

# Native Plugins Api

## 1.04

Generated by Doxygen 1.8.9.1

Mon Jul 6 2015 19:01:09



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List	3
<b>3</b>	<b>Namespace Documentation</b>	<b>5</b>
3.1	Package VoxelBusters	5
3.2	Package VoxelBusters.NativePlugins	5
3.2.1	Enumeration Type Documentation	7
3.2.1.1	eABAuthorizationStatus	7
3.2.1.2	eBillingTransactionState	7
3.2.1.3	eBillingTransactionVerificationState	8
3.2.1.4	eImageSource	8
3.2.1.5	ePickImageFinishReason	8
3.2.1.6	ePickVideoFinishReason	8
3.2.1.7	ePlayVideoFinishReason	9
3.2.1.8	NotificationType	9
3.2.1.9	eShareResult	9
3.2.1.10	eShareOptions	9
3.2.1.11	eTwitterComposerResult	9
3.2.1.12	eWebviewControllerType	10
3.3	Package VoxelBusters.NativePlugins.Internal	10
<b>4</b>	<b>Class Documentation</b>	<b>11</b>
4.1	AddressBook	11
4.1.1	Detailed Description	11
4.1.2	Member Function Documentation	11
4.1.2.1	GetAuthorizationStatus	11
4.1.2.2	ReadContacts	11
4.1.2.3	ReadContactsCompletion	12
4.2	AddressBookContact	12
4.2.1	Detailed Description	12

4.2.2	Constructor & Destructor Documentation	13
4.2.2.1	AddressBookContact	13
4.2.2.2	AddressBookContact	13
4.2.3	Member Function Documentation	13
4.2.3.1	GetImageAsync	13
4.2.3.2	ToString	13
4.2.4	Property Documentation	13
4.2.4.1	FirstName	13
4.2.4.2	LastName	13
4.2.4.3	ImagePath	13
4.2.4.4	PhoneNumberList	13
4.2.4.5	EmailIDList	13
4.3	MediaLibrarySettings.AndroidSettings	14
4.3.1	Detailed Description	14
4.3.2	Property Documentation	14
4.3.2.1	YoutubeAPIKey	14
4.4	BillingSettings.AndroidSettings	14
4.4.1	Detailed Description	14
4.5	NetworkConnectivitySettings.AndroidSettings	14
4.5.1	Detailed Description	14
4.5.2	Property Documentation	15
4.5.2.1	TimeOutPeriod	15
4.5.2.2	MaxRetryCount	15
4.5.2.3	TimeGapBetweenPolling	15
4.6	NotificationServiceSettings.AndroidSettings	15
4.6.1	Detailed Description	15
4.6.2	Property Documentation	15
4.6.2.1	SenderIDList	15
4.6.2.2	AllowCustomIcon	16
4.6.2.3	TickerTextKey	16
4.6.2.4	ContentTitleKey	16
4.6.2.5	ContentTextKey	16
4.6.2.6	UserInfoKey	16
4.6.2.7	TagKey	16
4.7	Billing	16
4.7.1	Detailed Description	17
4.7.2	Member Function Documentation	17
4.7.2.1	RequestForBillingProducts	17
4.7.2.2	IsProductPurchased	17
4.7.2.3	BuyProduct	18

4.7.2.4	RestoreCompletedTransactions	18
4.7.2.5	CustomVerificationFinished	18
4.7.2.6	BillingProductsRequestCompletion	18
4.7.2.7	TransactionCompletion	18
4.7.3	Event Documentation	19
4.7.3.1	BillingProductsRequestFinishedEvent	19
4.7.3.2	TransactionFinishedEvent	19
4.8	BillingProduct	19
4.8.1	Detailed Description	19
4.8.2	Member Function Documentation	19
4.8.2.1	Copy	19
4.8.2.2	ToString	20
4.8.3	Property Documentation	20
4.8.3.1	Name	20
4.8.3.2	Description	20
4.8.3.3	IsConsumable	20
4.8.3.4	Price	20
4.8.3.5	LocalizedPrice	20
4.8.3.6	ProductIdentifier	20
4.9	BillingSettings	20
4.9.1	Detailed Description	21
4.9.2	Property Documentation	21
4.9.2.1	Products	21
4.9.2.2	iOS	21
4.9.2.3	Android	21
4.10	BillingTransaction	21
4.10.1	Detailed Description	22
4.10.2	Property Documentation	22
4.10.2.1	ProductIdentifier	22
4.10.2.2	TransactionDateUTC	22
4.10.2.3	TransactionDateLocal	22
4.10.2.4	TransactionIdentifier	22
4.10.2.5	TransactionReceipt	22
4.10.2.6	TransactionState	23
4.10.2.7	VerificationState	23
4.10.2.8	Error	23
4.10.2.9	RawPurchaseData	23
4.11	CrossPlatformNotification	23
4.11.1	Detailed Description	23
4.11.2	Member Function Documentation	24

4.11.2.1	ToString	24
4.11.3	Property Documentation	24
4.11.3.1	AlertBody	24
4.11.3.2	FireDate	24
4.11.3.3	UserInfo	24
4.11.3.4	iOSProperties	24
4.11.3.5	AndroidProperties	24
4.12	NetworkConnectivitySettings.EditorSettings	24
4.12.1	Detailed Description	25
4.12.2	Property Documentation	25
4.12.2.1	TimeOutPeriod	25
4.12.2.2	MaxRetryCount	25
4.12.2.3	TimeGapBetweenPolling	25
4.13	BillingSettings.iOSSettings	25
4.13.1	Detailed Description	25
4.13.2	Property Documentation	25
4.13.2.1	SupportsReceiptValidation	25
4.13.2.2	ValidateUsingAppleServer	26
4.13.2.3	ValidateUsingServerURL	26
4.14	NotificationServiceSettings.iOSSettings	26
4.14.1	Detailed Description	26
4.14.2	Property Documentation	26
4.14.2.1	UserInfoKey	26
4.15	MediaLibrary	26
4.15.1	Detailed Description	27
4.15.2	Member Function Documentation	27
4.15.2.1	IsCameraSupported	27
4.15.2.2	PickImage	27
4.15.2.3	SaveScreenshotToGallery	28
4.15.2.4	SavelImageToGallery	28
4.15.2.5	SavelImageToGallery	28
4.15.2.6	SavelImageToGallery	28
4.15.2.7	PlayYoutubeVideo	28
4.15.2.8	PlayEmbeddedVideo	29
4.15.2.9	PlayVideoFromURL	29
4.15.2.10	PlayVideoFromGallery	29
4.15.2.11	PickImageCompletion	29
4.15.2.12	SavelImageToGalleryCompletion	29
4.15.2.13	PickVideoCompletion	29
4.15.2.14	PlayVideoCompletion	30

4.16	MediaLibrarySettings	30
4.16.1	Detailed Description	30
4.16.2	Property Documentation	30
4.16.2.1	Android	30
4.17	NetworkConnectivity	30
4.17.1	Detailed Description	31
4.17.2	Member Function Documentation	31
4.17.2.1	Initialise	31
4.17.2.2	NetworkConnectivityChanged	31
4.17.3	Property Documentation	31
4.17.3.1	IsConnected	31
4.17.4	Event Documentation	31
4.17.4.1	NetworkConnectivityChangedEvent	31
4.18	NetworkConnectivitySettings	31
4.18.1	Detailed Description	32
4.18.2	Property Documentation	32
4.18.2.1	IPAddress	32
4.18.2.2	Editor	32
4.18.2.3	Android	32
4.19	NotificationService	32
4.19.1	Detailed Description	33
4.19.2	Member Function Documentation	33
4.19.2.1	RegisterNotificationTypes	33
4.19.2.2	ScheduleLocalNotification	33
4.19.2.3	CancelLocalNotification	33
4.19.2.4	CancelAllLocalNotification	34
4.19.2.5	ClearNotifications	34
4.19.2.6	RegisterForRemoteNotifications	34
4.19.2.7	UnregisterForRemoteNotifications	34
4.19.2.8	RegisterForRemoteNotificationCompletion	34
4.19.2.9	ReceivedNotificationResponse	34
4.19.3	Event Documentation	34
4.19.3.1	DidFinishRegisterForRemoteNotificationEvent	34
4.19.3.2	DidReceiveLocalNotificationEvent	34
4.19.3.3	DidReceiveRemoteNotificationEvent	35
4.20	NotificationServiceSettings	35
4.20.1	Detailed Description	35
4.20.2	Property Documentation	35
4.20.2.1	iOS	35
4.20.2.2	Android	35

4.21	Sharing	35
4.21.1	Detailed Description	37
4.21.2	Member Function Documentation	37
4.21.2.1	ShareTextMessageOnSocialNetwork	37
4.21.2.2	ShareURLOnSocialNetwork	37
4.21.2.3	ShareScreenShotOnSocialNetwork	37
4.21.2.4	ShareImageOnSocialNetwork	38
4.21.2.5	ShareImageOnSocialNetwork	38
4.21.2.6	ShareMessage	38
4.21.2.7	ShareURL	38
4.21.2.8	ShareScreenShot	38
4.21.2.9	ShareImage	39
4.21.2.10	ShareImageAtPath	39
4.21.2.11	Share	39
4.21.2.12	SharingCompletion	39
4.21.2.13	IsMailServiceAvailable	40
4.21.2.14	SendPlainTextMail	40
4.21.2.15	SendHTMLTextMail	40
4.21.2.16	SendMailWithScreenshot	40
4.21.2.17	SendMailWithTexture	40
4.21.2.18	SendMailWithAttachment	41
4.21.2.19	SendMail	41
4.21.2.20	IsMessagingServiceAvailable	41
4.21.2.21	SendTextMessage	42
4.21.2.22	IsWhatsAppServiceAvailable	43
4.21.2.23	ShareTextMessageOnWhatsApp	43
4.21.2.24	ShareScreenshotOnWhatsApp	43
4.21.2.25	ShareImageOnWhatsApp	43
4.21.2.26	ShareImageOnWhatsApp	43
4.21.2.27	ShareImageOnWhatsApp	43
4.22	Twitter	44
4.22.1	Detailed Description	45
4.22.2	Member Function Documentation	45
4.22.2.1	Initialise	45
4.22.2.2	Login	45
4.22.2.3	Logout	45
4.22.2.4	IsLoggedIn	45
4.22.2.5	GetAuthToken	45
4.22.2.6	GetAuthTokenSecret	46
4.22.2.7	GetUserID	46



4.22.2.8	GetUserName	46
4.22.2.9	ShowTweetComposerWithMessage	46
4.22.2.10	ShowTweetComposerWithScreenshot	46
4.22.2.11	ShowTweetComposerWithImage	46
4.22.2.12	ShowTweetComposerWithLink	47
4.22.2.13	ShowTweetComposer	47
4.22.2.14	RequestAccountDetails	47
4.22.2.15	RequestEmailAccess	47
4.22.2.16	GetURLRequest	47
4.22.2.17	PostURLRequest	48
4.22.2.18	PutURLRequest	49
4.22.2.19	DeleteURLRequest	49
4.22.2.20	TWTRLoginCompletion	49
4.22.2.21	TWTRTweetCompletion	49
4.22.2.22	TWTRAccountDetailsCompletion	49
4.22.2.23	TWTREmailAccessCompletion	49
4.22.2.24	TWTRResonse	50
4.23	TwitterSession	50
4.23.1	Detailed Description	50
4.23.2	Member Function Documentation	50
4.23.2.1	ToString	50
4.23.3	Property Documentation	50
4.23.3.1	AuthToken	50
4.23.3.2	AuthTokenSecret	51
4.23.3.3	UserName	51
4.23.3.4	UserID	51
4.24	TwitterSettings	51
4.24.1	Detailed Description	51
4.24.2	Property Documentation	51
4.24.2.1	ConsumerKey	51
4.24.2.2	ConsumerSecret	51
4.25	TwitterUser	52
4.25.1	Detailed Description	52
4.25.2	Member Function Documentation	52
4.25.2.1	ToString	52
4.25.3	Property Documentation	52
4.25.3.1	UserID	52
4.25.3.2	Name	52
4.25.3.3	IsVerified	52
4.25.3.4	IsProtected	53

4.25.3.5	ProfileImageUrl	53
4.26	UI	53
4.26.1	Detailed Description	53
4.26.2	Member Function Documentation	54
4.26.2.1	ShowAlertDialogWithSingleButton	54
4.26.2.2	ShowAlertDialogWithMultipleButtons	54
4.26.2.3	AlertDialogCompletion	54
4.26.2.4	SetPopoverPoint	54
4.26.2.5	SetPopoverPointAtLastTouchPosition	54
4.26.2.6	ShowSingleFieldPromptDialogWithPlainText	54
4.26.2.7	ShowSingleFieldPromptDialogWithSecuredText	55
4.26.2.8	ShowLoginPromptDialog	55
4.26.2.9	SingleFieldPromptCompletion	55
4.26.2.10	LoginPromptCompletion	55
4.27	Utility	56
4.27.1	Detailed Description	56
4.27.2	Member Function Documentation	56
4.27.2.1	GetUUID	56
4.27.2.2	OpenStoreLink	56
4.27.2.3	SetApplicationIconBadgeNumber	56
4.27.2.4	GetBundleVersion	57
4.27.2.5	GetBundleIdentifier	57
4.28	UtilitySettings	57
4.28.1	Detailed Description	57
4.29	WebViewMessage	57
4.29.1	Detailed Description	57
4.29.2	Property Documentation	58
4.29.2.1	SchemeName	58
4.29.2.2	Host	58
4.29.2.3	Arguments	58
<b>Index</b>		<b>59</b>

## **Chapter 1**

## **Namespace Index**



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">AddressBook</a>	Address Book provides access to a centralized contacts database. . . . .	11
<a href="#">AddressBookContact</a>	Data container for each contact detail. . . . .	12
<a href="#">MediaLibrarySettings.AndroidSettings</a>	Media Library Settings specific to Android platform. . . . .	14
<a href="#">BillingSettings.AndroidSettings</a>	Billing Settings specific to Android platform. . . . .	14
<a href="#">NetworkConnectivitySettings.AndroidSettings</a>	Network Connectivity Settings specific to Android Editor. . . . .	14
<a href="#">NotificationServiceSettings.AndroidSettings</a>	Notification Service Settings specific to Android platform. . . . .	15
<a href="#">Billing</a>	<a href="#">Billing</a> provides an unique interface to provide in-app purchases from with in the application. . .	16
<a href="#">BillingProduct</a>	This provides unique access to a product which is purchasable. This encloses details for access- ing different properties of a in-app product. . . . .	19
<a href="#">BillingSettings</a>	<a href="#">Billing</a> Settings provides interface to configure properties related to native billing. . . . .	20
<a href="#">BillingTransaction</a>	Maintains transaction information for the purchased product. . . . .	21
<a href="#">CrossPlatformNotification</a>	Is a wrapper class to represent notifications that an application can schedule, handle and present. . . . .	23
<a href="#">NetworkConnectivitySettings.EditorSettings</a>	Network Connectivity Settings specific to Unity Editor. . . . .	24
<a href="#">BillingSettings.iOSSettings</a>	<a href="#">Billing</a> Settings specific to iOS platform. . . . .	25
<a href="#">NotificationServiceSettings.iOSSettings</a>	Notification Service Settings specific to iOS platform. . . . .	26
<a href="#">MediaLibrary</a>	Media library provides an unique way to access devices's media gallery and camera for picking images and playing videos. . . . .	26
<a href="#">MediaLibrarySettings</a>	Media Library Settings provides interface to configure properties related to native media library.	30
<a href="#">NetworkConnectivity</a>	Check if the device is connected to internet. . . . .	30

<a href="#">NetworkConnectivitySettings</a>	
Network Connectivity Settings provides interface to configure properties related to network connectivity. . . . .	31
<a href="#">NotificationService</a>	
Used for scheduling, registering and handling both local and remote notifications. . . . .	32
<a href="#">NotificationServiceSettings</a>	
Notification Service Settings provides interface to configure properties related to notification. . .	35
<a href="#">Sharing</a>	
<a href="#">Sharing</a> gives access to different forms of sharing. Can share text or image with Social network, Mail, SMS, WhatsApp. . . . .	35
<a href="#">Twitter</a>	
Provides support for sending <a href="#">Twitter</a> requests on behalf of the user and for composing and sending tweets. . . . .	44
<a href="#">TwitterSession</a>	
<a href="#">TwitterSession</a> represents a user's session authenticated with the <a href="#">Twitter</a> API. . . . .	50
<a href="#">TwitterSettings</a>	
<a href="#">Twitter</a> Settings provides interface to configure TwitterKit. . . . .	51
<a href="#">TwitterUser</a>	
<a href="#">TwitterUser</a> represents a user on <a href="#">Twitter</a> . . . . .	52
<a href="#">UI</a>	
Provides interface for Native <a href="#">UI</a> system. . . . .	53
<a href="#">Utility</a>	
<a href="#">Utility</a> provides an unique way to access some utility functions. . . . .	56
<a href="#">UtilitySettings</a>	
<a href="#">Utility</a> Settings provides interface to configure properties related to utility components such as RateMyApp. . . . .	57
<a href="#">WebViewMessage</a>	
<a href="#">WebViewMessage</a> is a wrapped data which is sent from WebView to Unity. . . . .	57

## Chapter 3

# Namespace Documentation

### 3.1 Package VoxelBusters

#### Namespaces

- package [NativePlugins](#)

### 3.2 Package VoxelBusters.NativePlugins

#### Namespaces

- package [Internal](#)

#### Classes

- class [AddressBook](#)  
*Address Book provides access to a centralized contacts database.*
- class [AddressBookContact](#)  
*Data container for each contact detail.*
- class [Billing](#)  
*Billing provides an unique interface to provide in-app purchases from with in the application.*
- class [BillingProduct](#)  
*This provides unique access to a product which is purchasable. This encloses details for accessing different properties of a in-app product.*
- class [BillingSettings](#)  
*Billing Settings provides interface to configure properties related to native billing.*
- class [BillingTransaction](#)  
*Maintains transaction information for the purchased product.*
- class [CrossPlatformNotification](#)  
*Is a wrapper class to represent notifications that an application can schedule, handle and present.*
- class [MediaLibrary](#)  
*Media library provides an unique way to access devices's media gallery and camera for picking images and playing videos.*
- class [MediaLibrarySettings](#)  
*Media Library Settings provides interface to configure properties related to native media library.*
- class [NetworkConnectivity](#)  
*Check if the device is connected to internet.*

- class [NetworkConnectivitySettings](#)  
*Network Connectivity Settings provides interface to configure properties related to network connectivity.*
- class [NotificationService](#)  
*Used for scheduling, registering and handling both local and remote notifications.*
- class [NotificationServiceSettings](#)  
*Notification Service Settings provides interface to configure properties related to notification.*
- class [Sharing](#)  
*[Sharing](#) gives access to different forms of sharing. Can share text or image with Social network, Mail, SMS, WhatsApp.*
- class [Twitter](#)  
*Provides support for sending [Twitter](#) requests on behalf of the user and for composing and sending tweets.*
- class [TwitterSession](#)  
*[TwitterSession](#) represents a user's session authenticated with the [Twitter](#) API.*
- class [TwitterSettings](#)  
*[Twitter](#) Settings provides interface to configure [TwitterKit](#).*
- class [TwitterUser](#)  
*[TwitterUser](#) represents a user on [Twitter](#).*
- class [UI](#)  
*Provides interface for Native [UI](#) system.*
- class [Utility](#)  
*[Utility](#) provides an unique way to access some utility functions.*
- class [UtilitySettings](#)  
*[Utility](#) Settings provides interface to configure properties related to utility components such as [RateMyApp](#).*
- class [WebViewMessage](#)  
*[WebViewMessage](#) is a wrapped data which is sent from [WebView](#) to [Unity](#).*

## Enumerations

- enum [eABAuthorizationStatus](#) {  
    NOT\_DETERMINED,  
    RESTRICTED,  
    DENIED,  
    AUTHORIZED }  
*Different possible values for the authorization status with respect to address book data access.*
- enum [eBillingTransactionState](#) {  
    PURCHASED,  
    FAILED,  
    RESTORED,  
    REFUNDED }  
*State of purchased transaction*
- enum [eBillingTransactionVerificationState](#) {  
    NOT\_CHECKED,  
    SUCCESS,  
    FAILED }  
*Verification state of purchased transaction.*
- enum [eImageSource](#) {  
    ALBUM,  
    CAMERA,  
    BOTH }  
*Lists different sources available for picking an image.*
- enum [ePickImageFinishReason](#) {  
    SELECTED,  
    CANCELLED,  
    FAILED }



*Lists different status for Pick image action.*

- enum `ePickVideoFinishReason` {  
    SELECTED,  
    CANCELLED,  
    FAILED }

*Lists different status for Pick video action.*

- enum `ePlayVideoFinishReason` {  
    PLAYBACK\_ENDED,  
    PLAYBACK\_ERROR,  
    USER\_EXITED }

*Lists different status for Play video action.*

- enum `NotificationType` {  
    Badge = 1,  
    Sound,  
    Alert = 4 }

*Specifies the types of notifications the app supports.*

- enum `eShareResult` { CLOSED }

*Status of shared action.*

- enum `eShareOptions` { ,  
    MESSAGE,  
    MAIL,  
    FB,  
    TWITTER,  
    WHATSAPP }

*Lists different share options.*

- enum `eTwitterComposerResult` {  
    CANCELLED,  
    DONE }

*Possible reasons when user finishes composing tweets.*

- enum `eWebviewControllerType` {  
    NO\_CONTROLS,  
    CLOSE\_BUTTON,  
    TOOLBAR }

*Specifies the types of control available for webview.*

### 3.2.1 Enumeration Type Documentation

#### 3.2.1.1 enum `eABAuthorizationStatus`

Different possible values for the authorization status with respect to address book data access.

Enumerator

**NOT\_DETERMINED** The user has not yet made a choice regarding whether this app can access the data class.

**RESTRICTED** The application is not authorized to access the address book data.

**DENIED** The user explicitly denied access to address book data for this application.

**AUTHORIZED** The application is authorized to access address book data.

#### 3.2.1.2 enum `eBillingTransactionState`

State of purchased transaction

## Enumerator

**PURCHASED** [Billing](#) product purchase transaction successful.

**FAILED** [Billing](#) product purchase transaction failed.

**RESTORED** [Billing](#) product purchase transaction restored. This will be the state when RestoreCompleted↵ Transactions is called.

**REFUNDED** [Billing](#) product purchase transaction refunded back to the user.

## 3.2.1.3 enum eBillingTransactionVerificationState

Verification state of purchased transaction.

## Enumerator

**NOT\_CHECKED** Transaction verification not initiated.

**SUCCESS** Transaction verification is successful.

**FAILED** Transaction verification failed because of invalid receipt/signature.

## 3.2.1.4 enum eImageSource

Lists different sources available for picking an image.

## Enumerator

**ALBUM** Pick from gallery/album.

**CAMERA** Pick from camera.

**BOTH** Pick from gallery/album and camera.

## 3.2.1.5 enum ePickImageFinishReason

Lists different status for Pick image action.

## Enumerator

**SELECTED** Selected image from specified [eImageSource](#) source.

**CANCELLED** Cancelled picking image action.

**FAILED** Failed picking image from specified [eImageSource](#) source.

## 3.2.1.6 enum ePickVideoFinishReason

Lists different status for Pick video action.

## Enumerator

**SELECTED** Selected video from gallery.

**CANCELLED** Cancelled picking video from gallery.

**FAILED** Picking video from gallery failed.

### 3.2.1.7 enum ePlayVideoFinishReason

Lists different status for Play video action.

Enumerator

**PLAYBACK\_ENDED** Video playback ended. Played till end of the video.

**PLAYBACK\_ERROR** Error while playing video.

**USER\_EXITED** User exited without playing the complete video.

### 3.2.1.8 enum NotificationType

Specifies the types of notifications the app supports.

Enumerator

**Badge** Notification is a badge on application's icon

**Sound** Notification is an alert sound.

**Alert** Notification is an alert message.

### 3.2.1.9 enum eShareResult

Status of shared action.

Enumerator

**CLOSED** [Sharing](#) window got closed.

### 3.2.1.10 enum eShareOptions

Lists different share options.

Enumerator

**MESSAGE** Share via Messaging service. Can be SMS/MMS or Messenger apps on Tablets.

**MAIL** Share via Mail service.

**FB** Share via Facebook service.

**TWITTER** Share via [Twitter](#) service.

**WHATSAPP** Share via WhatsApp service.

### 3.2.1.11 enum eTwitterComposerResult

Possible reasons when user finishes composing tweets.

Enumerator

**CANCELLED** The composer is dismissed without sending the Tweet (i.e. the user selects Cancel, or the account is unavailable).

**DONE** The composer is dismissed and the message is being sent in the background, after the user selects Done.

### 3.2.1.12 enum eWebviewControllerType

Specifies the types of control available for webview.

#### Enumerator

**NO\_CONTROLS** No controls are shown for WebView. Fits well for banner ads.

**CLOSE\_BUTTON** Close button is shown at top-right corner of the webview. Note that WebView is hide on pressing this button. Please call Destroy, to permanetly remove webview instance.

**TOOLBAR** Toolbar consists of 4 buttons Back, Forward, Reload, Done Back and Forward buttons allows the user to move back and forward through the webpage history. Reload button allows user to reload the webpage contents. Done buttons allows user to hide WebView.

## 3.3 Package VoxelBusters.NativePlugins.Internal

## Chapter 4

# Class Documentation

### 4.1 AddressBook

Address Book provides access to a centralized contacts database.

Inherits MonoBehaviour, and MonoBehaviour.

#### Public Member Functions

- virtual [eABAuthorizationStatus](#) [GetAuthorizationStatus](#) ()  
*Get status of the app's access to contact data.*
- virtual void [ReadContacts](#) ([ReadContactsCompletion](#) \_onCompletion)  
*Request to fetch the contacts.*
- delegate void [ReadContactsCompletion](#) ([eABAuthorizationStatus](#) \_authorizationStatus, [AddressBook](#)↔  
[Contact](#)[] \_contactList)  
*Use this delegate type to get callback when reading contacts information completes.*

#### 4.1.1 Detailed Description

Address Book provides access to a centralized contacts database.

Provides access to get First Name, Last Name , Contact Picture, Phone numbers and Email ID list details for each contact.

#### 4.1.2 Member Function Documentation

##### 4.1.2.1 virtual [eABAuthorizationStatus](#) [GetAuthorizationStatus](#) ( ) [virtual]

Get status of the app's access to contact data.

##### Returns

The authorization status.

##### 4.1.2.2 virtual void [ReadContacts](#) ( [ReadContactsCompletion](#) \_onCompletion ) [virtual]

Request to fetch the contacts.

## Parameters

<code>_onCompletion</code>	Callback triggered once reading contacts is finished.
----------------------------	---

4.1.2.3 `delegate void ReadContactsCompletion ( eABAuthorizationStatus _authorizationStatus, AddressBookContact[] _contactList )`

Use this delegate type to get callback when reading contacts information completes.

## Parameters

<code>_authorizationStatus</code>	Authorization status for address book access .
<code>_contactList</code>	List of contacts read from address book database.

## 4.2 AddressBookContact

Data container for each contact detail.

### Public Member Functions

- [AddressBookContact](#) ()  
*Initializes a new instance of the [AddressBookContact](#) class.*
- [AddressBookContact](#) ([AddressBookContact](#) \_source)  
*Initializes a new instance of the [AddressBookContact](#) class by details from \_source.*
- void [GetImageAsync](#) (DownloadTexture.Completion \_onCompletion)  
*Helper for getting Texture from image path.*
- override string [ToString](#) ()  
*String representation of [AddressBookContact](#).*

### Properties

- string [FirstName](#) [get, protected set]  
*First Name of the contact.*
- string [LastName](#) [get, protected set]  
*Last Name of the contact.*
- string [ImagePath](#) [get, protected set]  
*Absolute image path of the contact.*
- List< string > [PhoneNumberList](#) [get, protected set]  
*List of phone numbers in this contact.*
- List< string > [EmailIDList](#) [get, protected set]  
*List of Email ID's in this contact.*

#### 4.2.1 Detailed Description

Data container for each contact detail.

## 4.2.2 Constructor & Destructor Documentation

### 4.2.2.1 AddressBookContact ( )

Initializes a new instance of the [AddressBookContact](#) class.

### 4.2.2.2 AddressBookContact ( AddressBookContact \_source )

Initializes a new instance of the [AddressBookContact](#) class by details from \_source.

Parameters

<code>_source</code>	Source to copy from.
----------------------	----------------------

## 4.2.3 Member Function Documentation

### 4.2.3.1 void GetImageAsync ( DownloadTexture.Completion \_onCompletion )

Helper for getting Texture from image path.

Parameters

<code>_onCompletion</code>	Callback to be triggered after downloading the image.
----------------------------	---

### 4.2.3.2 override string ToString ( )

String representation of [AddressBookContact](#).

## 4.2.4 Property Documentation

### 4.2.4.1 string FirstName [get],[protected set]

First Name of the contact.

### 4.2.4.2 string LastName [get],[protected set]

Last Name of the contact.

### 4.2.4.3 string ImagePath [get],[protected set]

Absolute image path of the contact.

### 4.2.4.4 List<string> PhoneNumberList [get],[protected set]

List of phone numbers in this contact.

### 4.2.4.5 List<string> EmailIDList [get],[protected set]

List of Email ID's in this contact.

## 4.3 MediaLibrarySettings.AndroidSettings

Media Library Settings specific to Android platform.

### Properties

- string [YoutubeAPIKey](#) [get, set]  
*Gets or sets the youtube API key.*

### 4.3.1 Detailed Description

Media Library Settings specific to Android platform.

### 4.3.2 Property Documentation

#### 4.3.2.1 string YoutubeAPIKey [get], [set]

Gets or sets the youtube API key.

The youtube API key.

For more information, please check <https://developers.google.com/youtube/v3/>.

## 4.4 BillingSettings.AndroidSettings

[Billing](#) Settings specific to Android platform.

### 4.4.1 Detailed Description

[Billing](#) Settings specific to Android platform.

## 4.5 NetworkConnectivitySettings.AndroidSettings

Network Connectivity Settings specific to Android Editor.

### Properties

- int [TimeOutPeriod](#) [get, set]  
*Gets or sets the time out period.*
- int [MaxRetryCount](#) [get, set]  
*Gets or sets the max retry count, whenever polling fails.*
- float [TimeGapBetweenPolling](#) [get, set]  
*Gets or sets the time gap between successive polling.*

### 4.5.1 Detailed Description

Network Connectivity Settings specific to Android Editor.



## 4.5.2 Property Documentation

### 4.5.2.1 int TimeOutPeriod [get], [set]

Gets or sets the time out period.

The time out period.

### 4.5.2.2 int MaxRetryCount [get], [set]

Gets or sets the max retry count, whenever polling fails.

The max retry count.

### 4.5.2.3 float TimeGapBetweenPolling [get], [set]

Gets or sets the time gap between successive polling.

The time gap between polling.

## 4.6 NotificationServiceSettings.AndroidSettings

Notification Service Settings specific to Android platform.

### Properties

- string[] [SenderIdList](#) [get, set]  
*Gets or sets the sender identifier list.*
- bool [AllowCustomIcon](#) [get, set]  
*Selects if custom icon needs to be considered for notification icon.*
- string [TickerTextKey](#) [get, set]  
*Gets or sets the ticker text key.*
- string [ContentTitleKey](#) [get, set]  
*Gets or sets the content title key.*
- string [ContentTextKey](#) [get, set]  
*Gets or sets the content text key.*
- string [UserInfoKey](#) [get, set]  
*Gets or sets the user info key.*
- string [TagKey](#) [get, set]  
*Gets or sets the tag key.*

### 4.6.1 Detailed Description

Notification Service Settings specific to Android platform.

## 4.6.2 Property Documentation

### 4.6.2.1 string[] SenderIDList [get], [set]

Gets or sets the sender identifier list.

Contains list of sender ids.

**Note**

Sender ID list is required to receive Remote Notifications. Please check in Google Play Developer Console, select your application -> SERVICES & APIS -> GOOGLE CLOUD MESSAGING section. /value>

**4.6.2.2 bool AllowCustomIcon** [get], [set]

Selects if custom icon needs to be considered for notification icon.

**Note**

: You need to update the icon in drawable folder {Assets/Plugins/Android/native-plugins\_lib/res/Drawable/[app\_icon\_custom\_white.png/app\_icon\_custom\_coloured.png]}

If true selects the custom icon, else uses the app icon for drawing.

**4.6.2.3 string TickerTextKey** [get], [set]

Gets or sets the ticker text key.

Custom ticker text key used for remote notifications.

**4.6.2.4 string ContentTitleKey** [get], [set]

Gets or sets the content title key.

Custom content title key used for remote notifications.

**4.6.2.5 string ContentTextKey** [get], [set]

Gets or sets the content text key.

Custom content text key used in remote notifications.

**4.6.2.6 string UserInfoKey** [get], [set]

Gets or sets the user info key.

User info key where user can add custom data and retrieve back with UserInfo variable of [CrossPlatformNotification](#).

**4.6.2.7 string TagKey** [get], [set]

Gets or sets the tag key.

The Tag key used in remote notifications. Specifying uniquely can avoid overwriting previous notification displayed.

## 4.7 Billing

[Billing](#) provides an unique interface to provide in-app purchases from within the application.

Inherits MonoBehaviour, and MonoBehaviour.

## Public Member Functions

- virtual void [RequestForBillingProducts](#) (List< [BillingProduct](#) > \_billingProducts)  
*Request details for given list of billing products.*
- virtual bool [IsProductPurchased](#) (string \_productID)  
*Check purchase status of specified billing product.*
- virtual void [BuyProduct](#) (string \_productID)  
*Buy the specified billing product.*
- virtual void [RestoreCompletedTransactions](#) ()  
*Restores the previous non-consumable/Managed billing products that were purchased previously, if any.*
- virtual void [CustomVerificationFinished](#) ([BillingTransaction](#) \_transaction)  
*This adds the specified billing transaction to the inventory of purchased products.*
- delegate void [BillingProductsRequestCompletion](#) (List< [BillingProduct](#) > \_regProductsList, string \_error)  
*Use this delegate type to get callback when products request is finished.*
- delegate void [TransactionCompletion](#) (List< [BillingTransaction](#) > \_finishedTransactions)  
*Use this delegate type to get callback when any transaction gets finished.*

## Events

- static [BillingProductsRequestCompletion](#) [BillingProductsRequestFinishedEvent](#)  
*Occurs when billing products request finished.*
- static [TransactionCompletion](#) [TransactionFinishedEvent](#)  
*Occurs when transaction is finished and ready with a result. See [eBillingTransactionState](#) for the status of the transaction.*

### 4.7.1 Detailed Description

[Billing](#) provides an unique interface to provide in-app purchases from with in the application.

You can set the list of billing products you will use for billing in the NPSettings editor utility or can provide details of the products runtime.

### 4.7.2 Member Function Documentation

#### 4.7.2.1 virtual void RequestForBillingProducts ( List< [BillingProduct](#) > \_billingProducts ) [virtual]

Request details for given list of billing products.

#### Warning

Details of billing products needs to be fetched first with this method before purchasing any product.

#### Parameters

<a href="#">_billingProducts</a>	List of billing products which needs details.
----------------------------------	---

#### 4.7.2.2 virtual bool IsProductPurchased ( string \_productID ) [virtual]

Check purchase status of specified billing product.

This works only for non-consumable/Managed billing products. For consumable, this always returns false.

#### Returns

true if the specified \_productID is already purchased; otherwise, false.

## Parameters

<code>_productID</code>	Product ID to check for purchase status.
-------------------------	--

4.7.2.3 `virtual void BuyProduct ( string _productID ) [virtual]`

Buy the specified billing product.

## Note

`_productID` should already have billing product details ahead. see [RequestForBillingProducts](#) for getting the details.

## Parameters

<code>_productID</code>	product identifier that needs to be purchased.
-------------------------	--

4.7.2.4 `virtual void RestoreCompletedTransactions ( ) [virtual]`

Restores the previous non-consumable/Managed billing products that were purchased previously, if any.

4.7.2.5 `virtual void CustomVerificationFinished ( BillingTransaction _transaction ) [virtual]`

This adds the specified billing transaction to the inventory of purchased products.

## Note

iOS has external validation option. So once external validation is done, this should be called with the transaction details. On Android this call has no effect as the products are internally verified and products will be updated internally right after the purchase.

## Parameters

<code>_transaction</code>	Transaction instance thats updated with verification state after custom verification.
---------------------------	---

4.7.2.6 `delegate void BillingProductsRequestCompletion ( List< BillingProduct > _regProductsList, string _error )`

Use this delegate type to get callback when products request is finished.

## Parameters

<code>_regProductsList</code>	List of billing products with detailed information.
<code>_error</code>	Error description if request billing products fail.

4.7.2.7 `delegate void TransactionCompletion ( List< BillingTransaction > _finishedTransactions )`

Use this delegate type to get callback when any transaction gets finished.

## Parameters

<code>_finishedTransactions</code>	List of billing transactions. This can be multiple for restore purchases. Rest of the times it will be a single transaction.
------------------------------------	--

### 4.7.3 Event Documentation

#### 4.7.3.1 BillingProductsRequestCompletion BillingProductsRequestFinishedEvent [static]

Occurs when billing products request finished.

#### 4.7.3.2 TransactionCompletion TransactionFinishedEvent [static]

Occurs when transaction is finished and ready with a result. See [eBillingTransactionState](#) for the status of the transaction.

## 4.8 BillingProduct

This provides unique access to a product which is purchasable. This encloses details for accessing different properties of a in-app product.

### Public Member Functions

- [BillingProduct Copy \(\)](#)  
*Copy this instance.*
- override string [ToString \(\)](#)  
*Returns a System.String that represents the current [BillingProduct](#).*

### Properties

- string [Name](#) [get, set]  
*Gets or sets the name of the product.*
- string [Description](#) [get, set]  
*Gets or sets the description.*
- bool [IsConsumable](#) [get, set]  
*Gets or sets a value indicating whether this billing product is consumable.*
- float [Price](#) [get, set]  
*Gets or sets the price.*
- string [LocalizedPrice](#) [get, set]  
*Gets or sets the localized price.*
- string [ProductIdentifier](#) [get, protected set]  
*Gets or sets the product identifier.*

### 4.8.1 Detailed Description

This provides unique access to a product which is purchasable. This encloses details for accessing different properties of a in-app product.

### 4.8.2 Member Function Documentation

#### 4.8.2.1 BillingProduct Copy ( )

Copy this instance.

#### 4.8.2.2 override string ToString ( )

Returns a System.String that represents the current [BillingProduct](#).

##### Returns

A System.String that represents the current [BillingProduct](#).

### 4.8.3 Property Documentation

#### 4.8.3.1 string Name [get], [set]

Gets or sets the name of the product.

Name of the billing product. This is used only for reference or for display.

#### 4.8.3.2 string Description [get], [set]

Gets or sets the description.

Description of the billing product

#### 4.8.3.3 bool IsConsumable [get], [set]

Gets or sets a value indicating whether this billing product is consumable.

true if this billing product is consumable; otherwise, false.

#### 4.8.3.4 float Price [get], [set]

Gets or sets the price.

Amount/value of this billing product in local currency.

#### 4.8.3.5 string LocalizedPrice [get], [set]

Gets or sets the localized price.

Contains localized price of this product with local currency information.

#### 4.8.3.6 string ProductIdentifier [get], [protected set]

Gets or sets the product identifier.

Hold identifier to uniquely identity this billing product.

## 4.9 BillingSettings

[Billing](#) Settings provides interface to configure properties related to native billing.

### Classes

- class [AndroidSettings](#)  
*Billing Settings specific to Android platform.*

- class [iOSSettings](#)  
*Billing Settings specific to iOS platform.*

## Properties

- List< [BillingProduct](#) > [Products](#) [get, set]  
*Gets or sets the billing products.*
- [iOSSettings](#) [iOS](#) [get, set]  
*Gets or sets the [Billing](#) Settings specific to iOS platform.*
- [AndroidSettings](#) [Android](#) [get, set]  
*Gets or sets the [Billing](#) Settings specific to Android platform.*

### 4.9.1 Detailed Description

[Billing](#) Settings provides interface to configure properties related to native billing.

### 4.9.2 Property Documentation

#### 4.9.2.1 List<BillingProduct> Products [get], [set]

Gets or sets the billing products.

The billing products.

#### 4.9.2.2 iOSSettings iOS [get], [set]

Gets or sets the [Billing](#) Settings specific to iOS platform.

The [Billing](#) Settings specific to iOS platform.

#### 4.9.2.3 AndroidSettings Android [get], [set]

Gets or sets the [Billing](#) Settings specific to Android platform.

The [Billing](#) Settings specific to Android platform.

## 4.10 BillingTransaction

Maintains transaction information for the purchased product.

## Properties

- string [ProductIdentifier](#) [get, protected set]  
*Gets the product identifier.*
- System.DateTime [TransactionDateUTC](#) [get, protected set]  
*Gets the transaction date in UTC time zone.*
- System.DateTime [TransactionDateLocal](#) [get, protected set]  
*Gets the transaction date in local timezone.*
- string [TransactionIdentifier](#) [get, protected set]  
*Gets the transaction identifier.*
- string [TransactionReceipt](#) [get, protected set]

*Gets or sets the transaction receipt.*

- `eBillingTransactionState TransactionState` [get, protected set]

*Gets the state of the transaction.*

- `eBillingTransactionVerificationState VerificationState` [get, set]

*Gets the state of the transaction verification.*

- `string Error` [get, protected set]

*Gets the error if transaction fails.*

- `string RawPurchaseData` [get, protected set]

*Gets the purchase data.*

### 4.10.1 Detailed Description

Maintains transaction information for the purchased product.

### 4.10.2 Property Documentation

#### 4.10.2.1 `string ProductIdentifier` [get], [protected set]

Gets the product identifier.

Product identifier which uniquely defines a billing product.

#### 4.10.2.2 `System.DateTime TransactionDateUTC` [get], [protected set]

Gets the transaction date in UTC time zone.

Transaction date in universal time zone

#### 4.10.2.3 `System.DateTime TransactionDateLocal` [get], [protected set]

Gets the transaction date in local timezone.

Transaction date in local time format

#### 4.10.2.4 `string TransactionIdentifier` [get], [protected set]

Gets the transaction identifier.

Unique identifier to identify the transaction.

#### 4.10.2.5 `string TransactionReceipt` [get], [protected set]

Gets or sets the transaction receipt.

#### Note

On iOS this alone is enough to validate a transaction. On Android along with this(signature), original json may be required if you need external validation.

Receipt used to validate this transaction.



#### 4.10.2.6 eBillingTransactionState TransactionState [get], [protected set]

Gets the state of the transaction.

State of the transaction.

#### 4.10.2.7 eBillingTransactionVerificationState VerificationState [get], [set]

Gets the state of the transaction verification.

Transaction verification state.

#### 4.10.2.8 string Error [get], [protected set]

Gets the error if transaction fails.

Error description if any failure in the transaction happens.

#### 4.10.2.9 string RawPurchaseData [get], [protected set]

Gets the purchase data.

The purchase data in raw format.

## 4.11 CrossPlatformNotification

Is a wrapper class to represent notifications that an application can schedule, handle and present.

### Public Member Functions

- override string [ToString](#) ()  
*String representation of [CrossPlatformNotification](#).*

### Properties

- string [AlertBody](#) [get, set]  
*Gets or sets the message displayed in the notification alert.*
- System.DateTime [FireDate](#) [get, set]  
*Gets or sets the date and time when the system should deliver the notification.*
- IDictionary [UserInfo](#) [get, set]  
*Gets or sets the dictionary for passing custom information to the notified application.*
- iOSSpecificProperties [iOSProperties](#) [get, set]  
*Gets or sets the iOS platform specific notification properties*
- AndroidSpecificProperties [AndroidProperties](#) [get, set]  
*Gets or sets the android platform specific notification properties.*

#### 4.11.1 Detailed Description

Is a wrapper class to represent notifications that an application can schedule, handle and present.

### 4.11.2 Member Function Documentation

#### 4.11.2.1 override string ToString ( )

String representation of [CrossPlatformNotification](#).

### 4.11.3 Property Documentation

#### 4.11.3.1 string AlertBody [get], [set]

Gets or sets the message displayed in the notification alert.

The alert body.

#### 4.11.3.2 System.DateTime FireDate [get], [set]

Gets or sets the date and time when the system should deliver the notification.

The fire date.

#### 4.11.3.3 IDictionary UserInfo [get], [set]

Gets or sets the dictionary for passing custom information to the notified application.

The user info.

#### 4.11.3.4 iOSSpecificProperties iOSProperties [get], [set]

Gets or sets the iOS platform specific notification properties

The iOS properties.

#### 4.11.3.5 AndroidSpecificProperties AndroidProperties [get], [set]

Gets or sets the android platform specific notification properties.

The android properties.

## 4.12 NetworkConnectivitySettings.EditorSettings

Network Connectivity Settings specific to Unity Editor.

### Properties

- int [TimeOutPeriod](#) [get, set]  
*Gets or sets the time out period.*
- int [MaxRetryCount](#) [get, set]  
*Gets or sets the max retry count, whenever polling fails.*
- float [TimeGapBetweenPolling](#) [get, set]  
*Gets or sets the time gap between successive polling.*

### 4.12.1 Detailed Description

Network Connectivity Settings specific to Unity Editor.

### 4.12.2 Property Documentation

#### 4.12.2.1 `int TimeOutPeriod` `[get]`, `[set]`

Gets or sets the time out period.

The time out period.

#### 4.12.2.2 `int MaxRetryCount` `[get]`, `[set]`

Gets or sets the max retry count, whenever polling fails.

The max retry count.

#### 4.12.2.3 `float TimeGapBetweenPolling` `[get]`, `[set]`

Gets or sets the time gap between successive polling.

The time gap between polling.

## 4.13 BillingSettings.iOSSettings

[Billing](#) Settings specific to iOS platform.

### Properties

- `bool SupportsReceiptValidation` `[get]`, `[set]`  
*Gets or sets a value indicating whether [Billing](#) supports receipt validation.*
- `bool ValidateUsingAppleServer` `[get]`, `[set]`  
*Gets or sets a value indicating whether [BillingTransaction](#) is validated using apple server or custom server.*
- `string ValidateUsingServerURL` `[get]`, `[set]`  
*Gets or sets the server URL used for [BillingTransaction](#) receipt validation.*

### 4.13.1 Detailed Description

[Billing](#) Settings specific to iOS platform.

### 4.13.2 Property Documentation

#### 4.13.2.1 `bool SupportsReceiptValidation` `[get]`, `[set]`

Gets or sets a value indicating whether [Billing](#) supports receipt validation.

`true` if supports receipt validation; otherwise, `false`.

#### 4.13.2.2 bool ValidateUsingAppleServer [get], [set]

Gets or sets a value indicating whether [BillingTransaction](#) is validated using apple server or custom server.

true if validate using apple server; otherwise, false.

#### 4.13.2.3 string ValidateUsingServerURL [get], [set]

Gets or sets the server URL used for [BillingTransaction](#) receipt validation.

The validate [BillingTransaction](#) receipt using server URL.

Incase if you dont want to use Apple itunes server for receipt validation, then specify the URL that will used for validating transaction receipts.

## 4.14 NotificationServiceSettings.iOSSettings

Notification Service Settings specific to iOS platform.

### Properties

- string [UserInfoKey](#) [get, set]  
*Gets or sets the key used to access user data within notification payload.*

#### 4.14.1 Detailed Description

Notification Service Settings specific to iOS platform.

#### 4.14.2 Property Documentation

##### 4.14.2.1 string UserInfoKey [get], [set]

Gets or sets the key used to access user data within notification payload.

The user info key.

## 4.15 MediaLibrary

Media library provides an unique way to access devices's media gallery and camera for picking images and playing videos.

Inherits MonoBehaviour, and MonoBehaviour.

### Public Member Functions

- virtual bool [IsCameraSupported](#) ()  
*Determines if camera is supported.*
- virtual void [PickImage](#) (eImageSource \_source, float \_scaleFactor, [PickImageCompletion](#) \_onCompletion)  
*Picks the image.*
- void [SaveScreenshotToGallery](#) ([SaveImageToGalleryCompletion](#) \_onCompletion)  
*Saves the screenshot to gallery/media library of the users device.*
- void [SaveImageToGallery](#) (URL \_URL, [SaveImageToGalleryCompletion](#) \_onCompletion)

- Saves an image from the specified url to gallery/media library of the users device.*
- void [SaveImageToGallery](#) (Texture2D \_texture, [SaveImageToGalleryCompletion](#) \_onCompletion)  
*Saves the specified texture to gallery.*
- virtual void [SaveImageToGallery](#) (byte[] \_imageByteArray, [SaveImageToGalleryCompletion](#) \_onCompletion)  
*Saves the specified image data source to gallery.*
- virtual void [PlayYoutubeVideo](#) (string \_videoID, [PlayVideoCompletion](#) \_onCompletion)  
*Plays youtube video for the corresponding video id specified.*
- virtual void [PlayEmbeddedVideo](#) (string \_embedHTMLString, [PlayVideoCompletion](#) \_onCompletion)  
*Plays embedded video using web view.*
- virtual void [PlayVideoFromURL](#) (URL \_URL, [PlayVideoCompletion](#) \_onCompletion)  
*Play video from specified URL. This can be an URL pointing to local/remote file.*
- virtual void [PlayVideoFromGallery](#) ([PickVideoCompletion](#) \_onPickVideoCompletion, [PlayVideoCompletion](#) ↔ \_onPlayVideoCompletion)  
*Allows to pick a video from gallery and play it.*
- delegate void [PickImageCompletion](#) ([ePickImageFinishReason](#) \_reason, Texture2D \_image)  
*Use this delegate type to get callback when pick image action is done. Check \_reason for success or failure. If successful, returns the texture of picked image.*
- delegate void [SaveImageToGalleryCompletion](#) (bool \_savedSuccessfully)  
*Use this delegate type to get callback when image is saved to gallery.*
- delegate void [PickVideoCompletion](#) ([ePickVideoFinishReason](#) \_reason)  
*Use this delegate type to get callback when pick video action is done. Check \_reason for success or failure.*
- delegate void [PlayVideoCompletion](#) ([ePlayVideoFinishReason](#) \_reason)  
*Use this delegate type to get callback when play video action is done. Check \_reason for success or failure.*

### 4.15.1 Detailed Description

Media library provides an unique way to access devices's media gallery and camera for picking images and playing videos.

### 4.15.2 Member Function Documentation

#### 4.15.2.1 virtual bool IsCameraSupported ( ) [virtual]

Determines if camera is supported.

#### Returns

true if camera is supported; otherwise, false.

#### 4.15.2.2 virtual void PickImage ( eImageSource \_source, float \_scaleFactor, [PickImageCompletion](#) \_onCompletion ) [virtual]

Picks the image.

#### Parameters

<code>_source</code>	Specify from where you want to pick the image from. <a href="#">eImageSource</a>
<code>_scaleFactor</code>	Specify if scaled up or scaled down version of pick image is required. 1.0f returns the image with out any modification.

<code>_onCompletion</code>	Callback triggered once Pick from the source is finised.
----------------------------	--

#### 4.15.2.3 void SaveScreenshotToGallery ( SaveImageToGalleryCompletion *\_onCompletion* )

Saves the screenshot to gallery/media library of the users device.

##### Parameters

<code>_onCompletion</code>	Callback triggered once save to gallery is done.
----------------------------	--

#### 4.15.2.4 void SaveImageToGallery ( URL *\_URL*, SaveImageToGalleryCompletion *\_onCompletion* )

Saves an image from the specified url to gallery/media library of the users device.

##### Note

The path needs to be absolute path if its local file. Take care of the path on multiple platforms as the file structure will be different.

##### Parameters

<code>_URL</code>	URL to pick the source from. This can be a file url existing on local storage or a web url at remote location.
<code>_onCompletion</code>	Callback triggered once save to gallery is done.

#### 4.15.2.5 void SaveImageToGallery ( Texture2D *\_texture*, SaveImageToGalleryCompletion *\_onCompletion* )

Saves the specified texture to gallery.

##### Parameters

<code>_texture</code>	Texture that needs to be saved to gallery.
<code>_onCompletion</code>	Callback triggered once save to gallery is done.

#### 4.15.2.6 virtual void SaveImageToGallery ( byte[] *\_imageByteArray*, SaveImageToGalleryCompletion *\_onCompletion* ) [virtual]

Saves the specified image data source to gallery.

##### Parameters

<code>_imageByte↵ Array</code>	image byte array to use as source.
<code>_onCompletion</code>	Callback triggered once save to gallery is done.

#### 4.15.2.7 virtual void PlayYoutubeVideo ( string *\_videoID*, PlayVideoCompletion *\_onCompletion* ) [virtual]

Plays youtube video for the corresponding video id specified.

## Parameters

<code>_videoID</code>	Video id of the youtube video.
<code>_onCompletion</code>	Callback triggered on completion or failure. see <a href="#">ePlayVideoFinishReason</a> for status.

4.15.2.8 `virtual void PlayEmbeddedVideo ( string _embedHTMLString, PlayVideoCompletion _onCompletion )`  
`[virtual]`

Plays embedded video using web view.

## Parameters

<code>_embedHTMLString</code>	Embed HTML string to load in the webview.
<code>_onCompletion</code>	Callback triggered on completion or failure. see <a href="#">ePlayVideoFinishReason</a> for status.

4.15.2.9 `virtual void PlayVideoFromURL ( URL _URL, PlayVideoCompletion _onCompletion )` `[virtual]`

Play video from specified URL. This can be an URL pointing to local/remote file.

## Warning

This URL needs to point directly to the video. Direct video link ex: [www.voxelbusters.com/movie.mp4](http://www.voxelbusters.com/movie.mp4)

## Parameters

<code>_URL</code>	URL of the video to play.
<code>_onCompletion</code>	Callback triggered on completion or failure. see <a href="#">ePlayVideoFinishReason</a> for status.

4.15.2.10 `virtual void PlayVideoFromGallery ( PickVideoCompletion _onPickVideoCompletion, PlayVideoCompletion _onPlayVideoCompletion )` `[virtual]`

Allows to pick a video from gallery and play it.

## Parameters

<code>_onPickVideoCompletion</code>	Triggered once pick video action is done.
<code>_onPlayVideoCompletion</code>	Callback triggered on completion or failure. see <a href="#">ePlayVideoFinishReason</a> for status.

4.15.2.11 `delegate void PickImageCompletion ( ePickImageFinishReason _reason, Texture2D _image )`

Use this delegate type to get callback when pick image action is done. Check `_reason` for success or failure. If successful, returns the texture of picked image.

4.15.2.12 `delegate void SaveImageToGalleryCompletion ( bool _savedSuccessfully )`

Use this delegate type to get callback when image is saved to gallery.

4.15.2.13 `delegate void PickVideoCompletion ( ePickVideoFinishReason _reason )`

Use this delegate type to get callback when pick video action is done. Check `_reason` for success or failure.

#### 4.15.2.14 delegate void PlayVideoCompletion ( ePlayVideoFinishReason \_reason )

Use this delegate type to get callback when play video action is done. Check \_reason for success or failure.

## 4.16 MediaLibrarySettings

Media Library Settings provides interface to configure properties related to native media library.

### Classes

- class [AndroidSettings](#)  
*Media Library Settings specific to Android platform.*

### Properties

- [AndroidSettings Android](#) [get, set]  
*Gets or sets the Media Library Settings specific to Android platform.*

#### 4.16.1 Detailed Description

Media Library Settings provides interface to configure properties related to native media library.

#### 4.16.2 Property Documentation

##### 4.16.2.1 [AndroidSettings Android](#) [get], [set]

Gets or sets the Media Library Settings specific to Android platform.

The android.

## 4.17 NetworkConnectivity

Check if the device is connected to internet.

Inherits MonoBehaviour, and MonoBehaviour.

### Public Member Functions

- virtual void [Initialise](#) ()  
*Initialise Network reachability checking.*
- delegate void [NetworkConnectivityChanged](#) (bool \_isConnected)  
*Use this delegate type to get callback when any network connectivity change is detected.*

### Properties

- bool [IsConnected](#) [get, protected set]  
*Gets a value indicating whether network is connected.*



## Events

- static [NetworkConnectivityChanged](#) [NetworkConnectivityChangedEvent](#)  
*Occurs when network connectivity status changes.*

### 4.17.1 Detailed Description

Check if the device is connected to internet.

### 4.17.2 Member Function Documentation

#### 4.17.2.1 virtual void Initialise ( ) [virtual]

Initialise Network reachability checking.

This starts checking if IP Address psecified in the connectivity settings is reachable or not and delivers events.

#### 4.17.2.2 delegate void NetworkConnectivityChanged ( bool *\_isConnected* )

Use this delegate type to get callback when any network connectivity change is detected.

##### Parameters

<i>_isConnected</i>	true when network is connected to internet, else false.
---------------------	---

### 4.17.3 Property Documentation

#### 4.17.3.1 bool IsConnected [get], [protected set]

Gets a value indicating whether network is connected.

true if network is connected; otherwise, false.

### 4.17.4 Event Documentation

#### 4.17.4.1 NetworkConnectivityChanged NetworkConnectivityChangedEvent [static]

Occurs when network connectivity status changes.

## 4.18 NetworkConnectivitySettings

Network Connectivity Settings provides interface to configure properties related to network connectivity.

## Classes

- class [AndroidSettings](#)  
*Network Connectivity Settings specific to Android Editor.*
- class [EditorSettings](#)  
*Network Connectivity Settings specific to Unity Editor.*

## Properties

- string [IPAddress](#) [get, set]  
*Gets or sets the IP Address. IP Address is used to check connectivity.*
- [EditorSettings Editor](#) [get, set]  
*Gets or sets the Network Connectivity Settings specific to Unity Editor.*
- [AndroidSettings Android](#) [get, set]  
*Gets or sets the Network Connectivity Settings specific to Android platform.*

### 4.18.1 Detailed Description

Network Connectivity Settings provides interface to configure properties related to network connectivity.

### 4.18.2 Property Documentation

#### 4.18.2.1 string [IPAddress](#) [get], [set]

Gets or sets the IP Address. IP Address is used to check connectivity.

Address to check reachability.

#### 4.18.2.2 [EditorSettings Editor](#) [get], [set]

Gets or sets the Network Connectivity Settings specific to Unity Editor.

The android.

#### 4.18.2.3 [AndroidSettings Android](#) [get], [set]

Gets or sets the Network Connectivity Settings specific to Android platform.

The android.

## 4.19 NotificationService

Used for scheduling, registering and handling both local and remote notifications.

Inherits MonoBehaviour, and MonoBehaviour.

### Public Member Functions

- virtual void [RegisterNotificationTypes](#) ([NotificationType](#) \_notificationTypes)  
*Apps that use either local or remote notifications must register the types of notifications they intend to deliver.*
- virtual string [ScheduleLocalNotification](#) ([CrossPlatformNotification](#) \_notification)  
*Schedules a local notification.*
- virtual void [CancelLocalNotification](#) (string \_notificationID)  
*Cancels the delivery of the specified scheduled local notification.*
- virtual void [CancelAllLocalNotification](#) ()  
*Cancels the delivery of all scheduled local notification.*
- virtual void [ClearNotifications](#) ()  
*Discards all received notifications.*
- virtual void [RegisterForRemoteNotifications](#) ()

*Register to receive remote notifications via Push Notification service.*

- virtual void [UnregisterForRemoteNotifications](#) ()

*Unregister for all remote notifications received via Push Notification service.*

- delegate void [RegisterForRemoteNotificationCompletion](#) (string \_deviceToken, string \_error)

*Use this delegate type to get callback when app finishes registering with Push Notification service.*

- delegate void [ReceivedNotificationResponse](#) ([CrossPlatformNotification](#) \_notification)

*Use this delegate type to get callback when app receives a notification.*

## Events

- static [RegisterForRemoteNotificationCompletion](#) [DidFinishRegisterForRemoteNotificationEvent](#)

*Occurs when app finishes registering with Push Notification service.*

- static [ReceivedNotificationResponse](#) [DidReceiveLocalNotificationEvent](#)

*Occurs when app receives a local notification.*

- static [ReceivedNotificationResponse](#) [DidReceiveRemoteNotificationEvent](#)

*Occurs when app receives a remote notification.*

### 4.19.1 Detailed Description

Used for scheduling, registering and handling both local and remote notifications.

### 4.19.2 Member Function Documentation

#### 4.19.2.1 virtual void RegisterNotificationTypes ( [NotificationType](#) \_notificationTypes ) [virtual]

Apps that use either local or remote notifications must register the types of notifications they intend to deliver.

##### Parameters

<a href="#">_notificationTypes</a>	The notification types that your app supports.
------------------------------------	--

If your app displays alerts, play sounds, or badges its icon, you must call this method during your launch.

#### 4.19.2.2 virtual string ScheduleLocalNotification ( [CrossPlatformNotification](#) \_notification ) [virtual]

Schedules a local notification.

##### Returns

Notification ID that can be used to uniquely identify every scheduled notification.

##### Parameters

<a href="#">_notification</a>	Notification to be scheduled.
-------------------------------	-------------------------------

#### 4.19.2.3 virtual void CancelLocalNotification ( string \_notificationID ) [virtual]

Cancels the delivery of the specified scheduled local notification.

## Parameters

<code>_notificationID</code>	Identifier for the scheduled notification to be cancelled.
------------------------------	--

4.19.2.4 `virtual void CancelAllLocalNotification ( ) [virtual]`

Cancels the delivery of all scheduled local notification.

4.19.2.5 `virtual void ClearNotifications ( ) [virtual]`

Discards all received notifications.

4.19.2.6 `virtual void RegisterForRemoteNotifications ( ) [virtual]`

Register to receive remote notifications via Push Notification service.

4.19.2.7 `virtual void UnregisterForRemoteNotifications ( ) [virtual]`

Unregister for all remote notifications received via Push Notification service.

4.19.2.8 `delegate void RegisterForRemoteNotificationCompletion ( string _deviceToken, string _error )`

Use this delegate type to get callback when app finishes registering with Push Notification service.

## Parameters

<code>_deviceToken</code>	The device token received after registering successfully. This will be nil if register fails with error.
<code>_error</code>	An error that details why registration failed.

4.19.2.9 `delegate void ReceivedNotificationResponse ( CrossPlatformNotification _notification )`

Use this delegate type to get callback when app receives a notification.

## Parameters

<code>_notification</code>	Received notification.
----------------------------	------------------------

## 4.19.3 Event Documentation

4.19.3.1 `RegisterForRemoteNotificationCompletion DidFinishRegisterForRemoteNotificationEvent [static]`

Occurs when app finishes registering with Push Notification service.

After you call the [RegisterForRemoteNotifications](#) method, the app calls this method when device registration completes successfully or when there is an error in the registration process. When registration completes successfully, connect with your push notification server and give the device token to it. Push notification server pushes notifications only to the device represented by the token.

4.19.3.2 `ReceivedNotificationResponse DidReceiveLocalNotificationEvent [static]`

Occurs when app receives a local notification.

#### 4.19.3.3 ReceivedNotificationResponse DidReceiveRemoteNotificationEvent [static]

Occurs when app receives a remote notification.

## 4.20 NotificationServiceSettings

Notification Service Settings provides interface to configure properties related to notification.

### Classes

- class [AndroidSettings](#)  
*Notification Service Settings specific to Android platform.*
- class [iOSSettings](#)  
*Notification Service Settings specific to iOS platform.*

### Properties

- [iOSSettings iOS](#) [get, set]  
*Gets or sets the Notification Service Settings specific to iOS platform.*
- [AndroidSettings Android](#) [get, set]  
*Gets or sets the Notification Service Settings specific to Android platform.*

#### 4.20.1 Detailed Description

Notification Service Settings provides interface to configure properties related to notification.

#### 4.20.2 Property Documentation

##### 4.20.2.1 iOSSettings iOS [get], [set]

Gets or sets the Notification Service Settings specific to iOS platform.

The i O.

##### 4.20.2.2 AndroidSettings Android [get], [set]

Gets or sets the Notification Service Settings specific to Android platform.

The android.

## 4.21 Sharing

[Sharing](#) gives access to different forms of sharing. Can share text or image with Social network, Mail, SMS, WhatsApp.

Inherits [MonoBehaviour](#), [MonoBehaviour](#), [MonoBehaviour](#), [MonoBehaviour](#), [MonoBehaviour](#), [MonoBehaviour](#), [MonoBehaviour](#), and [MonoBehaviour](#).

## Public Member Functions

- void [ShareTextMessageOnSocialNetwork](#) (string \_message, [SharingCompletion](#) \_onCompletion)  
*Shares the text message on social network.*
- void [ShareURLOnSocialNetwork](#) (string \_message, string \_URLString, [SharingCompletion](#) \_onCompletion)  
*Shares the given URL string on social network.*
- void [ShareScreenShotOnSocialNetwork](#) (string \_message, [SharingCompletion](#) \_onCompletion)  
*Shares the current screenshot on social network.*
- void [ShareImageOnSocialNetwork](#) (string \_message, Texture2D \_texture, [SharingCompletion](#) \_onCompletion)  
*Shares the image on social network.*
- void [ShareImageOnSocialNetwork](#) (string \_message, string \_imagePath, [SharingCompletion](#) \_onCompletion)  
*Shares the image on social network.*
- void [ShareMessage](#) (string \_message, [eShareOptions](#)[] \_excludedOptions, [SharingCompletion](#) \_onCompletion)  
*Shares the message.*
- void [ShareURL](#) (string \_message, string \_URLString, [eShareOptions](#)[] \_excludedOptions, [SharingCompletion](#) \_onCompletion)  
*Shares the URL string.*
- void [ShareScreenShot](#) (string \_message, [eShareOptions](#)[] \_excludedOptions, [SharingCompletion](#) \_onCompletion)  
*Shares the current screenshot.*
- void [ShareImage](#) (string \_message, Texture2D \_texture, [eShareOptions](#)[] \_excludedOptions, [SharingCompletion](#) \_onCompletion)  
*Shares the image.*
- void [ShareImageAtPath](#) (string \_message, string \_imagePath, [eShareOptions](#)[] \_excludedOptions, [SharingCompletion](#) \_onCompletion)  
*Shares the image at path.*
- void [Share](#) (string \_message, string \_URLString, byte[] \_imageByteArray, [eShareOptions](#)[] \_excludedOptions, [SharingCompletion](#) \_onCompletion)  
*Share message , URL or image*
- delegate void [SharingCompletion](#) ([eShareResult](#) \_result)  
*Use this delegate type to get callback when sharing action is done.*
- virtual bool [IsMailServiceAvailable](#) ()  
*Determines whether mail service is available.*
- void [SendPlainTextMail](#) (string \_subject, string \_body, string[] \_recipients, [SharingCompletion](#) \_onCompletion)  
*Sends the plain text mail.*
- void [SendHTMLTextMail](#) (string \_subject, string \_htmlBody, string[] \_recipients, [SharingCompletion](#) \_onCompletion)  
*Sends the HTML text mail.*
- void [SendMailWithScreenshot](#) (string \_subject, string \_body, bool \_isHTMLBody, string[] \_recipients, [SharingCompletion](#) \_onCompletion)  
*Sends the mail with screenshot.*
- void [SendMailWithTexture](#) (string \_subject, string \_body, bool \_isHTMLBody, Texture2D \_texture, string[] \_recipients, [SharingCompletion](#) \_onCompletion)  
*Sends the mail with texture as image.*
- void [SendMailWithAttachment](#) (string \_subject, string \_body, bool \_isHTMLBody, string \_attachmentPath, string \_mimeType, string[] \_recipients, [SharingCompletion](#) \_onCompletion)  
*Sends the mail with an attachment.*
- virtual void [SendMail](#) (string \_subject, string \_body, bool \_isHTMLBody, byte[] \_attachmentByteArray, string \_mimeType, string \_attachmentFileNameWithExtn, string[] \_recipients, [SharingCompletion](#) \_onCompletion)  
*Sends the mail with attachment from byte array.*

- virtual bool `IsMessagingServiceAvailable ()`  
*Determines whether messaging service is available.*
- virtual void `SendTextMessage (string _body, string[] _recipients, SharingCompletion _onCompletion)`  
*Sends the text message.*
- virtual bool `IsWhatsAppServiceAvailable ()`  
*Determines whether whatsapp service is available.*
- virtual void `ShareTextMessageOnWhatsApp (string _message, SharingCompletion _onCompletion)`  
*Shares the text message on whatsapp.*
- void `ShareScreenshotOnWhatsApp (SharingCompletion _onCompletion)`  
*Shares the screenshot on whatsapp.*
- void `ShareImageOnWhatsApp (string _imagePath, SharingCompletion _onCompletion)`  
*Shares the image on whatsapp.*
- void `ShareImageOnWhatsApp (Texture2D _texture, SharingCompletion _onCompletion)`  
*Shares the image on whatsapp.*
- virtual void `ShareImageOnWhatsApp (byte[] _imageByteArray, SharingCompletion _onCompletion)`  
*Shares the image on whatsapp.*

#### 4.21.1 Detailed Description

`Sharing` gives access to different forms of sharing. Can share text or image with Social network, Mail, SMS, WhatsApp.

#### 4.21.2 Member Function Documentation

##### 4.21.2.1 void ShareTextMessageOnSocialNetwork ( string \_message, SharingCompletion \_onCompletion )

Shares the text message on social network.

Parameters

<code>_message</code>	Message to share.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

##### 4.21.2.2 void ShareURLOnSocialNetwork ( string \_message, string \_URLString, SharingCompletion \_onCompletion )

Shares the given URL string on social network.

Parameters

<code>_message</code>	Message to share.
<code>_URLString</code>	URL string to share.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

##### 4.21.2.3 void ShareScreenShotOnSocialNetwork ( string \_message, SharingCompletion \_onCompletion )

Shares the current screenshot on social network.

Parameters

<code>_message</code>	Message to share.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.4 `void ShareImageOnSocialNetwork ( string _message, Texture2D _texture, SharingCompletion _onCompletion )`

Shares the image on social network.

Parameters

<code>_message</code>	Message to share.
<code>_texture</code>	Texture to share.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.5 `void ShareImageOnSocialNetwork ( string _message, string _imagePath, SharingCompletion _onCompletion )`

Shares the image on social network.

Parameters

<code>_message</code>	Message to share.
<code>_imagePath</code>	Path where image exists that needs to be shared.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.6 `void ShareMessage ( string _message, eShareOptions[] _excludedOptions, SharingCompletion _onCompletion )`

Shares the message.

Parameters

<code>_message</code>	Message to share.
<code>_excludedOptions</code>	List of sharing services to exclude while sharing.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.7 `void ShareURL ( string _message, string _URLString, eShareOptions[] _excludedOptions, SharingCompletion _onCompletion )`

Shares the URL string.

Parameters

<code>_message</code>	Message to share.
<code>_URLString</code>	URL string to share.
<code>_excludedOptions</code>	List of sharing services to exclude while sharing.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.8 `void ShareScreenShot ( string _message, eShareOptions[] _excludedOptions, SharingCompletion _onCompletion )`

Shares the current screenshot.



## Parameters

<code>_message</code>	Message to share.
<code>_excluded↔ Options</code>	List of sharing services to exclude while sharing.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

**4.21.2.9** `void ShareImage ( string _message, Texture2D _texture, eShareOptions[] _excludedOptions, SharingCompletion _onCompletion )`

Shares the image.

## Parameters

<code>_message</code>	Message to share.
<code>_texture</code>	Texture to share as image.
<code>_excluded↔ Options</code>	List of sharing services to exclude while sharing.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

**4.21.2.10** `void ShareImageAtPath ( string _message, string _imagePath, eShareOptions[] _excludedOptions, SharingCompletion _onCompletion )`

Shares the image at path.

## Parameters

<code>_message</code>	Message to share.
<code>_imagePath</code>	Path where image exists that needs to be shared.
<code>_excluded↔ Options</code>	List of sharing services to exclude while sharing.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

**4.21.2.11** `void Share ( string _message, string _URLString, byte[] _imageByteArray, eShareOptions[] _excludedOptions, SharingCompletion _onCompletion )`

Share message , URL or image

## Parameters

<code>_message</code>	Message to share.
<code>_URLString</code>	URL string to share.
<code>_imageByte↔ Array</code>	image byte array to create the image from.
<code>_excluded↔ Options</code>	List of sharing services to exclude while sharing.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

**4.21.2.12** `delegate void SharingCompletion ( eShareResult _result )`

Use this delegate type to get callback when sharing action is done.

## Parameters

<code>_result</code>	Status of share action. Currently CLOSED action is only available
----------------------	---

4.21.2.13 `virtual bool IsMailServiceAvailable ( ) [virtual]`

Determines whether mail service is available.

## Returns

`true` if this mail service is available; otherwise, `false`.

4.21.2.14 `void SendPlainTextMail ( string _subject, string _body, string[] _recipients, SharingCompletion _onCompletion )`

Sends the plain text mail.

## Parameters

<code>_subject</code>	Subject of the mail.
<code>_body</code>	Body of the mail.
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.15 `void SendHTMLTextMail ( string _subject, string _htmlBody, string[] _recipients, SharingCompletion _onCompletion )`

Sends the HTML text mail.

## Parameters

<code>_subject</code>	Subject of the mail.
<code>_htmlBody</code>	HTML string for body of the mail.
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.16 `void SendMailWithScreenshot ( string _subject, string _body, bool _isHTMLBody, string[] _recipients, SharingCompletion _onCompletion )`

Sends the mail with screenshot.

## Parameters

<code>_subject</code>	Subject of mail.
<code>_body</code>	Body of the mail.
<code>_isHTMLBody</code>	Indicates if the sent body string is HTML string.
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.17 `void SendMailWithTexture ( string _subject, string _body, bool _isHTMLBody, Texture2D _texture, string[] _recipients, SharingCompletion _onCompletion )`

Sends the mail with texture as image.

## Parameters

<code>_subject</code>	Subject of mail.
<code>_body</code>	Body of the mail.
<code>_isHTMLBody</code>	Indicates if the sent body string is HTML string.
<code>_texture</code>	Texture to create the image from.
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.18 `void SendMailWithAttachment ( string _subject, string _body, bool _isHTMLBody, string _attachmentPath, string _mimeType, string[] _recipients, SharingCompletion _onCompletion )`

Sends the mail with an attachment.

## Parameters

<code>_subject</code>	Subject of mail.
<code>_body</code>	Body of the mail.
<code>_isHTMLBody</code>	Indicates if the sent body string is HTML string.
<code>_attachmentPath</code>	Path to attachment file.
<code>_mimeType</code>	MIME type of attachment. ex: image/png, application/pdf .
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.19 `virtual void SendMail ( string _subject, string _body, bool _isHTMLBody, byte[] _attachmentByteArray, string _mimeType, string _attachmentFileNameWithExtn, string[] _recipients, SharingCompletion _onCompletion )`  
[virtual]

Sends the mail with attachment from byte array.

## Parameters

<code>_subject</code>	Subject of mail.
<code>_body</code>	Body of the mail.
<code>_isHTMLBody</code>	Indicates if the sent body string is HTML string.
<code>_attachment↵ ByteArray</code>	Attachment byte array.
<code>_mimeType</code>	MIME type of attachment. ex: image/png, application/pdf .
<code>_attachment↵ FileNameWith↵ Extn</code>	Attachment file name with extension.
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.20 `virtual bool IsMessagingServiceAvailable ( )` [virtual]

Determines whether messaging service is available.

## Returns

`true` if this messaging service is available; otherwise, `false`.

4.21.2.21 `virtual void SendTextMessage ( string _body, string[] _recipients, SharingCompletion _onCompletion )`  
[virtual]

Sends the text message.

## Parameters

<code>_body</code>	Body of message.
<code>_recipients</code>	List of recipients.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.22 `virtual bool IsWhatsAppServiceAvailable ( ) [virtual]`

Determines whether whatsapp service is available.

## Returns

`true` if whatsapp service is available; otherwise, `false`.

4.21.2.23 `virtual void ShareTextMessageOnWhatsApp ( string _message, SharingCompletion _onCompletion ) [virtual]`

Shares the text message on whatsapp.

## Parameters

<code>_message</code>	Message to share.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.24 `void ShareScreenshotOnWhatsApp ( SharingCompletion _onCompletion )`

Shares the screenshot on whatsapp.

## Parameters

<code>_onCompletion</code>	Callback to be triggered when sharing action completes.
----------------------------	---

4.21.2.25 `void ShareImageOnWhatsApp ( string _imagePath, SharingCompletion _onCompletion )`

Shares the image on whatsapp.

## Parameters

<code>_imagePath</code>	Path of the image to be shared.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.26 `void ShareImageOnWhatsApp ( Texture2D _texture, SharingCompletion _onCompletion )`

Shares the image on whatsapp.

## Parameters

<code>_texture</code>	Texture to take the image from.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

4.21.2.27 `virtual void ShareImageOnWhatsApp ( byte[] _imageByteArray, SharingCompletion _onCompletion ) [virtual]`

Shares the image on whatsapp.

## Parameters

<code>_imageByte↔ Array</code>	Image byte array to create the image from.
<code>_onCompletion</code>	Callback to be triggered when sharing action completes.

## 4.22 Twitter

Provides support for sending [Twitter](#) requests on behalf of the user and for composing and sending tweets.

Inherits [MonoBehaviour](#), and [MonoBehaviour](#).

### Public Member Functions

- virtual bool [Initialise](#) ()  
*Initialise [Twitter](#) feature.*
- virtual void [Login](#) ([TWTRLoginCompletion](#) \_onCompletion)  
*Triggers user authentication with [Twitter](#).*
- virtual void [Logout](#) ()  
*Deletes the local [Twitter](#) user session from this app.*
- virtual bool [IsLoggedIn](#) ()  
*Determines whether user is logged in or not.*
- virtual string [GetAuthToken](#) ()  
*Gets the authorization token. Returns NULL if user isnt logged in.*
- virtual string [GetAuthTokenSecret](#) ()  
*Gets the authorization token secret. Returns NULL if user isnt logged in.*
- virtual string [GetUserID](#) ()  
*Gets the user ID of twitter user. Returns NULL if user isnt logged in.*
- virtual string [GetUserName](#) ()  
*Gets the name of the twitter user. Returns NULL if user isnt logged in.*
- void [ShowTweetComposerWithMessage](#) (string \_message, [TWTRTweetCompletion](#) \_onCompletion)  
*Presents composer to tweet text.*
- void [ShowTweetComposerWithScreenshot](#) (string \_message, [TWTRTweetCompletion](#) \_onCompletion)  
*Presents composer to tweet text along with screenshot.*
- void [ShowTweetComposerWithImage](#) (string \_message, Texture2D \_texture, [TWTRTweetCompletion](#) \_on↔  
Completion)  
*Presents composer to tweet text along with attached image.*
- void [ShowTweetComposerWithLink](#) (string \_message, string \_URL, [TWTRTweetCompletion](#) \_onCompletion)  
*Presents composer to tweet text along with URL.*
- virtual void [ShowTweetComposer](#) (string \_message, string \_URL, byte[] \_imgByteArray, [TWTRTweet↔  
Completion](#) \_onCompletion)  
*Presents composer to tweet text along with URL and image.*
- virtual void [RequestAccountDetails](#) ([TWTRAccountDetailsCompletion](#) \_onCompletion)  
*Sends request to load users account details.*
- virtual void [RequestEmailAccess](#) ([TWTREmailAccessCompletion](#) \_onCompletion)  
*Requests user to provide access to his email address.*
- void [GetURLRequest](#) (string \_URL, IDictionary \_parameters, [TWTRResonse](#) \_onCompletion)  
*Sends a [Twitter](#) GET request.*
- void [PostURLRequest](#) (string \_URL, IDictionary \_parameters, [TWTRResonse](#) \_onCompletion)  
*Sends a [Twitter](#) POST request.*
- void [PutURLRequest](#) (string \_URL, IDictionary \_parameters, [TWTRResonse](#) \_onCompletion)

*Sends a [Twitter](#) PUT request.*

- void [DeleteURLRequest](#) (string \_URL, IDictionary \_parameters, [TWTRResonse](#) \_onCompletion)

*Sends a [Twitter](#) DELETE request.*

- delegate void [TWTRLoginCompletion](#) ([TwitterSession](#) \_session, string \_error)

*Use this delegate type to get callback when user login succeeds or fails.*

- delegate void [TWTRTweetCompletion](#) ([eTwitterComposerResult](#) \_result)

*Use this delegate type to get callback when the user finishes composing a Tweet.*

- delegate void [TWTRAccountDetailsCompletion](#) ([TwitterUser](#) \_user, string \_error)

*Use this delegate type to get callback when load user details request succeeds or fails.*

- delegate void [TWTREmailAccessCompletion](#) (string \_email, string \_error)

*Use this delegate type to get callback when the user accepts or denies access to their email address.*

- delegate void [TWTRResonse](#) (IDictionary \_response, string \_error)

*Use this delegate type to get callback when the network request succeeds or fails.*

### 4.22.1 Detailed Description

Provides support for sending [Twitter](#) requests on behalf of the user and for composing and sending tweets.

### 4.22.2 Member Function Documentation

#### 4.22.2.1 virtual bool Initialise ( ) [virtual]

Initialise [Twitter](#) feature.

Make sure this is called before using twitter functions. Returns true on proper initialisation.

#### 4.22.2.2 virtual void Login ( [TWTRLoginCompletion](#) \_onCompletion ) [virtual]

Triggers user authentication with [Twitter](#).

Parameters

<a href="#">_onCompletion</a>	Calls the delegate after authentication is successful or if there is an error.
-------------------------------	--

This method will present [UI](#) to allow the user to log in if there are no saved [Twitter](#) login credentials.

#### 4.22.2.3 virtual void Logout ( ) [virtual]

Deletes the local [Twitter](#) user session from this app.

#### 4.22.2.4 virtual bool IsLoggedIn ( ) [virtual]

Determines whether user is logged in or not.

Returns

true if user is logged in; otherwise, false.

#### 4.22.2.5 virtual string GetAuthToken ( ) [virtual]

Gets the authorization token. Returns NULL if user isnt logged in.

Returns

The authorization token.

#### 4.22.2.6 `virtual string GetAuthTokenSecret ( ) [virtual]`

Gets the authorization token secret. Returns NULL if user isnt logged in.

##### Returns

The authorization token secret.

#### 4.22.2.7 `virtual string GetUserID ( ) [virtual]`

Gets the user ID of twitter user. Returns NULL if user isnt logged in.

##### Returns

The user I.

#### 4.22.2.8 `virtual string GetUserName ( ) [virtual]`

Gets the name of the twitter user. Returns NULL if user isnt logged in.

##### Returns

The user name.

#### 4.22.2.9 `void ShowTweetComposerWithMessage ( string _message, TWTRTweetCompletion _onCompletion )`

Presents composer to tweet text.

##### Parameters

<code>_message</code>	The text to tweet.
<code>_onCompletion</code>	Calls the delegate when the user finished or cancelled the Tweet composition.

#### 4.22.2.10 `void ShowTweetComposerWithScreenshot ( string _message, TWTRTweetCompletion _onCompletion )`

Presents composer to tweet text along with screenshot.

##### Parameters

<code>_message</code>	The text to tweet.
<code>_onCompletion</code>	Calls the delegate when the user finished or cancelled the Tweet composition.

#### 4.22.2.11 `void ShowTweetComposerWithImage ( string _message, Texture2D _texture, TWTRTweetCompletion _onCompletion )`

Presents composer to tweet text along with attached image.

##### Parameters

<code>_message</code>	The text to tweet.
-----------------------	--------------------



<code>_texture</code>	The image to attached along with tweet message.
<code>_onCompletion</code>	Calls the delegate when the user finished or cancelled the Tweet composition.

4.22.2.12 `void ShowTweetComposerWithLink ( string _message, string _URL, TWTRTweetCompletion _onCompletion )`

Presents composer to tweet text along with URL.

Parameters

<code>_message</code>	The text to tweet.
<code>_URL</code>	URL to be shared.
<code>_onCompletion</code>	Calls the delegate when the user finished or cancelled the Tweet composition.

4.22.2.13 `virtual void ShowTweetComposer ( string _message, string _URL, byte[] _imgByteArray, TWTRTweetCompletion _onCompletion ) [virtual]`

Presents composer to tweet text along with URL and image.

Parameters

<code>_message</code>	The text to tweet.
<code>_URL</code>	URL to be shared.
<code>_imgByteArray</code>	The image data to be attached along with the tweet.
<code>_onCompletion</code>	Calls the delegate when the user finished or cancelled the Tweet composition.

4.22.2.14 `virtual void RequestAccountDetails ( TWTRAccountDetailsCompletion _onCompletion ) [virtual]`

Sends request to load users account details.

Parameters

<code>_onCompletion</code>	Calls the delegate when load user details request succeeds or fails.
----------------------------	--

4.22.2.15 `virtual void RequestEmailAccess ( TWTREmailAccessCompletion _onCompletion ) [virtual]`

Requests user to provide access to his email address.

Parameters

<code>_onCompletion</code>	Calls the delegate when user either accepts or denies access to their email address.
----------------------------	--

4.22.2.16 `void GetURLRequest ( string _URL, IDictionary _parameters, TWTRResonse _onCompletion )`

Sends a [Twitter](#) GET request.

Parameters

<code>_URL</code>	Request URL. This is the full <a href="#">Twitter</a> API URL. E.g. <a href="https://api.twitter.com/1.1/statuses/user_timeline.json">https://api.twitter.com/1.1/statuses/user_timeline.json</a> .
<code>_parameters</code>	Request parameters.
<code>_onCompletion</code>	Calls the delegate on response.

4.22.2.17 void PostURLRequest ( string *\_URL*, IDictionary *\_parameters*, TWTRResonse *\_onCompletion* )

Sends a [Twitter](#) POST request.

## Parameters

<code>_URL</code>	Request URL. This is the full <a href="#">Twitter</a> API URL. E.g. <a href="https://api.twitter.com/1.1/statuses/user_timeline.json">https://api.twitter.com/1.1/statuses/user_timeline.json</a> .
<code>_parameters</code>	Request parameters.
<code>_onCompletion</code>	Calls the delegate on response.

4.22.2.18 `void PutURLRequest ( string _URL, IDictionary _parameters, TWTRResonse _onCompletion )`

Sends a [Twitter](#) PUT request.

## Parameters

<code>_URL</code>	Request URL. This is the full <a href="#">Twitter</a> API URL. E.g. <a href="https://api.twitter.com/1.1/statuses/user_timeline.json">https://api.twitter.com/1.1/statuses/user_timeline.json</a> .
<code>_parameters</code>	Request parameters.
<code>_onCompletion</code>	Calls the delegate on response.

4.22.2.19 `void DeleteURLRequest ( string _URL, IDictionary _parameters, TWTRResonse _onCompletion )`

Sends a [Twitter](#) DELETE request.

## Parameters

<code>_URL</code>	Request URL. This is the full <a href="#">Twitter</a> API URL. E.g. <a href="https://api.twitter.com/1.1/statuses/user_timeline.json">https://api.twitter.com/1.1/statuses/user_timeline.json</a> .
<code>_parameters</code>	Request parameters.
<code>_onCompletion</code>	Calls the delegate on response.

4.22.2.20 `delegate void TWTRLoginCompletion ( TwitterSession _session, string _error )`

Use this delegate type to get callback when user login succeeds or fails.

## Parameters

<code>_session</code>	Contains the OAuth tokens and minimal information associated with the logged in user or nil.
<code>_error</code>	Error that will be non nil if the authentication request failed.

4.22.2.21 `delegate void TWTRTweetCompletion ( eTwitterComposerResult _result )`

Use this delegate type to get callback when the user finishes composing a Tweet.

4.22.2.22 `delegate void TWTRAccountDetailsCompletion ( TwitterUser _user, string _error )`

Use this delegate type to get callback when load user details request succeeds or fails.

## Parameters

<code>_user</code>	The <a href="#">Twitter</a> User.
<code>_error</code>	Error that will be set if the API request failed.

4.22.2.23 `delegate void TWTREmailAccessCompletion ( string _email, string _error )`

Use this delegate type to get callback when the user accepts or denies access to their email address.

## Parameters

<code>_email</code>	The user's email address. This will be nil if the user does not grant access to their email address or your application is not allowed to request email addresses.
<code>_error</code>	An error that details why a user's email address could not be provided.

4.22.2.24 `delegate void TWTRResonse ( IDictionary _response, string _error )`

Use this delegate type to get callback when the network request succeeds or fails.

## Parameters

<code>_response</code>	Content data of the response.
<code>_error</code>	Describes the network error that occurred.

## 4.23 TwitterSession

[TwitterSession](#) represents a user's session authenticated with the [Twitter](#) API.

### Public Member Functions

- override string [ToString](#) ()  
*String representation of [TwitterSession](#).*

### Properties

- string [AuthToken](#) [get, protected set]  
*Gets a value indicating authorization token.*
- string [AuthTokenSecret](#) [get, protected set]  
*Gets a value indicating authorization token secret.*
- string [UserName](#) [get, protected set]  
*Gets a value indicating username associated with the access token.*
- string [UserID](#) [get, protected set]  
*Gets a value indicating user ID associated with the access token.*

#### 4.23.1 Detailed Description

[TwitterSession](#) represents a user's session authenticated with the [Twitter](#) API.

#### 4.23.2 Member Function Documentation

##### 4.23.2.1 override string ToString ( )

String representation of [TwitterSession](#).

#### 4.23.3 Property Documentation

##### 4.23.3.1 string AuthToken [get],[protected set]

Gets a value indicating authorization token.

The auth token.

**4.23.3.2** `string AuthTokenSecret` `[get], [protected set]`

Gets a value indicating authorization token secret.

The auth token secret.

**4.23.3.3** `string UserName` `[get], [protected set]`

Gets a value indicating username associated with the access token.

The name of the user.

**4.23.3.4** `string UserID` `[get], [protected set]`

Gets a value indicating user ID associated with the access token.

The user ID.

## 4.24 TwitterSettings

[Twitter](#) Settings provides interface to configure TwitterKit.

### Properties

- `string ConsumerKey` `[get, set]`  
*Gets or sets the consumer key.*
- `string ConsumerSecret` `[get, set]`  
*Gets or sets the consumer secret.*

### 4.24.1 Detailed Description

[Twitter](#) Settings provides interface to configure TwitterKit.

To configure TwitterKit, you will need a consumer key and a consumer secret (collectively, a “Twitter app”). Visit Twitter’s developer site (<https://apps.twitter.com/>) to create or review an application settings.

### 4.24.2 Property Documentation

**4.24.2.1** `string ConsumerKey` `[get], [set]`

Gets or sets the consumer key.

The consumer key.

**4.24.2.2** `string ConsumerSecret` `[get], [set]`

Gets or sets the consumer secret.

The consumer secret.

## 4.25 TwitterUser

[TwitterUser](#) represents a user on [Twitter](#).

### Public Member Functions

- override string [ToString](#) ()  
*String representation of [TwitterUser](#).*

### Properties

- string [UserID](#) [get, protected set]  
*Gets a value indicating ID of the [Twitter](#) User.*
- string [Name](#) [get, protected set]  
*Gets a value indicating name of the [Twitter](#) User.*
- bool [IsVerified](#) [get, protected set]  
*Gets a value indicating whether the user has been verified by [Twitter](#).*
- bool [IsProtected](#) [get, protected set]  
*Gets a value indicating whether user is protected.*
- string [ProfileImageUrl](#) [get, protected set]  
*Gets a value indicating HTTPS URL of the user's profile image.*

#### 4.25.1 Detailed Description

[TwitterUser](#) represents a user on [Twitter](#).

#### 4.25.2 Member Function Documentation

##### 4.25.2.1 override string ToString ( )

String representation of [TwitterUser](#).

#### 4.25.3 Property Documentation

##### 4.25.3.1 string UserID [get],[protected set]

Gets a value indicating ID of the [Twitter](#) User.

The user ID.

##### 4.25.3.2 string Name [get],[protected set]

Gets a value indicating name of the [Twitter](#) User.

The name.

##### 4.25.3.3 bool IsVerified [get],[protected set]

Gets a value indicating whether the user has been verified by [Twitter](#).

true if this user is verified; otherwise, false.

#### 4.25.3.4 bool IsProtected [get], [protected set]

Gets a value indicating whether user is protected.

true if this user is protected; otherwise, false.

#### 4.25.3.5 string ProfileImageUrl [get], [protected set]

Gets a value indicating HTTPS URL of the user's profile image.

The profile image UR.

## 4.26 UI

Provides interface for Native UI system.

Inherits MonoBehaviour, MonoBehaviour, MonoBehaviour, MonoBehaviour, and MonoBehaviour.

### Public Member Functions

- void [ShowAlertDialogWithSingleButton](#) (string \_title, string \_message, string \_button, [AlertDialogCompletion](#) \_onCompletion)  
*Shows the alert dialog with single button.*
- void [ShowAlertDialogWithMultipleButtons](#) (string \_title, string \_message, string[] \_buttonsList, [AlertDialogCompletion](#) \_onCompletion)  
*Shows the alert dialog with multiple buttons.*
- delegate void [AlertDialogCompletion](#) (string \_buttonPressed)  
*Use this delegate type to get callback when the user clicks a button on an alert dialog.*
- virtual void [SetPopoverPoint](#) (Vector2 \_position)  
*Sets position of popover controller to given position. Popover controllers are used to Pick Media and to present Share options in iPad. Default position is set to (0.0, 0.0). Note that this is iOS only feature. On Android this call has no effect.*
- void [SetPopoverPointAtLastTouchPosition](#) ()  
*Sets position of popover controller to last touch position. Popover controllers are used to Pick Media and to present Share options in iPad. Default position is set to (0.0, 0.0). Note that this is iOS only feature. On Android this call has no effect.*
- void [ShowSingleFieldPromptDialogWithPlainText](#) (string \_title, string \_message, string \_placeholder, string[] \_buttonsList, [SingleFieldPromptCompletion](#) \_onCompletion)  
*Prompt dialog that allows the user to enter text.*
- void [ShowSingleFieldPromptDialogWithSecuredText](#) (string \_title, string \_message, string \_placeholder, string[] \_buttonsList, [SingleFieldPromptCompletion](#) \_onCompletion)  
*Prompt dialog that allows the user to enter text. The text field is obscured.*
- virtual void [ShowLoginPromptDialog](#) (string \_title, string \_message, string \_usernamePlaceholder, string \_passwordPlaceholder, string[] \_buttonsList, [LoginPromptCompletion](#) \_onCompletion)  
*Prompt dialog that allows the user to enter login identifier and password.*
- delegate void [SingleFieldPromptCompletion](#) (string \_buttonPressed, string \_inputText)  
*Use this delegate type to get callback when the user clicks a button on single field prompt dialog.*
- delegate void [LoginPromptCompletion](#) (string \_buttonPressed, string \_usernameText, string \_passwordText)  
*Use this delegate type to get callback when the user clicks a button on login prompt dialog.*

### 4.26.1 Detailed Description

Provides interface for Native UI system.

## 4.26.2 Member Function Documentation

### 4.26.2.1 void ShowAlertDialogWithSingleButton ( string \_title, string \_message, string \_button, AlertDialogCompletion \_onCompletion )

Shows the alert dialog with single button.

Parameters

<code>_title</code>	Text that appears in title bar.
<code>_message</code>	Descriptive text that provides more details.
<code>_button</code>	Title of action button.
<code>_onCompletion</code>	Calls the delegate when action button is pressed.

### 4.26.2.2 void ShowAlertDialogWithMultipleButtons ( string \_title, string \_message, string[] \_buttonsList, AlertDialogCompletion \_onCompletion )

Shows the alert dialog with multiple buttons.

Parameters

<code>_title</code>	Text that appears in title bar.
<code>_message</code>	Descriptive text that provides more details.
<code>_buttonsList</code>	Title of action buttons.
<code>_onCompletion</code>	Calls the delegate when action button is pressed.

### 4.26.2.3 delegate void AlertDialogCompletion ( string \_buttonPressed )

Use this delegate type to get callback when the user clicks a button on an alert dialog.

Parameters

<code>_buttonPressed</code>	Title of the button that was pressed.
-----------------------------	---------------------------------------

### 4.26.2.4 virtual void SetPopoverPoint ( Vector2 \_position ) [virtual]

Sets position of popover controller to given position. Popover controllers are used to Pick Media and to present Share options in iPad. Default position is set to (0.0, 0.0). Note that this is iOS only feature. On Android this call has no effect.

Parameters

<code>_position</code>	Screen position where popover is displayed.
------------------------	---

### 4.26.2.5 void SetPopoverPointAtLastTouchPosition ( )

Sets position of popover controller to last touch position. Popover controllers are used to Pick Media and to present Share options in iPad. Default position is set to (0.0, 0.0). Note that this is iOS only feature. On Android this call has no effect.

### 4.26.2.6 void ShowSingleFieldPromptDialogWithPlainText ( string \_title, string \_message, string \_placeholder, string[] \_buttonsList, SingleFieldPromptCompletion \_onCompletion )

Prompt dialog that allows the user to enter text.



## Parameters

<code>_title</code>	Text that appears in title bar.
<code>_message</code>	Descriptive text that provides more details.
<code>_placeholder</code>	String that is displayed when there is no other text in the textfield.
<code>_buttonsList</code>	Title of action buttons.
<code>_onCompletion</code>	Calls the delegate when action button is pressed.

4.26.2.7 `void ShowSingleFieldPromptDialogWithSecuredText ( string _title, string _message, string _placeholder, string[] _buttonsList, SingleFieldPromptCompletion _onCompletion )`

Prompt dialog that allows the user to enter text. The text field is obscured.

## Parameters

<code>_title</code>	Text that appears in title bar.
<code>_message</code>	Descriptive text that provides more details.
<code>_placeholder</code>	String that is displayed when there is no other text in the textfield.
<code>_buttonsList</code>	Title of action buttons.
<code>_onCompletion</code>	Calls the delegate when action button is pressed.

4.26.2.8 `virtual void ShowLoginPromptDialog ( string _title, string _message, string _usernamePlaceHolder, string _passwordPlaceHolder, string[] _buttonsList, LoginPromptCompletion _onCompletion ) [virtual]`

Prompt dialog that allows the user to enter login identifier and password.

## Parameters

<code>_title</code>	Text that appears in title bar.
<code>_message</code>	Descriptive text that provides more details.
<code>_usernamePlaceHolder</code>	String that is displayed when there is no other text in the username textfield.
<code>_passwordPlaceHolder</code>	String that is displayed when there is no other text in the password textfield.
<code>_buttonsList</code>	Title of action buttons.
<code>_onCompletion</code>	Calls the delegate when action button is pressed.

4.26.2.9 `delegate void SingleFieldPromptCompletion ( string _buttonPressed, string _inputText )`

Use this delegate type to get callback when the user clicks a button on single field prompt dialog.

## Parameters

<code>_buttonPressed</code>	Title of the button that was pressed.
<code>_inputText</code>	Text contents of textfield.

4.26.2.10 `delegate void LoginPromptCompletion ( string _buttonPressed, string _usernameText, string _passwordText )`

Use this delegate type to get callback when the user clicks a button on login prompt dialog.

## Parameters

<code>_buttonPressed</code>	Title of the button that was pressed.
<code>_usernameText</code>	Text contents of login textfield.
<code>_passwordText</code>	Text contents of password textfield.

## 4.27 Utility

[Utility](#) provides an unique way to access some utility functions.

Inherits MonoBehaviour.

### Public Member Functions

- virtual string [GetUUID](#) ()  
*Gets the UUID.*
- virtual void [OpenStoreLink](#) (string \_applicationID)  
*Opens the store link for that platform.*
- virtual void [SetApplicationIconBadgeNumber](#) (int \_badgeNumber)  
*Sets the number as the badge of the app icon.*
- string [GetBundleVersion](#) ()  
*Gets the bundle version.*
- string [GetBundleIdentifier](#) ()  
*Gets the bundle identifier.*

### 4.27.1 Detailed Description

[Utility](#) provides an unique way to access some utility functions.

### 4.27.2 Member Function Documentation

#### 4.27.2.1 virtual string GetUUID ( ) [virtual]

Gets the UUID.

Returns

An unique identifier.

#### 4.27.2.2 virtual void OpenStoreLink ( string \_applicationID ) [virtual]

Opens the store link for that platform.

Parameters

<code>_applicationID</code>	Application identifier which defines the application uniquely. On iOS this is the id that identifies app on appstore. On Android this is the build identifier (com.example.test)
-----------------------------	--

#### 4.27.2.3 virtual void SetApplicationIconBadgeNumber ( int \_badgeNumber ) [virtual]

Sets the number as the badge of the app icon.

## Parameters

<code>_badgeNumber</code>	The number currently set as badge.
---------------------------	------------------------------------

4.27.2.4 `string GetBundleVersion ( )`

Gets the bundle version.

## Returns

Application bundle version.

4.27.2.5 `string GetBundleIdentifier ( )`

Gets the bundle identifier.

## Returns

Application bundle identifier.

## 4.28 UtilitySettings

[Utility](#) Settings provides interface to configure properties related to utility components such as RateMyApp.

### 4.28.1 Detailed Description

[Utility](#) Settings provides interface to configure properties related to utility components such as RateMyApp.

## 4.29 WebViewMessage

[WebViewMessage](#) is a wrapped data which is sent from WebView to Unity.

### Properties

- `string SchemeName [get, protected set]`  
*Gets the scheme name.*
- `string Host [get, protected set]`  
*Gets the host name.*
- `Dictionary< string, string > Arguments [get, protected set]`  
*Gets the arguments. Each key-value pair represents name and value of the argument*

### 4.29.1 Detailed Description

[WebViewMessage](#) is a wrapped data which is sent from WebView to Unity.

## 4.29.2 Property Documentation

### 4.29.2.1 `string SchemeName` `[get]`, `[protected set]`

Gets the scheme name.

The scheme name.

### 4.29.2.2 `string Host` `[get]`, `[protected set]`

Gets the host name.

The host name.

### 4.29.2.3 `Dictionary<string, string> Arguments` `[get]`, `[protected set]`

Gets the arguments. Each key-value pair represents name and value of the argument

The arguments.

# Index

- ALBUM
  - VoxelBusters::NativePlugins, [8](#)
- AUTHORIZED
  - VoxelBusters::NativePlugins, [7](#)
- AddressBook, [11](#)
- AddressBookContact, [12](#)
  - VoxelBusters::NativePlugins::AddressBook↔  
Contact, [13](#)
- Alert
  - VoxelBusters::NativePlugins, [9](#)
- AlertBody
  - VoxelBusters::NativePlugins::CrossPlatform↔  
Notification, [24](#)
- AlertDialogCompletion
  - VoxelBusters::NativePlugins::UI, [54](#)
- AllowCustomIcon
  - VoxelBusters::NativePlugins::NotificationService↔  
Settings::AndroidSettings, [16](#)
- Android
  - VoxelBusters::NativePlugins::BillingSettings, [21](#)
  - VoxelBusters::NativePlugins::MediaLibrary↔  
Settings, [30](#)
  - VoxelBusters::NativePlugins::NetworkConnectivity↔  
Settings, [32](#)
  - VoxelBusters::NativePlugins::NotificationService↔  
Settings, [35](#)
- AndroidProperties
  - VoxelBusters::NativePlugins::CrossPlatform↔  
Notification, [24](#)
- Arguments
  - VoxelBusters::NativePlugins::WebViewMessage,  
[58](#)
- AuthToken
  - VoxelBusters::NativePlugins::TwitterSession, [50](#)
- AuthTokenSecret
  - VoxelBusters::NativePlugins::TwitterSession, [51](#)
- BOTH
  - VoxelBusters::NativePlugins, [8](#)
- Badge
  - VoxelBusters::NativePlugins, [9](#)
- Billing, [16](#)
- BillingProduct, [19](#)
- BillingProductsRequestCompletion
  - VoxelBusters::NativePlugins::Billing, [18](#)
- BillingProductsRequestFinishedEvent
  - VoxelBusters::NativePlugins::Billing, [19](#)
- BillingSettings, [20](#)
- BillingSettings.AndroidSettings, [14](#)
- BillingSettings.iOSSettings, [25](#)
- BillingTransaction, [21](#)
- BuyProduct
  - VoxelBusters::NativePlugins::Billing, [18](#)
- CAMERA
  - VoxelBusters::NativePlugins, [8](#)
- CANCELLED
  - VoxelBusters::NativePlugins, [8, 9](#)
- CLOSE\_BUTTON
  - VoxelBusters::NativePlugins, [10](#)
- CLOSED
  - VoxelBusters::NativePlugins, [9](#)
- CancelAllLocalNotification
  - VoxelBusters::NativePlugins::NotificationService,  
[34](#)
- CancelLocalNotification
  - VoxelBusters::NativePlugins::NotificationService,  
[33](#)
- ClearNotifications
  - VoxelBusters::NativePlugins::NotificationService,  
[34](#)
- ConsumerKey
  - VoxelBusters::NativePlugins::TwitterSettings, [51](#)
- ConsumerSecret
  - VoxelBusters::NativePlugins::TwitterSettings, [51](#)
- ContentTextKey
  - VoxelBusters::NativePlugins::NotificationService↔  
Settings::AndroidSettings, [16](#)
- ContentTitleKey
  - VoxelBusters::NativePlugins::NotificationService↔  
Settings::AndroidSettings, [16](#)
- Copy
  - VoxelBusters::NativePlugins::BillingProduct, [19](#)
- CrossPlatformNotification, [23](#)
- CustomVerificationFinished
  - VoxelBusters::NativePlugins::Billing, [18](#)
- DENIED
  - VoxelBusters::NativePlugins, [7](#)
- DONE
  - VoxelBusters::NativePlugins, [9](#)
- DeleteURLRequest
  - VoxelBusters::NativePlugins::Twitter, [49](#)
- Description
  - VoxelBusters::NativePlugins::BillingProduct, [20](#)
- DidFinishRegisterForRemoteNotificationEvent
  - VoxelBusters::NativePlugins::NotificationService,  
[34](#)
- DidReceiveLocalNotificationEvent

- VoxelBusters::NativePlugins::NotificationService, 34
- DidReceiveRemoteNotificationEvent
  - VoxelBusters::NativePlugins::NotificationService, 34
- eABAuthorizationStatus
  - VoxelBusters::NativePlugins, 7
- eBillingTransactionState
  - VoxelBusters::NativePlugins, 7
- eBillingTransactionVerificationState
  - VoxelBusters::NativePlugins, 8
- eImageSource
  - VoxelBusters::NativePlugins, 8
- ePickImageFinishReason
  - VoxelBusters::NativePlugins, 8
- ePickVideoFinishReason
  - VoxelBusters::NativePlugins, 8
- ePlayVideoFinishReason
  - VoxelBusters::NativePlugins, 8
- eShareOptions
  - VoxelBusters::NativePlugins, 9
- eShareResult
  - VoxelBusters::NativePlugins, 9
- eTwitterComposerResult
  - VoxelBusters::NativePlugins, 9
- eWebviewControllerType
  - VoxelBusters::NativePlugins, 9
- Editor
  - VoxelBusters::NativePlugins::NetworkConnectivity↔ Settings, 32
- EmailIDList
  - VoxelBusters::NativePlugins::AddressBook↔ Contact, 13
- Error
  - VoxelBusters::NativePlugins::BillingTransaction, 23
- FAILED
  - VoxelBusters::NativePlugins, 8
- FB
  - VoxelBusters::NativePlugins, 9
- FireDate
  - VoxelBusters::NativePlugins::CrossPlatform↔ Notification, 24
- FirstName
  - VoxelBusters::NativePlugins::AddressBook↔ Contact, 13
- GetAuthToken
  - VoxelBusters::NativePlugins::Twitter, 45
- GetAuthTokenSecret
  - VoxelBusters::NativePlugins::Twitter, 45
- GetAuthorizationStatus
  - VoxelBusters::NativePlugins::AddressBook, 11
- GetBundleIdentifier
  - VoxelBusters::NativePlugins::Utility, 57
- GetBundleVersion
  - VoxelBusters::NativePlugins::Utility, 57
- GetImageAsync
  - VoxelBusters::NativePlugins::AddressBook↔ Contact, 13
- GetURLRequest
  - VoxelBusters::NativePlugins::Twitter, 47
- GetUUID
  - VoxelBusters::NativePlugins::Utility, 56
- GetUserID
  - VoxelBusters::NativePlugins::Twitter, 46
- GetUserName
  - VoxelBusters::NativePlugins::Twitter, 46
- Host
  - VoxelBusters::NativePlugins::WebViewMessage, 58
- iOS
  - VoxelBusters::NativePlugins::BillingSettings, 21
  - VoxelBusters::NativePlugins::NotificationService↔ Settings, 35
- iOSProperties
  - VoxelBusters::NativePlugins::CrossPlatform↔ Notification, 24
- IPAddress
  - VoxelBusters::NativePlugins::NetworkConnectivity↔ Settings, 32
- ImagePath
  - VoxelBusters::NativePlugins::AddressBook↔ Contact, 13
- Initialise
  - VoxelBusters::NativePlugins::NetworkConnectivity, 31
  - VoxelBusters::NativePlugins::Twitter, 45
- IsCameraSupported
  - VoxelBusters::NativePlugins::MediaLibrary, 27
- IsConnected
  - VoxelBusters::NativePlugins::NetworkConnectivity, 31
- IsConsumable
  - VoxelBusters::NativePlugins::BillingProduct, 20
- IsLoggedIn
  - VoxelBusters::NativePlugins::Twitter, 45
- IsMailServiceAvailable
  - VoxelBusters::NativePlugins::Sharing, 40
- IsMessagingServiceAvailable
  - VoxelBusters::NativePlugins::Sharing, 41
- IsProductPurchased
  - VoxelBusters::NativePlugins::Billing, 17
- IsProtected
  - VoxelBusters::NativePlugins::TwitterUser, 52
- IsVerified
  - VoxelBusters::NativePlugins::TwitterUser, 52
- IsWhatsAppServiceAvailable
  - VoxelBusters::NativePlugins::Sharing, 43
- LastName
  - VoxelBusters::NativePlugins::AddressBook↔ Contact, 13
- LocalizedPrice
  - VoxelBusters::NativePlugins::BillingProduct, 20

- Login
  - VoxelBusters::NativePlugins::Twitter, [45](#)
- LoginPromptCompletion
  - VoxelBusters::NativePlugins::UI, [55](#)
- Logout
  - VoxelBusters::NativePlugins::Twitter, [45](#)
- MAIL
  - VoxelBusters::NativePlugins, [9](#)
- MESSAGE
  - VoxelBusters::NativePlugins, [9](#)
- MaxRetryCount
  - VoxelBusters::NativePlugins::NetworkConnectivity↔
    - Settings::AndroidSettings, [15](#)
  - VoxelBusters::NativePlugins::NetworkConnectivity↔
    - Settings::EditorSettings, [25](#)
- MediaLibrary, [26](#)
- MediaLibrarySettings, [30](#)
- MediaLibrarySettings.AndroidSettings, [14](#)
- NO\_CONTROLS
  - VoxelBusters::NativePlugins, [10](#)
- NOT\_CHECKED
  - VoxelBusters::NativePlugins, [8](#)
- NOT\_DETERMINED
  - VoxelBusters::NativePlugins, [7](#)
- Name
  - VoxelBusters::NativePlugins::BillingProduct, [20](#)
  - VoxelBusters::NativePlugins::TwitterUser, [52](#)
- NetworkConnectivity, [30](#)
- NetworkConnectivityChanged
  - VoxelBusters::NativePlugins::NetworkConnectivity, [31](#)
- NetworkConnectivityChangedEvent
  - VoxelBusters::NativePlugins::NetworkConnectivity, [31](#)
- NetworkConnectivitySettings, [31](#)
- NetworkConnectivitySettings.AndroidSettings, [14](#)
- NetworkConnectivitySettings.EditorSettings, [24](#)
- NotificationService, [32](#)
- NotificationServiceSettings, [35](#)
- NotificationServiceSettings.AndroidSettings, [15](#)
- NotificationServiceSettings.iOSSettings, [26](#)
- NotificationType
  - VoxelBusters::NativePlugins, [9](#)
- OpenStoreLink
  - VoxelBusters::NativePlugins::Utility, [56](#)
- PLAYBACK\_ENDED
  - VoxelBusters::NativePlugins, [9](#)
- PLAYBACK\_ERROR
  - VoxelBusters::NativePlugins, [9](#)
- PURCHASED
  - VoxelBusters::NativePlugins, [8](#)
- PhoneNumberList
  - VoxelBusters::NativePlugins::AddressBook↔
    - Contact, [13](#)
- PickImage
  - VoxelBusters::NativePlugins::MediaLibrary, [27](#)
- PickImageCompletion
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- PickVideoCompletion
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- PlayEmbeddedVideo
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- PlayVideoCompletion
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- PlayVideoFromGallery
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- PlayVideoFromURL
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- PlayYoutubeVideo
  - VoxelBusters::NativePlugins::MediaLibrary, [28](#)
- PostURLRequest
  - VoxelBusters::NativePlugins::Twitter, [47](#)
- Price
  - VoxelBusters::NativePlugins::BillingProduct, [20](#)
- ProductIdentifier
  - VoxelBusters::NativePlugins::BillingProduct, [20](#)
  - VoxelBusters::NativePlugins::BillingTransaction, [22](#)
- Products
  - VoxelBusters::NativePlugins::BillingSettings, [21](#)
- ProfileImageURL
  - VoxelBusters::NativePlugins::TwitterUser, [53](#)
- PutURLRequest
  - VoxelBusters::NativePlugins::Twitter, [49](#)
- REFUNDED
  - VoxelBusters::NativePlugins, [8](#)
- RESTORED
  - VoxelBusters::NativePlugins, [8](#)
- RESTRICTED
  - VoxelBusters::NativePlugins, [7](#)
- RawPurchaseData
  - VoxelBusters::NativePlugins::BillingTransaction, [23](#)
- ReadContacts
  - VoxelBusters::NativePlugins::AddressBook, [11](#)
- ReadContactsCompletion
  - VoxelBusters::NativePlugins::AddressBook, [12](#)
- ReceivedNotificationResponse
  - VoxelBusters::NativePlugins::NotificationService, [34](#)
- RegisterForRemoteNotificationCompletion
  - VoxelBusters::NativePlugins::NotificationService, [34](#)
- RegisterForRemoteNotifications
  - VoxelBusters::NativePlugins::NotificationService, [34](#)
- RegisterNotificationTypes
  - VoxelBusters::NativePlugins::NotificationService, [33](#)
- RequestAccountDetails
  - VoxelBusters::NativePlugins::Twitter, [47](#)
- RequestEmailAccess
  - VoxelBusters::NativePlugins::Twitter, [47](#)
- RequestForBillingProducts
  - VoxelBusters::NativePlugins::Billing, [17](#)

- RestoreCompletedTransactions
  - VoxelBusters::NativePlugins::Billing, [18](#)
- SELECTED
  - VoxelBusters::NativePlugins, [8](#)
- SUCCESS
  - VoxelBusters::NativePlugins, [8](#)
- SaveImageToGallery
  - VoxelBusters::NativePlugins::MediaLibrary, [28](#)
- SaveImageToGalleryCompletion
  - VoxelBusters::NativePlugins::MediaLibrary, [29](#)
- SaveScreenshotToGallery
  - VoxelBusters::NativePlugins::MediaLibrary, [28](#)
- ScheduleLocalNotification
  - VoxelBusters::NativePlugins::NotificationService, [33](#)
- SchemaName
  - VoxelBusters::NativePlugins::WebViewMessage, [58](#)
- SendHTMLTextMail
  - VoxelBusters::NativePlugins::Sharing, [40](#)
- SendMail
  - VoxelBusters::NativePlugins::Sharing, [41](#)
- SendMailWithAttachment
  - VoxelBusters::NativePlugins::Sharing, [41](#)
- SendMailWithScreenshot
  - VoxelBusters::NativePlugins::Sharing, [40](#)
- SendMailWithTexture
  - VoxelBusters::NativePlugins::Sharing, [40](#)
- SendPlainTextMail
  - VoxelBusters::NativePlugins::Sharing, [40](#)
- SendTextMessage
  - VoxelBusters::NativePlugins::Sharing, [41](#)
- SenderIdList
  - VoxelBusters::NativePlugins::NotificationService↔  
Settings::AndroidSettings, [15](#)
- SetApplicationIconBadgeNumber
  - VoxelBusters::NativePlugins::Utility, [56](#)
- SetPopoverPoint
  - VoxelBusters::NativePlugins::UI, [54](#)
- SetPopoverPointAtLastTouchPosition
  - VoxelBusters::NativePlugins::UI, [54](#)
- Share
  - VoxelBusters::NativePlugins::Sharing, [39](#)
- ShareImage
  - VoxelBusters::NativePlugins::Sharing, [39](#)
- ShareImageAtPath
  - VoxelBusters::NativePlugins::Sharing, [39](#)
- ShareImageOnSocialNetwork
  - VoxelBusters::NativePlugins::Sharing, [38](#)
- ShareImageOnWhatsApp
  - VoxelBusters::NativePlugins::Sharing, [43](#)
- ShareMessage
  - VoxelBusters::NativePlugins::Sharing, [38](#)
- ShareScreenShot
  - VoxelBusters::NativePlugins::Sharing, [38](#)
- ShareScreenShotOnSocialNetwork
  - VoxelBusters::NativePlugins::Sharing, [37](#)
- ShareScreenshotOnWhatsApp
  - VoxelBusters::NativePlugins::Sharing, [43](#)
- ShareTextMessageOnSocialNetwork
  - VoxelBusters::NativePlugins::Sharing, [37](#)
- ShareTextMessageOnWhatsApp
  - VoxelBusters::NativePlugins::Sharing, [43](#)
- ShareURL
  - VoxelBusters::NativePlugins::Sharing, [38](#)
- ShareURLOnSocialNetwork
  - VoxelBusters::NativePlugins::Sharing, [37](#)
- Sharing, [35](#)
- SharingCompletion
  - VoxelBusters::NativePlugins::Sharing, [39](#)
- ShowAlertDialogWithMultipleButtons
  - VoxelBusters::NativePlugins::UI, [54](#)
- ShowAlertDialogWithSingleButton
  - VoxelBusters::NativePlugins::UI, [54](#)
- ShowLoginPromptDialog
  - VoxelBusters::NativePlugins::UI, [55](#)
- ShowSingleFieldPromptDialogWithPlainText
  - VoxelBusters::NativePlugins::UI, [54](#)
- ShowSingleFieldPromptDialogWithSecuredText
  - VoxelBusters::NativePlugins::UI, [55](#)
- ShowTweetComposer
  - VoxelBusters::NativePlugins::Twitter, [47](#)
- ShowTweetComposerWithImage
  - VoxelBusters::NativePlugins::Twitter, [46](#)
- ShowTweetComposerWithLink
  - VoxelBusters::NativePlugins::Twitter, [47](#)
- ShowTweetComposerWithMessage
  - VoxelBusters::NativePlugins::Twitter, [46](#)
- ShowTweetComposerWithScreenshot
  - VoxelBusters::NativePlugins::Twitter, [46](#)
- SingleFieldPromptCompletion
  - VoxelBusters::NativePlugins::UI, [55](#)
- Sound
  - VoxelBusters::NativePlugins, [9](#)
- SupportsReceiptValidation
  - VoxelBusters::NativePlugins::BillingSettings::iOS↔  
Settings, [25](#)
- TOOLBAR
  - VoxelBusters::NativePlugins, [10](#)
- TWITTER
  - VoxelBusters::NativePlugins, [9](#)
- TWTRAccountDetailsCompletion
  - VoxelBusters::NativePlugins::Twitter, [49](#)
- TWTREmailAccessCompletion
  - VoxelBusters::NativePlugins::Twitter, [49](#)
- TWTRLoginCompletion
  - VoxelBusters::NativePlugins::Twitter, [49](#)
- TWTRResonse
  - VoxelBusters::NativePlugins::Twitter, [50](#)
- TWTRTweetCompletion
  - VoxelBusters::NativePlugins::Twitter, [49](#)
- TagKey
  - VoxelBusters::NativePlugins::NotificationService↔  
Settings::AndroidSettings, [16](#)
- TickerTextKey



- VoxelBusters::NativePlugins::NotificationService↔
  - Settings::AndroidSettings, 16
- TimeGapBetweenPolling
  - VoxelBusters::NativePlugins::NetworkConnectivity↔
    - Settings::AndroidSettings, 15
  - VoxelBusters::NativePlugins::NetworkConnectivity↔
    - Settings::EditorSettings, 25
- TimeOutPeriod
  - VoxelBusters::NativePlugins::NetworkConnectivity↔
    - Settings::AndroidSettings, 15
  - VoxelBusters::NativePlugins::NetworkConnectivity↔
    - Settings::EditorSettings, 25
- ToString
  - VoxelBusters::NativePlugins::AddressBook↔
    - Contact, 13
  - VoxelBusters::NativePlugins::BillingProduct, 19
  - VoxelBusters::NativePlugins::CrossPlatform↔
    - Notification, 24
  - VoxelBusters::NativePlugins::TwitterSession, 50
  - VoxelBusters::NativePlugins::TwitterUser, 52
- TransactionCompletion
  - VoxelBusters::NativePlugins::Billing, 18
- TransactionDateLocal
  - VoxelBusters::NativePlugins::BillingTransaction, 22
- TransactionDateUTC
  - VoxelBusters::NativePlugins::BillingTransaction, 22
- TransactionFinishedEvent
  - VoxelBusters::NativePlugins::Billing, 19
- TransactionIdentifier
  - VoxelBusters::NativePlugins::BillingTransaction, 22
- TransactionReceipt
  - VoxelBusters::NativePlugins::BillingTransaction, 22
- TransactionState
  - VoxelBusters::NativePlugins::BillingTransaction, 22
- Twitter, 44
- TwitterSession, 50
- TwitterSettings, 51
- TwitterUser, 52
- UI, 53
- USER\_EXITED
  - VoxelBusters::NativePlugins, 9
- UnregisterForRemoteNotifications
  - VoxelBusters::NativePlugins::NotificationService, 34
- UserID
  - VoxelBusters::NativePlugins::TwitterSession, 51
  - VoxelBusters::NativePlugins::TwitterUser, 52
- UserInfo
  - VoxelBusters::NativePlugins::CrossPlatform↔
    - Notification, 24
- UserInfoKey
  - VoxelBusters::NativePlugins::NotificationService↔
    - Settings::AndroidSettings, 16
  - VoxelBusters::NativePlugins::NotificationService↔
    - Settings::iOSSettings, 26
- UserName
  - VoxelBusters::NativePlugins::TwitterSession, 51
- Utility, 56
- UtilitySettings, 57
- ValidateUsingAppleServer
  - VoxelBusters::NativePlugins::BillingSettings::iOS↔
    - Settings, 25
- ValidateUsingServerURL
  - VoxelBusters::NativePlugins::BillingSettings::iOS↔
    - Settings, 26
- VerificationState
  - VoxelBusters::NativePlugins::BillingTransaction, 23
- VoxelBusters, 5
- VoxelBusters.NativePlugins, 5
- VoxelBusters.NativePlugins.Internal, 10
- VoxelBusters::NativePlugins
  - ALBUM, 8
  - AUTHORIZED, 7
  - Alert, 9
  - BOTH, 8
  - Badge, 9
  - CAMERA, 8
  - CANCELLED, 8, 9
  - CLOSE\_BUTTON, 10
  - CLOSED, 9
  - DENIED, 7
  - DONE, 9
  - eABAuthorizationStatus, 7
  - eBillingTransactionState, 7
  - eBillingTransactionVerificationState, 8
  - eImageSource, 8
  - ePickImageFinishReason, 8
  - ePickVideoFinishReason, 8
  - ePlayVideoFinishReason, 8
  - eShareOptions, 9
  - eShareResult, 9
  - eTwitterComposerResult, 9
  - eWebviewControllerType, 9
  - FAILED, 8
  - FB, 9
  - MAIL, 9
  - MESSAGE, 9
  - NO\_CONTROLS, 10
  - NOT\_CHECKED, 8
  - NOT\_DETERMINED, 7
  - NotificationType, 9
  - PLAYBACK\_ENDED, 9
  - PLAYBACK\_ERROR, 9
  - PURCHASED, 8
  - REFUNDED, 8
  - RESTORED, 8
  - RESTRICTED, 7
  - SELECTED, 8
  - SUCCESS, 8
  - Sound, 9
  - TOOLBAR, 10
  - TWITTER, 9
  - USER\_EXITED, 9
  - WHATSAPP, 9
- VoxelBusters::NativePlugins::AddressBook
  - GetAuthorizationStatus, 11

- ReadContacts, [11](#)
- ReadContactsCompletion, [12](#)
- VoxelBusters::NativePlugins::AddressBookContact
  - AddressBookContact, [13](#)
  - EmailIDList, [13](#)
  - FirstName, [13](#)
  - GetImageAsync, [13](#)
  - ImagePath, [13](#)
  - LastName, [13](#)
  - PhoneNumberList, [13](#)
  - ToString, [13](#)
- VoxelBusters::NativePlugins::Billing
  - BillingProductsRequestCompletion, [18](#)
  - BillingProductsRequestFinishedEvent, [19](#)
  - BuyProduct, [18](#)
  - CustomVerificationFinished, [18](#)
  - IsProductPurchased, [17](#)
  - RequestForBillingProducts, [17](#)
  - RestoreCompletedTransactions, [18](#)
  - TransactionCompletion, [18](#)
  - TransactionFinishedEvent, [19](#)
- VoxelBusters::NativePlugins::BillingProduct
  - Copy, [19](#)
  - Description, [20](#)
  - IsConsumable, [20](#)
  - LocalizedPrice, [20](#)
  - Name, [20](#)
  - Price, [20](#)
  - ProductIdentifier, [20](#)
  - ToString, [19](#)
- VoxelBusters::NativePlugins::BillingSettings
  - Android, [21](#)
  - iOS, [21](#)
  - Products, [21](#)
- VoxelBusters::NativePlugins::BillingSettings::iOS↔
  - Settings
  - SupportsReceiptValidation, [25](#)
  - ValidateUsingAppleServer, [25](#)
  - ValidateUsingServerURL, [26](#)
- VoxelBusters::NativePlugins::BillingTransaction
  - Error, [23](#)
  - ProductIdentifier, [22](#)
  - RawPurchaseData, [23](#)
  - TransactionDateLocal, [22](#)
  - TransactionDateUTC, [22](#)
  - TransactionIdentifier, [22](#)
  - TransactionReceipt, [22](#)
  - TransactionState, [22](#)
  - VerificationState, [23](#)
- VoxelBusters::NativePlugins::CrossPlatformNotification
  - AlertBody, [24](#)
  - AndroidProperties, [24](#)
  - FireDate, [24](#)
  - iOSProperties, [24](#)
  - ToString, [24](#)
  - UserInfo, [24](#)
- VoxelBusters::NativePlugins::MediaLibrary
  - IsCameraSupported, [27](#)
  - PickImage, [27](#)
  - PickImageCompletion, [29](#)
  - PickVideoCompletion, [29](#)
  - PlayEmbeddedVideo, [29](#)
  - PlayVideoCompletion, [29](#)
  - PlayVideoFromGallery, [29](#)
  - PlayVideoFromURL, [29](#)
  - PlayYoutubeVideo, [28](#)
  - SaveImageToGallery, [28](#)
  - SaveImageToGalleryCompletion, [29](#)
  - SaveScreenshotToGallery, [28](#)
- VoxelBusters::NativePlugins::MediaLibrarySettings
  - Android, [30](#)
- VoxelBusters::NativePlugins::MediaLibrarySettings::↔
  - AndroidSettings
  - YoutubeAPIKey, [14](#)
- VoxelBusters::NativePlugins::NetworkConnectivity
  - Initialise, [31](#)
  - IsConnected, [31](#)
  - NetworkConnectivityChanged, [31](#)
  - NetworkConnectivityChangedEvent, [31](#)
- VoxelBusters::NativePlugins::NetworkConnectivity↔
  - Settings
  - Android, [32](#)
  - Editor, [32](#)
  - IPAddress, [32](#)
- VoxelBusters::NativePlugins::NetworkConnectivity↔
  - Settings::AndroidSettings
  - MaxRetryCount, [15](#)
  - TimeGapBetweenPolling, [15](#)
  - TimeOutPeriod, [15](#)
- VoxelBusters::NativePlugins::NetworkConnectivity↔
  - Settings::EditorSettings
  - MaxRetryCount, [25](#)
  - TimeGapBetweenPolling, [25](#)
  - TimeOutPeriod, [25](#)
- VoxelBusters::NativePlugins::NotificationService
  - CancelAllLocalNotification, [34](#)
  - CancelLocalNotification, [33](#)
  - ClearNotifications, [34](#)
  - DidFinishRegisterForRemoteNotificationEvent, [34](#)
  - DidReceiveLocalNotificationEvent, [34](#)
  - DidReceiveRemoteNotificationEvent, [34](#)
  - ReceivedNotificationResponse, [34](#)
  - RegisterForRemoteNotificationCompletion, [34](#)
  - RegisterForRemoteNotifications, [34](#)
  - RegisterNotificationTypes, [33](#)
  - ScheduleLocalNotification, [33](#)
  - UnregisterForRemoteNotifications, [34](#)
- VoxelBusters::NativePlugins::NotificationService↔
  - Settings
  - Android, [35](#)
  - iOS, [35](#)
- VoxelBusters::NativePlugins::NotificationService↔
  - Settings::AndroidSettings
  - AllowCustomIcon, [16](#)
  - ContentTextKey, [16](#)
  - ContentTitleKey, [16](#)

- SenderIDList, [15](#)
- TagKey, [16](#)
- TickerTextKey, [16](#)
- UserInfoKey, [16](#)
- VoxelBusters::NativePlugins::NotificationService↔
  - Settings::iOSSettings
  - UserInfoKey, [26](#)
- VoxelBusters::NativePlugins::Sharing
  - IsMailServiceAvailable, [40](#)
  - IsMessagingServiceAvailable, [41](#)
  - IsWhatsAppServiceAvailable, [43](#)
  - SendHTMLTextMail, [40](#)
  - SendMail, [41](#)
  - SendMailWithAttachment, [41](#)
  - SendMailWithScreenshot, [40](#)
  - SendMailWithTexture, [40](#)
  - SendPlainTextMail, [40](#)
  - SendTextMessage, [41](#)
  - Share, [39](#)
  - ShareImage, [39](#)
  - ShareImageAtPath, [39](#)
  - ShareImageOnSocialNetwork, [38](#)
  - ShareImageOnWhatsApp, [43](#)
  - ShareMessage, [38](#)
  - ShareScreenShot, [38](#)
  - ShareScreenShotOnSocialNetwork, [37](#)
  - ShareScreenshotOnWhatsApp, [43](#)
  - ShareTextMessageOnSocialNetwork, [37](#)
  - ShareTextMessageOnWhatsApp, [43](#)
  - ShareURL, [38](#)
  - ShareURLOnSocialNetwork, [37](#)
  - SharingCompletion, [39](#)
- VoxelBusters::NativePlugins::Twitter
  - DeleteURLRequest, [49](#)
  - GetAuthToken, [45](#)
  - GetAuthTokenSecret, [45](#)
  - GetURLRequest, [47](#)
  - GetUserID, [46](#)
  - GetUserName, [46](#)
  - Initialise, [45](#)
  - IsLoggedIn, [45](#)
  - Login, [45](#)
  - Logout, [45](#)
  - PostURLRequest, [47](#)
  - PutURLRequest, [49](#)
  - RequestAccountDetails, [47](#)
  - RequestEmailAccess, [47](#)
  - ShowTweetComposer, [47](#)
  - ShowTweetComposerWithImage, [46](#)
  - ShowTweetComposerWithLink, [47](#)
  - ShowTweetComposerWithMessage, [46](#)
  - ShowTweetComposerWithScreenshot, [46](#)
  - TWTRAccountDetailsCompletion, [49](#)
  - TWTREmailAccessCompletion, [49](#)
  - TWTRLoginCompletion, [49](#)
  - TWTRResonse, [50](#)
  - TWTRTweetCompletion, [49](#)
- VoxelBusters::NativePlugins::TwitterSession
  - AuthToken, [50](#)
  - AuthTokenSecret, [51](#)
  - ToString, [50](#)
  - UserID, [51](#)
  - UserName, [51](#)
- VoxelBusters::NativePlugins::TwitterSettings
  - ConsumerKey, [51](#)
  - ConsumerSecret, [51](#)
- VoxelBusters::NativePlugins::TwitterUser
  - IsProtected, [52](#)
  - IsVerified, [52](#)
  - Name, [52](#)
  - ProfileImageURL, [53](#)
  - ToString, [52](#)
  - UserID, [52](#)
- VoxelBusters::NativePlugins::UI
  - AlertDialogCompletion, [54](#)
  - LoginPromptCompletion, [55](#)
  - SetPopoverPoint, [54](#)
  - SetPopoverPointAtLastTouchPosition, [54](#)
  - ShowAlertDialogWithMultipleButtons, [54](#)
  - ShowAlertDialogWithSingleButton, [54](#)
  - ShowLoginPromptDialog, [55](#)
  - ShowSingleFieldPromptDialogWithPlainText, [54](#)
  - ShowSingleFieldPromptDialogWithSecuredText, [55](#)
  - SingleFieldPromptCompletion, [55](#)
- VoxelBusters::NativePlugins::Utility
  - GetBundleIdentifier, [57](#)
  - GetBundleVersion, [57](#)
  - GetUUID, [56](#)
  - OpenStoreLink, [56](#)
  - SetApplicationIconBadgeNumber, [56](#)
- VoxelBusters::NativePlugins::WebViewMessage
  - Arguments, [58](#)
  - Host, [58](#)
  - SchemeName, [58](#)
- WHATSAPP
  - VoxelBusters::NativePlugins, [9](#)
- WebViewMessage, [57](#)
- YoutubeAPIKey
  - VoxelBusters::NativePlugins::MediaLibrary↔
    - Settings::AndroidSettings, [14](#)