DESCRIPTION
<a href="#">NPCEngineException</a>	

### Result

Declaration

```
public ReturnType Result { get; }
```

Property Value

TYPE	DESCRIPTION
ReturnType	

### ResultReady

Declaration

```
public bool ResultReady { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

## Methods

ErrorCallback(NPCEngineException)

Declaration

```
public void errorCallback(NPCEngineException error)
```

Parameters

TYPE	NAME	DESCRIPTION
<a href="#">NPCEngineException</a>	error	

ResultFinishedCallback(ReturnType)

Declaration

```
public void ResultFinishedCallback(ReturnType result)
```

Parameters

TYPE	NAME	DESCRIPTION
ReturnType	result	

# Class ServiceConfigDescriptor

Inheritance

System.Object  
ServiceConfigDescriptor

Namespace: **NPCEngine**  
Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]  
public class ServiceConfigDescriptor
```

## Fields

### name

Declaration

```
public string name
```

Field Value

TYPE	DESCRIPTION
System.String	

### path

Declaration

```
public string path
```

Field Value

TYPE	DESCRIPTION
System.String	

### start

Declaration

```
public bool start
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

### type

Declaration

```
public string type
```

Field Value

TYPE	DESCRIPTION
System.String	

# Namespace NPCEngine.API

## Classes

### Classification

Classification services RPC Interface.

### Control

`Control1` provides RPC interface to control service.

### PersonaDialogue

`Chatbot` provides remote procedure calls to inference engine's TextGeneration services.

### SemanticQuery

Static class `SemanticQuery` provides remote procedure calls to inference engine's Semantic similarity model. Prefer predefining queries via `PredefineQuery` and then using `QueryPredefined` instead of directly using `Query`

### ServiceMetadata

Data class containing service metadata.

### SpeechToText

Static class `SpeechToText` provides remote procedure calls to inference engine's speech to text module.

### TextGeneration<ContextType>

`Chatbot` provides remote procedure calls to inference engine's TextGeneration services.

### TextToSpeech

Static class `TextToSpeech` provides remote procedure calls to inference engine's text to speech module.

## Enums

### ServiceStatus

Service status enum.



# Class Classification

Classification services RPC Interface.

Inheritance

System.Object

Classification

Namespace: **NPCEngine.API**

Assembly: cs.temp.dll.dll

Syntax

```
public class Classification : RPCBase
```

## Methods

Classify(String, List<String>, Action<List<List<Single>>>)

Declaration

```
public IEnumerator Classify(string query, List<string> context, Action<List<List<float>>> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	query	
System.Collections.Generic.List<System.String>	context	
System.Action<System.Collections.Generic.List<System.Collections.Generic.List<System.Single>>>	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

# Class Control

`Control` provides RPC interface to control service.

Inheritance

System.Object

Control

Namespace: `NPCEngine.API`

Assembly: `cs.temp.dll.dll`

Syntax

```
public class Control : RPCBase
```

## Methods

`GetServiceMetadata(String, Action<ServiceMetadata>)`

Get metadata for one service.

Declaration

```
public IEnumerable GetServiceMetadata(string service_id, Action<ServiceMetadata> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	Resolvable service name (i.e. id, type or API name)
System.Action< <a href="#">ServiceMetadata</a> >	outputCallback	Callback action to consume results.

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	<code>ServiceMetadata</code> for resolved service.

`GetServiceMetadataFuture(String)`

Declaration

```
public ResultFuture<ServiceMetadata> GetServiceMetadataFuture(string service_id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	

Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture</a> < <a href="#">ServiceMetadata</a> >	

GetServicesMetadata(Action<List<ServiceMetadata>>)

Get metadata for all the services.

Declaration

```
public IEnumerable GetServicesMetadata(Action<List<ServiceMetadata>> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Collections.Generic.List<ServiceMetadata>>	outputCallback	Callback action to consume results.

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	A list of ServiceMetadata for each service.

GetServicesMetadataFuture()

Declaration

```
public ResultFuture<List<ServiceMetadata>> GetServicesMetadataFuture()
```

Returns

TYPE	DESCRIPTION
ResultFuture<System.Collections.Generic.List<ServiceMetadata>>	

GetServiceStatus(String, Action<ServiceStatus>)

Get service status.

Declaration

```
public IEnumerable GetServiceStatus(string service_id, Action<ServiceStatus> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	
System.Action<ServiceStatus>	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

GetServiceStatusFuture(String)

Declaration

```
public ResultFuture<ServiceStatus> GetServiceStatusFuture(string service_id)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	

#### Returns

TYPE	DESCRIPTION
ResultFuture<ServiceStatus>	

### RestartService(String)

Restart service.

#### Declaration

```
public IEnumerable RestartService(string service_id)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	Resolvable service name (i.e. id, type or API name)

#### Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### StartService(String)

#### Declaration

```
public IEnumerable StartService(string service_id)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	

#### Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### StartServiceNoConfirm(String)

#### Declaration

```
public void StartServiceNoConfirm(string service_id)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	

### StopService(String)

Declaration

```
public IEnumerator StopService(string service_id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### StopServiceNoConfirm(String)

Declaration

```
public void StopServiceNoConfirm(string service_id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	service_id	

# Class PersonaDialogue

Chatbot provides remote procedure calls to inference engine's TextGeneration services.

Inheritance

System.Object

PersonaDialogue

Namespace: NPCEngine.API

Assembly: cs.temp.dll.dll

Syntax

```
public class PersonaDialogue : RPCBase
```

Methods

StartDialogue(String, String, String, String, String, String, Action<String>, String)

Declaration

```
public IEnumerable StartDialogue(string name1, string persona1, string name2, string persona2, string location_name, string location_description, Action<string> outputCallback, string dialogue_id = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name1	
System.String	persona1	
System.String	name2	
System.String	persona2	
System.String	location_name	
System.String	location_description	
System.Action<System.String>	outputCallback	
System.String	dialogue_id	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

StepDialogue(String, String, Boolean, Action<Tuple<String, Boolean>>, Single, List<String>, String)

Step dialogue.

If utterance is null, it will be generated. If scripted utterances are not null they will be compared to the utterance and replace it if similarity score is above scripted\_threshold (score is in range [0,1]). If update\_history is true, the dialogue history will be updated with the utterance.

Declaration

```
public IEnumerator StepDialogue(string dialogue_id, string speaker_id, bool update_history,
Action<Tuple<string, bool>> outputCallback, float scripted_threshold = 0.5F, List<string> scripted_utterances
= null, string utterance = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dialogue_id	
System.String	speaker_id	
System.Boolean	update_history	
System.Action<System.Tuple<System.String, System.Boolean>>	outputCallback	Callback that accepts results: Tuple with utterance and bool flag that is True if scripted utterance was used
System.Single	scripted_threshold	
System.Collections.Generic.List<System.String>	scripted_utterances	
System.String	utterance	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

# Class SemanticQuery

Static class `SemanticQuery` provides remote procedure calls to inference engine's Semantic similarity model. Prefer predefining queries via `PredefineQuery` and then using `QueryPredefined` instead of directly using `Query`

## Inheritance

System.Object  
SemanticQuery

Namespace: `NPCEngine.API`

Assembly: `cs.temp.dll.dll`

## Syntax

```
public class SemanticQuery : RPCBase
```

## Methods

### Cache(List<String>)

#### Declaration

```
public void Cache(List<string> queryIds)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	queryIds	

### Compare(String, List<String>, Action<List<Single>>)

#### Declaration

```
public IEnumerable Compare(string query, List<string> context, Action<List<float>> outputCallback)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	query	
System.Collections.Generic.List<System.String>	context	
System.Action<System.Collections.Generic.List<System.Single> >	outputCallback	

#### Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### CompareFuture(String, List<String>)

#### Declaration

```
public ResultFuture<List<float>> CompareFuture(string query, List<string> context)
```

#### Parameters



TYPE	NAME	DESCRIPTION
System.String	query	
System.Collections.Generic.List<System.String>	context	

Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture</a> <System.Collections.Generic.List<System.Single> >	

# Class ServiceMetadata

Data class containing service metadata.

### Inheritance

System.Object

ServiceMetadata

Namespace: **NPCEngine.API**

Assembly: cs.temp.dll.dll

### Syntax

```
[Serializable]
public class ServiceMetadata
```

### Fields

#### api\_name

Name of the API class.

### Declaration

```
public string api_name
```

### Field Value

TYPE	DESCRIPTION
System.String	

#### id

ID of the service.

### Declaration

```
public string id
```

### Field Value

TYPE	DESCRIPTION
System.String	

#### path

Path to the service's folder.

### Declaration

```
public string path
```

### Field Value

TYPE	DESCRIPTION
System.String	

#### readme

Model specific readme.

Declaration

```
public string readme
```

Field Value

TYPE	DESCRIPTION
System.String	

service

Name of the service class.

Declaration

```
public string service
```

Field Value

TYPE	DESCRIPTION
System.String	

service\_description

Long service class description

Declaration

```
public string service_description
```

Field Value

TYPE	DESCRIPTION
System.String	

service\_short\_description

Short service class description

Declaration

```
public string service_short_description
```

Field Value

TYPE	DESCRIPTION
System.String	

# Enum ServiceStatus

Service status enum.

Namespace: [NPCEngine.API](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public enum ServiceStatus
```

## Fields

NAME	DESCRIPTION
AWAITING	
ERROR	
RUNNING	
STARTING	
STOPPED	
TIMEOUT	
UNKNOWN	

# Class SpeechToText

Static class `SpeechToText` provides remote procedure calls to inference engine's speech to text module.

Inheritance

System.Object

SpeechToText

Namespace: `NPCEngine.API`

Assembly: `cs.temp.dll.dll`

Syntax

```
public class SpeechToText : RPCBase
```

## Methods

### GetDevices(Action<List<String>>)

Declaration

```
public IEnumerator GetDevices(Action<List<string>> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Collections.Generic.List<System.String>>	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### GetDevicesFuture()

Declaration

```
public ResultFuture<List<string>> GetDevicesFuture()
```

Returns

TYPE	DESCRIPTION
<code>ResultFuture</code> <System.Collections.Generic.List<System.String>>	

### InitializeMicrophoneInput()

Declaration

```
public IEnumerator InitializeMicrophoneInput()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### Listen(String, Action<String>)

Declaration

```
public IEnumerator Listen(string context, Action<string> outputCallback)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	context	
System.Action<System.String>	outputCallback	

#### Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### ListenFuture(String)

#### Declaration

```
public ResultFuture<string> ListenFuture(string context)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	context	

#### Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture</a> <System.String>	

### SetDevice(Int32, Action<List<String>>)

#### Declaration

```
public IEnumerator SetDevice(int deviceId, Action<List<string>> outputCallback)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Int32	deviceId	
System.Action<System.Collections.Generic.List<System.String>>	outputCallback	

#### Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### SetDeviceFuture(Int32)

#### Declaration

```
public ResultFuture<List<string>> SetDeviceFuture(int deviceId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	deviceId	

Returns

TYPE	DESCRIPTION
ResultFuture<System.Collections.Generic.List<System.String> >	

Transcribe(List<Single>, Action<String>)

Declaration

```
public IEnumerator Transcribe(List<float> audio, Action<string> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Single>	audio	
System.Action<System.String>	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

TranscribeFuture(List<Single>)

Declaration

```
public ResultFuture<string> TranscribeFuture(List<float> audio)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Single>	audio	

Returns

TYPE	DESCRIPTION
ResultFuture<System.String>	

# Class TextGeneration<ContextType>

Chatbot provides remote procedure calls to inference engine's TextGeneration services.

## Inheritance

System.Object  
TextGeneration<ContextType>  
[FantasyChatbotTextGeneration](#)

Namespace: [NPCEngine.API](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public class TextGeneration<ContextType> : RPCBase where ContextType : new()
```

## Type Parameters

NAME	DESCRIPTION
ContextType	

## Methods

GenerateReply(ContextType, Action<String>, Single, Int32)

## Declaration

```
public IEnumerable GenerateReply(ContextType context, Action<string> outputCallback, float temperature = 0.8F, int topk = 5)
```

## Parameters

TYPE	NAME	DESCRIPTION
ContextType	context	
System.Action<System.String>	outputCallback	
System.Single	temperature	
System.Int32	topk	

## Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

GenerateReplyFuture(ContextType, Single, Int32)

## Declaration

```
public ResultFuture<string> GenerateReplyFuture(ContextType context, float temperature = 0.8F, int topk = 5)
```

## Parameters

TYPE	NAME	DESCRIPTION
ContextType	context	



TYPE	NAME	DESCRIPTION
System.Single	temperature	
System.Int32	topk	

Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture</a> <System.String>	

GetPromptTemplate(Action<String>)

Declaration

```
public IEnumerator GetPromptTemplate(Action<string> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.String>	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

GetPromptTemplateFuture()

Declaration

```
public ResultFuture<string> GetPromptTemplateFuture()
```

Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture</a> <System.String>	

GetSpecialTokens(Action<Dictionary<String, String>>)

Declaration

```
public IEnumerator GetSpecialTokens(Action<Dictionary<string, string>> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Collections.Generic.Dictionary<System.String, System.String>>	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

# GetSpecialTokensFuture()

## Declaration

```
public ResultFuture<Dictionary<string, string>> GetSpecialTokensFuture()
```

## Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture</a> <System.Collections.Generic.Dictionary<System.String, System.String> >	

# Class TextToSpeech

Static class `TextToSpeech` provides remote procedure calls to inference engine's text to speech module.

Inheritance

System.Object

TextToSpeech

Namespace: `NPCEngine.API`

Assembly: `cs.temp.dll.dll`

Syntax

```
public class TextToSpeech : RPCBase
```

## Methods

### GetNextResult(Action<List<Single>>)

Declaration

```
public IEnumerator GetNextResult(Action<List<float>> outputCallback)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Collections.Generic.List<System.Single> >	outputCallback	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### GetNextResultFuture()

Declaration

```
public ResultFuture<List<float>> GetNextResultFuture()
```

Returns

TYPE	DESCRIPTION
<code>ResultFuture</code> <System.Collections.Generic.List<System.Single> >	

### GetSpeakerIdsFuture()

Declaration

```
public ResultFuture<List<string>> GetSpeakerIdsFuture()
```

Returns

TYPE	DESCRIPTION
<code>ResultFuture</code> <System.Collections.Generic.List<System.String> >	

### StartTTS(String, String, Int32, Action)

Declaration

```
public IEnumerator StartTTS(string voiceId, string line, int n_chunks, Action outputCallback)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	voiceId	
System.String	line	
System.Int32	n_chunks	
System.Action	outputCallback	

#### Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

### StartTTSFuture(String, String, Int32)

#### Declaration

```
public void StartTTSFuture(string speaker_id, string text, int n_chunks = 1)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	speaker_id	
System.String	text	
System.Int32	n_chunks	

# Namespace NPCEngine.Components

## Classes

### [AbstractDialogueSystem](#)

Abstract class for dialogue system integration. Concrete implementations are used by NonPlayerCharacter component for scripted replies.

### [AbstractSpeechToText](#)

Abstract class for speech to text system. Concrete implementations are used by PlayerCharacter component for receiving speech.

### [Character](#)

Character natural language descriptions.

### [ChatLine](#)

### [ColliderLocationTrigger](#)

Simple collider trigger that sets location in PlayerCharacter

Requires Trigger collider.

### [DialogueUI](#)

Simple UI script that uses NonPlayerCharacter callbacks.

### [DictationRecognizerSTT](#)

Speech to text component using Windows Speech Recognition DictationRecognizer as backend.

### [FantasyChatbotContext](#)

Text generation context for LIGHT dataset format (2 personas and location)

### [FantasyChatbotTextGeneration](#)

Specific MonoBehaviour to exclude templating from the class.

### [Location](#)

Location natural language description.

### [NonPlayerCharacter](#)

### [NPCEngineManager](#)

Manager class that handles services and server lifetime and status.

### [NPCEngineSTT](#)

### [PlayerCharacter](#)

MonoBehaviour that handles dialogue initiation and passing recognized speech to NPCs.

# Class AbstractDialogueSystem

Abstract class for dialogue system integration. Concrete implementations are used by NonPlayerCharacter component for scripted replies.

Inheritance

System.Object

AbstractDialogueSystem

Namespace: NPCEngine.Components

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class AbstractDialogueSystem : MonoBehaviour
```

## Methods

CurrentNodeIsPlayer()

Returns true if it's players turn in the dialogue.

Declaration

```
public abstract bool CurrentNodeIsPlayer()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

CurrentNodeNPCAudio()

Get the current node's audio for the NPC line (Never called for player). Return null if no audio is set or if the node is player.

Declaration

```
public abstract AudioClip CurrentNodeNPCAudio()
```

Returns

TYPE	DESCRIPTION
AudioClip	

CurrentNodeNPCLine()

If the current node is NPC, return the NPC's line. else return empty string.

Declaration

```
public abstract string CurrentNodeNPCLine()
```

Returns

TYPE	DESCRIPTION
System.String	

CurrentNodeThreshold()

Get the semantic similarity threshold for the current node in the dialogue. Return -1 if no threshold is set.

Declaration

```
public abstract float CurrentNodeThreshold()
```

Returns

TYPE	DESCRIPTION
System.Single	

EndDialog()

Will be called when dialogue ends.

Declaration

```
public abstract void EndDialog()
```

GetCurrentNodeOptions()

Get the lines associated with the current node in the dialogue.

Declaration

```
public abstract List<string> GetCurrentNodeOptions()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

GetCurrentNodeTopics()

Get the short abstract descriptions of the lines associated with the current node in the dialogue.

Declaration

```
public abstract List<string> GetCurrentNodeTopics()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<System.String>	

Next()

Move to the next node in the dialogue.

Declaration

```
public abstract void Next()
```

SelectOption(Int32)

Select a line from the current node in the dialogue.

Declaration

```
public abstract void SelectOption(int optionId)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Int32	optionId	

#### StartDialogue()

Will be called when dialogue starts.

#### Declaration

```
public abstract void StartDialogue()
```



# Class AbstractSpeechToText

Abstract class for speech to text system. Concrete implementations are used by PlayerCharacter component for receiving speech.

Inheritance

System.Object  
AbstractSpeechToText  
[DictationRecognizerSTT](#)  
[NPCEngineSTT](#)

Namespace: [NPCEngine.Components](#)  
Assembly: cs.temp.dll.dll

Syntax

```
public abstract class AbstractSpeechToText : MonoBehaviour
```

Properties

Context

Consumed context to be used in the next STT request

Declaration

```
public string Context { protected get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

SpeechRecognized(String)

Declaration

```
protected void SpeechRecognized(string result)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	result	

StartListening()

Called when system must start listening for speech.

Declaration

```
public abstract void StartListening()
```

StopListening()

Called when system should stop listening for speech.

Declaration

```
public abstract void StopListening()
```

## Events

### OnSpeechRecognitionFailed

Can be used to handle speech recogintion errors

#### Declaration

```
public event Action<string> OnSpeechRecognitionFailed
```

#### Event Type

TYPE	DESCRIPTION
System.Action<System.String>	

### OnSpeechRecognized

PlayerCharacter subscribes to this event to get the Speech-to-Text result

#### Declaration

```
public event Action<string> OnSpeechRecognized
```

#### Event Type

TYPE	DESCRIPTION
System.Action<System.String>	

# Class Character

Character natural language descriptions.

Inheritance

System.Object

Character

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Character : ScriptableObject
```

## Fields

### Name

Declaration

```
public string Name
```

Field Value

TYPE	DESCRIPTION
System.String	

### Persona

Character's persona natural language description.

Declaration

```
public string Persona
```

Field Value

TYPE	DESCRIPTION
System.String	

# Class ChatLine

Inheritance

System.Object

ChatLine

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class ChatLine
```

## Fields

line

Declaration

```
public string line
```

Field Value

TYPE	DESCRIPTION
System.String	

speaker

Declaration

```
public string speaker
```

Field Value

TYPE	DESCRIPTION
System.String	

# Class ColliderLocationTrigger

Simple collider trigger that sets location in PlayerCharacter

Requires Trigger collider.

Inheritance

System.Object

ColliderLocationTrigger

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class ColliderLocationTrigger : MonoBehaviour
```

## Fields

### location

Location to set on trigger enter.

Declaration

```
public Location location
```

Field Value

TYPE	DESCRIPTION
<a href="#">Location</a>	

### locationCollider

Trigger

Declaration

```
public Collider locationCollider
```

Field Value

TYPE	DESCRIPTION
Collider	

# Class DialogueUI

Simple UI script that uses NonPlayerCharacter callbacks.

Inheritance

System.Object

DialogueUI

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DialogueUI : MonoBehaviour
```

Fields

dialogueHistory

Text field for the dialogue.

Declaration

```
protected Text dialogueHistory
```

Field Value

TYPE	DESCRIPTION
Text	

dialogueTopics

Text field for the topics of the possible dialogue options.

Declaration

```
protected Text dialogueTopics
```

Field Value

TYPE	DESCRIPTION
Text	

highlightColor

Declaration

```
public Color highlightColor
```

Field Value

TYPE	DESCRIPTION
Color	

lineLimit

Line limit for the text field.

Declaration

```
public int lineLimit
```

Field Value

TYPE	DESCRIPTION
System.Int32	

## uiObject

GameObject that will be enabled, disabled and oriented towards the player.

Declaration

```
public GameObject uiObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

## usernameColor

Declaration

```
public Color usernameColor
```

Field Value

TYPE	DESCRIPTION
Color	

## Methods

### AddLine(ChatLine, Boolean)

Adds the given text to the dialogue history. Used with the OnDialogueLine callback.

Declaration

```
public void AddLine(ChatLine chatLine, bool highlight = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ChatLine	chatLine	
System.Boolean	highlight	

### Disable()

Disable Dialogue UI.

Declaration

```
public void Disable()
```

Enable()

Enable Dialogue UI.

Declaration

```
public void Enable()
```

UpdateTopics(List<String>)

Updates the dialogue topics. Used with NonPlayerCharacter.OnTopicHintsUpdate.

Declaration

```
public void UpdateTopics(List<string> topics)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.String>	topics	



# Class DictationRecognizerSTT

Speech to text component using Windows Speech Recognition DictationRecognizer as backend.

Inheritance

System.Object  
[AbstractSpeechToText](#)  
DictationRecognizerSTT

Inherited Members

[AbstractSpeechToText.Context](#)  
[AbstractSpeechToText.OnSpeechRecognized](#)  
[AbstractSpeechToText.OnSpeechRecognitionFailed](#)  
[AbstractSpeechToText.SpeechRecognized\(String\)](#)

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class DictationRecognizerSTT : AbstractSpeechToText
```

Fields

debugLogs

Show debug messages.

Declaration

```
public bool debugLogs
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

PolicyNotAccepted

Property that indicates that windows speech is not configured.

Declaration

```
public bool PolicyNotAccepted { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

RecognizerStarted

Property that indicates that recognizer is running.

Declaration

```
public bool RecognizerStarted { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

StartListening()

Declaration

```
public override void StartListening()
```

Overrides

[AbstractSpeechToText.StartListening\(\)](#)

StopListening()

Declaration

```
public override void StopListening()
```

Overrides

[AbstractSpeechToText.StopListening\(\)](#)

# Class FantasyChatbotContext

Text generation context for LIGHT dataset format (2 personas and location)

Inheritance

System.Object

FantasyChatbotContext

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class FantasyChatbotContext
```

Fields

history

Declaration

```
public List<ChatLine> history
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< <a href="#">ChatLine</a> >	

location

Declaration

```
public string location
```

Field Value

TYPE	DESCRIPTION
System.String	

location\_name

Declaration

```
public string location_name
```

Field Value

TYPE	DESCRIPTION
System.String	

name

Declaration

```
public string name
```

Field Value

TYPE	DESCRIPTION
System.String	

other\_name

Declaration

```
public string other_name
```

Field Value

TYPE	DESCRIPTION
System.String	

other\_persona

Declaration

```
public string other_persona
```

Field Value

TYPE	DESCRIPTION
System.String	

persona

Declaration

```
public string persona
```

Field Value

TYPE	DESCRIPTION
System.String	

# Class FantasyChatbotTextGeneration

Specific Monobehaviour to exclude templating from the class.

## Inheritance

System.Object

[TextGeneration](#)<[FantasyChatbotContext](#)>

FantasyChatbotTextGeneration

## Inherited Members

[TextGeneration](#)<[FantasyChatbotContext](#)>.GenerateReplyFuture([FantasyChatbotContext](#), Single, Int32)

[TextGeneration](#)<[FantasyChatbotContext](#)>.GetSpecialTokensFuture()

[TextGeneration](#)<[FantasyChatbotContext](#)>.GetPromptTemplateFuture()

[TextGeneration](#)<[FantasyChatbotContext](#)>.GenerateReply([FantasyChatbotContext](#), Action<String>, Single, Int32)

[TextGeneration](#)<[FantasyChatbotContext](#)>.GetSpecialTokens(Action<Dictionary<String, String>>)

[TextGeneration](#)<[FantasyChatbotContext](#)>.GetPromptTemplate(Action<String>)

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public class FantasyChatbotTextGeneration : TextGeneration<FantasyChatbotContext>
```

# Class Location

Location natural language description.

Inheritance

System.Object

Location

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Location : ScriptableObject
```

Fields

Description

Declaration

```
public string Description
```

Field Value

TYPE	DESCRIPTION
System.String	

Name

Declaration

```
public string Name
```

Field Value

TYPE	DESCRIPTION
System.String	

# Class NonPlayerCharacter

Inheritance

System.Object

NonPlayerCharacter

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NonPlayerCharacter : MonoBehaviour
```

## Fields

### audioSourceQueue

Audio source queue that plays generated speech on the fly one-by-one.

Declaration

```
public AudioSourceQueue audioSourceQueue
```

Field Value

TYPE	DESCRIPTION
<a href="#">AudioSourceQueue</a>	

### character

Characters assigned to this NPC.

Declaration

```
public Character character
```

Field Value

TYPE	DESCRIPTION
<a href="#">Character</a>	

### defaultThreshold

Default semantic similarity threshold for dialogue tree triggers.

Declaration

```
public float defaultThreshold
```

Field Value

TYPE	DESCRIPTION
System.Single	

### dialogueSystem

Dialogue System integration.

Declaration

```
public AbstractDialogueSystem dialogueSystem
```

Field Value

TYPE	DESCRIPTION
<a href="#">AbstractDialogueSystem</a>	

## history

Dialogue history.

Declaration

```
public List<ChatLine> history
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< <a href="#">ChatLine</a> >	

## OnDialogueEnd

Event called when dialogue ends

Declaration

```
public UnityEvent OnDialogueEnd
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

## OnDialogueLine

Event called when new line is added to history where parameters are (ChatLine newLine, bool scriptedLine)

Declaration

```
public UnityEvent<ChatLine, bool> OnDialogueLine
```

Field Value

TYPE	DESCRIPTION
UnityEvent< <a href="#">ChatLine</a> , System.Boolean>	

## OnDialogueStart

Dialogue start event (triggered when the NPC starts talking)

Declaration

```
public UnityEvent OnDialogueStart
```

Field Value



TYPE	DESCRIPTION
UnityEvent	

### OnProcessingEnd

Event called in the end of an input processing

Declaration

```
public UnityEvent OnProcessingEnd
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

### OnProcessingStart

Event called in the beginning of an input processing

Declaration

```
public UnityEvent OnProcessingStart
```

Field Value

TYPE	DESCRIPTION
UnityEvent	

### OnTopicHintsUpdate

Event called when new dialogue topic hints appear

Declaration

```
public UnityEvent<List<string>> OnTopicHintsUpdate
```

Field Value

TYPE	DESCRIPTION
UnityEvent<System.Collections.Generic.List<System.String>>	

### testLocation

Declaration

```
public Location testLocation
```

Field Value

TYPE	DESCRIPTION
Location	

### testOtherCharacter

Declaration

```
public Character testOtherCharacter
```

Field Value

TYPE	DESCRIPTION
Character	

voiceId

Voice ID of the character for TTS.

Declaration

```
public string voiceId
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

EndDialog()

End dialogue.

Declaration

```
public void EndDialog()
```

GenerateAndPlaySpeech(String)

Coroutine to generate speech from text.

Declaration

```
public IEnumerator GenerateAndPlaySpeech(string line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	line	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

GenerateReply(String, String, String)

Run Chatbot API and generate and play speech

Declaration

```
public IEnumerator GenerateReply(string otherName, string otherPersona, string line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	otherName	Other conversant name
System.String	otherPersona	Other conversant persona
System.String	line	Utterance to generate reply to.

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

HandleLine(String, String, String)

Step dialogue by line from player.

Declaration

```
public void HandleLine(string otherName, string otherPersona, string line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	otherName	
System.String	otherPersona	
System.String	line	

HandleLineCoroutine(String, String, String)

Line handling coroutine

Declaration

```
public IEnumerator HandleLineCoroutine(string otherName, string otherPersona, string line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	otherName	
System.String	otherPersona	
System.String	line	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

HandlePlayerLineCoroutine(String, String, String)

Player line handling coroutine

Declaration

```
public IEnumerator HandlePlayerLineCoroutine(string otherName, string otherPersona, string line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	otherName	
System.String	otherPersona	
System.String	line	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

SayNPCLines()

Scan dialogue tree and say all scripted NPC lines

Declaration

```
public IEnumerator SayNPCLines()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

StartDialogue()

Initialize dialogue.

Declaration

```
public void StartDialogue()
```

# Class NPCEngineManager

Manager class that handles services and server lifetime and status.

Inheritance

System.Object  
[Singleton<NPCEngineManager>](#)  
NPCEngineManager

Inherited Members

[Singleton<NPCEngineManager>.Instance](#)  
[Singleton<NPCEngineManager>.GetInstance\(\)](#)

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class NPCEngineManager : Singleton<NPCEngineManager>
```

## Properties

### InferenceEngineProcessId

Declaration

```
protected int InferenceEngineProcessId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

### InferenceEngineRunning

Declaration

```
public bool InferenceEngineRunning { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

## Services

Declaration

```
public List<ServiceMetadata> Services { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< <a href="#">ServiceMetadata</a> >	

## ServicesById

Declaration

```
public Dictionary<string, ServiceMetadata> ServicesById { get; }
```

#### Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, <a href="#">ServiceMetadata</a> >	

## ServiceStatuses

#### Declaration

```
public List<ServiceStatus> ServiceStatuses { get; }
```

#### Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< <a href="#">ServiceStatus</a> >	

## ServiceStatusesById

#### Declaration

```
public Dictionary<string, ServiceStatus> ServiceStatusesById { get; }
```

#### Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, <a href="#">ServiceStatus</a> >	

## Methods

### DownloadModel(String)

Download model by ID

#### Declaration

```
public void DownloadModel(string id)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	id	

### GetAPI<T>()

Get or construct API for the given type.

#### Declaration

```
public T GetAPI<T>()  
    where T : RPCBase
```

#### Returns

TYPE	DESCRIPTION
T	

#### Type Parameters

NAME	DESCRIPTION
T	API type

### GetAPI<T>(String)

Get or construct API for the given type and service ID.

#### Declaration

```
public T GetAPI<T>(string id)
    where T : RPCBase
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	id	Service ID or otherwise resolvable name

#### Returns

TYPE	DESCRIPTION
T	

#### Type Parameters

NAME	DESCRIPTION
T	API Type

### StartInferenceEngine()

Starts the inference engine server and managing coroutines.

#### Declaration

```
public void StartInferenceEngine()
```

### StopInferenceEngine()

Stop the inference engine server, services and dispose resources.

#### Declaration

```
public void StopInferenceEngine()
```

### UpdateServices()

Coroutine that updates service list.

Declaration

```
public IEnumerator UpdateServices()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

UpdateServiceStatuses()

Coroutine that updates service statuses.

Declaration

```
public IEnumerator UpdateServiceStatuses()
```

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	



# Class NPCEngineSTT

## Inheritance

System.Object

[AbstractSpeechToText](#)

NPCEngineSTT

## Inherited Members

[AbstractSpeechToText.Context](#)

[AbstractSpeechToText.OnSpeechRecognized](#)

[AbstractSpeechToText.OnSpeechRecognitionFailed](#)

[AbstractSpeechToText.SpeechRecognized\(String\)](#)

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

## Syntax

```
public class NPCEngineSTT : AbstractSpeechToText
```

## Methods

### CallSpeechToText()

#### Declaration

```
public void CallSpeechToText()
```

### StartListening()

#### Declaration

```
public override void StartListening()
```

#### Overrides

[AbstractSpeechToText.StartListening\(\)](#)

### StopListening()

#### Declaration

```
public override void StopListening()
```

#### Overrides

[AbstractSpeechToText.StopListening\(\)](#)

# Class PlayerCharacter

MonoBehaviour that handles dialogue initiation and passing recognized speech to NPCs.

Inheritance

System.Object  
[Singleton<PlayerCharacter>](#)  
PlayerCharacter

Inherited Members

[Singleton<PlayerCharacter>.Instance](#)  
[Singleton<PlayerCharacter>.GetInstance\(\)](#)

Namespace: [NPCEngine.Components](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class PlayerCharacter : Singleton<PlayerCharacter>
```

Fields

character

Character description of the player.

Declaration

```
public Character character
```

Field Value

TYPE	DESCRIPTION
<a href="#">Character</a>	

CheckCamera

Declaration

```
public Camera CheckCamera
```

Field Value

TYPE	DESCRIPTION
Camera	

currentLocation

Location description of the player.

Declaration

```
public Location currentLocation
```

Field Value

TYPE	DESCRIPTION
<a href="#">Location</a>	

## HorizontalMargin

Declaration

```
public float HorizontalMargin
```

Field Value

TYPE	DESCRIPTION
System.Single	

## MaxRange

Declaration

```
public float MaxRange
```

Field Value

TYPE	DESCRIPTION
System.Single	

## SpeechToText

Declaration

```
public AbstractSpeechToText SpeechToText
```

Field Value

TYPE	DESCRIPTION
<a href="#">AbstractSpeechToText</a>	

## VerticalMargin

Declaration

```
public float VerticalMargin
```

Field Value

TYPE	DESCRIPTION
System.Single	

## Methods

### CheckIsSeen(Vector3)

Check if NPC is seen.

Declaration

```
public bool CheckIsSeen(Vector3 dialogAgentPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
Vector3	dialogAgentPosition	

Returns

TYPE	DESCRIPTION
System.Boolean	

### DeregisterDialogueCandidate(NonPlayerCharacter)

Remove NPC from the list of potential conversants.

Declaration

```
public void DeregisterDialogueCandidate(NonPlayerCharacter agent)
```

Parameters

TYPE	NAME	DESCRIPTION
NonPlayerCharacter	agent	

### IsRegistered(NonPlayerCharacter)

Check if NPC is registered as a dialogue candidate.

Declaration

```
public bool IsRegistered(NonPlayerCharacter agent)
```

Parameters

TYPE	NAME	DESCRIPTION
NonPlayerCharacter	agent	

Returns

TYPE	DESCRIPTION
System.Boolean	

### LeaveDialog()

Leave current dialogue.

Declaration

```
public void LeaveDialog()
```

### RegisterDialogueCandidate(NonPlayerCharacter)

Adds NPC to the list of potential conversants.

Declaration

```
public void RegisterDialogueCandidate(NonPlayerCharacter agent)
```

Parameters

TYPE	NAME	DESCRIPTION
<a href="#">NonPlayerCharacter</a>	agent	

# Namespace NPCEngine.RPC

## Classes

### [APICommunicatorHTTPImpl](#)

RPC loop implementation for HTTP protocol.

### [APICommunicatorZMQImpl](#)

RPC loop implementation for ZMQ protocol.

### [Request](#)

### [RequestDispatcherImpl](#)

Abstract class for RPC loop.

### [RPCBase](#)

### [RPCRequestMessage<ParametersType>](#)

JSON RPC request message.

### [RPCResponseError](#)

RPC response error.

### [RPCResponseMessage<Return Type>](#)

JSON RPC response message.

## Enums

### [ServerType](#)

Transport layer for RPC enum.

# Class APICommunicatorHTTPImpl

RPC loop implementation for HTTP protocol.

Inheritance

System.Object  
[RequestDispatcherImpl](#)  
APICommunicatorHTTPImpl

Inherited Members

[RequestDispatcherImpl.id](#)  
[RequestDispatcherImpl.address](#)

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class APICommunicatorHTTPImpl : RequestDispatcherImpl
```

Constructors

APICommunicatorHTTPImpl(String, String)

Declaration

```
public APICommunicatorHTTPImpl(string address, string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	address	
System.String	id	

Methods

DispatchRequestsCoroutine(Queue<Request>)

Declaration

```
public override IEnumerator DispatchRequestsCoroutine(Queue<Request> taskQueue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Queue< <a href="#">Request</a> >	taskQueue	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Overrides

[RequestDispatcherImpl.DispatchRequestsCoroutine\(Queue<Request>\)](#)

# Class APICommunicatorZMQImpl

RPC loop implementation for ZMQ protocol.

Inheritance

System.Object  
[RequestDispatcherImpl](#)  
APICommunicatorZMQImpl

Inherited Members

[RequestDispatcherImpl.id](#)  
[RequestDispatcherImpl.address](#)

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class APICommunicatorZMQImpl : RequestDispatcherImpl
```

Constructors

APICommunicatorZMQImpl(String, String)

Declaration

```
public APICommunicatorZMQImpl(string address, string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	address	
System.String	id	

Methods

DispatchRequestsCoroutine(Queue<Request>)

Declaration

```
public override IEnumerable DispatchRequestsCoroutine(Queue<Request> taskQueue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Queue< <a href="#">Request</a> >	taskQueue	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

Overrides

[RequestDispatcherImpl.DispatchRequestsCoroutine\(Queue<Request>\)](#)

Finalize()

Declaration



```
protected void Finalize()
```

# Class Request

Inheritance

System.Object

System.Tuple<System.String, System.Action<System.String> >

Request

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Request : Tuple<string, Action<string>>, IStructuralEquatable, IStructuralComparable, IComparable, ITuple
```

Constructors

Request(String, Action<String>)

Declaration

```
public Request(string key, Action<string> action)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	
System.Action<System.String>	action	

# Class RequestDispatcherImpl

Abstract class for RPC loop.

Inheritance

System.Object  
RequestDispatcherImpl  
[APICommunicatorHTTPImpl](#)  
[APICommunicatorZMQImpl](#)

Namespace: [NPCEngine.RPC](#)  
Assembly: cs.temp.dll.dll

Syntax

```
public abstract class RequestDispatcherImpl
```

## Constructors

RequestDispatcherImpl(String, String)

Declaration

```
public RequestDispatcherImpl(string address, string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	address	
System.String	id	

## Fields

address

Declaration

```
protected string address
```

Field Value

TYPE	DESCRIPTION
System.String	

id

Declaration

```
protected string id
```

Field Value

TYPE	DESCRIPTION
System.String	

## Methods

DispatchRequestsCoroutine(Queue<Request>)

Declaration

```
public abstract IEnumerator DispatchRequestsCoroutine(Queue<Request> taskQueue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Queue<Request>	taskQueue	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	

# Class RPCBase

Inheritance

System.Object

RPCBase

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class RPCBase : MonoBehaviour
```

## Fields

serviceId

Declaration

```
public string serviceId
```

Field Value

TYPE	DESCRIPTION
System.String	

## Methods

Run<P, R>(String, P)

Declaration

```
public ResultFuture<R> Run<P, R>(string methodName, P parameters)
    where P : new()
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	methodName	
P	parameters	

Returns

TYPE	DESCRIPTION
<a href="#">ResultFuture&lt;R&gt;</a>	

Type Parameters

NAME	DESCRIPTION
P	
R	

# Class RPCRequestMessage<ParametersType>

JSON RPC request message.

Inheritance

System.Object

RPCRequestMessage<ParametersType>

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class RPCRequestMessage<ParametersType>
    where ParametersType : new()
```

Type Parameters

NAME	DESCRIPTION
ParametersType	

Constructors

RPCRequestMessage()

Declaration

```
public RPCRequestMessage()
```

Fields

id

Declaration

```
public int id
```

Field Value

TYPE	DESCRIPTION
System.Int32	

jsonrpc

Declaration

```
public string jsonrpc
```

Field Value

TYPE	DESCRIPTION
System.String	

method

Declaration

```
public string method
```

Field Value

TYPE	DESCRIPTION
System.String	

parameters

Declaration

```
public ParametersType parameters
```

Field Value

TYPE	DESCRIPTION
ParametersType	

# Class RPCResponseError

RPC response error.

Inheritance

System.Object

RPCResponseError

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class RPCResponseError
```

Fields

code

Declaration

```
public int code
```

Field Value

TYPE	DESCRIPTION
System.Int32	

message

Declaration

```
public string message
```

Field Value

TYPE	DESCRIPTION
System.String	



# Class RPCResponseMessage<ReturnType>

JSON RPC response message.

Inheritance

System.Object

RPCResponseMessage<ReturnType>

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
[Serializable]
public class RPCResponseMessage<ReturnType>
```

Type Parameters

NAME	DESCRIPTION
ReturnType	

Constructors

RPCResponseMessage()

Declaration

```
public RPCResponseMessage()
```

Fields

error

Declaration

```
public RPCResponseError error
```

Field Value

TYPE	DESCRIPTION
<a href="#">RPCResponseError</a>	

id

Declaration

```
public int id
```

Field Value

TYPE	DESCRIPTION
System.Int32	

jsonrpc

Declaration

```
public string jsonrpc
```

Field Value

TYPE	DESCRIPTION
System.String	

result

Declaration

```
public ReturnType result
```

Field Value

TYPE	DESCRIPTION
ReturnType	

# Enum ServerType

Transport layer for RPC enum.

Namespace: [NPCEngine.RPC](#)

Assembly: cs.temp.dll.dll

Syntax

```
public enum ServerType
```

## Fields

NAME	DESCRIPTION
HTTP	
ZMQ	

# Namespace NPCEngine.Utility

## Classes

### [AudioSourceQueue](#)

Audio source queue that plays all clips from the queue one-by-one seamlessly.

### [CoroutineUtility](#)

Utility that allows to run coroutines in a coherent way both in editor and in player modes.

### [Singleton<T>](#)

Singleton `MonoBehaviour` base class.

# Class AudioSourceQueue

Audio source queue that plays all clips from the queue one-by-one seamlessly.

Inheritance

System.Object

AudioSourceQueue

Namespace: [NPCEngine.Utility](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class AudioSourceQueue : MonoBehaviour
```

## Fields

audioSource

Declaration

```
public AudioSource audioSource
```

Field Value

TYPE	DESCRIPTION
AudioSource	

## Methods

PlaySound(AudioClip)

Declaration

```
public void PlaySound(AudioClip clip)
```

Parameters

TYPE	NAME	DESCRIPTION
AudioClip	clip	

# Class CoroutineUtility

Utility that allows to run coroutines in a coherent way both in editor and in player modes.

Inheritance

System.Object  
Singleton<CoroutineUtility>  
CoroutineUtility

Inherited Members

Singleton<CoroutineUtility>.Instance  
Singleton<CoroutineUtility>.GetInstance()

Namespace: NPCEngine.Utility

Assembly: cs.temp.dll.dll

Syntax

```
public class CoroutineUtility : Singleton<CoroutineUtility>
```

Methods

GetGameObjectPath(MonoBehaviour)

Declaration

```
public static string GetGameObjectPath(MonoBehaviour behaviour)
```

Parameters

TYPE	NAME	DESCRIPTION
MonoBehaviour	behaviour	

Returns

TYPE	DESCRIPTION
System.String	

IsRunning(MonoBehaviour, String)

Declaration

```
public static bool IsRunning(MonoBehaviour owner, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
MonoBehaviour	owner	
System.String	name	

Returns

TYPE	DESCRIPTION
System.Boolean	

StartCoroutine(IEnumerable, MonoBehaviour, String)

Declaration

```
public static void StartCoroutine(IEnumerable routine, MonoBehaviour owner, string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.IEnumerator	routine	
MonoBehaviour	owner	
System.String	id	

StopAllEditorCoroutines()

Declaration

```
public static void StopAllEditorCoroutines()
```

StopAllEditorCoroutines(MonoBehaviour)

Declaration

```
public static void StopAllEditorCoroutines(MonoBehaviour owner)
```

Parameters

TYPE	NAME	DESCRIPTION
MonoBehaviour	owner	

StopAllPlayingCoroutines(MonoBehaviour)

Declaration

```
public static void StopAllPlayingCoroutines(MonoBehaviour owner)
```

Parameters

TYPE	NAME	DESCRIPTION
MonoBehaviour	owner	

StopCoroutine(String, MonoBehaviour)

Declaration

```
public static void StopCoroutine(string id, MonoBehaviour owner)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	
MonoBehaviour	owner	

WaitForSeconds(Single)

Declaration

```
public static IEnumerable WaitForSeconds(float seconds)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	seconds	

Returns

TYPE	DESCRIPTION
System.Collections.IEnumerator	



# Class Singleton<T>

Singleton `MonoBehaviour` base class.

Inheritance

System.Object

Singleton<T>

[NPCEngineManager](#)

[PlayerCharacter](#)

[NPCEngineConfig](#)

[CoroutineUtility](#)

Namespace: [NPCEngine.Utility](#)

Assembly: cs.temp.dll.dll

Syntax

```
public class Singleton<T> : MonoBehaviour where T : MonoBehaviour
```

Type Parameters

NAME	DESCRIPTION
T	

Properties

Instance

Declaration

```
public static T Instance { get; }
```

Property Value

TYPE	DESCRIPTION
T	

Methods

GetInstance()

Declaration

```
public static T GetInstance()
```

Returns

TYPE	DESCRIPTION
T	