public final class

SmsManager

extends Object

java.lang.Object

↳android.telephony.SmsManager

Class Overview

Manages SMS operations such as sending data, text, and pdu SMS messages. Get this object by calling the static method SmsManager.getDefault().

Summary

Constants		
int	RESULT_ERROR_GENERIC_FAILURE	Generic failure cause
int	RESULT_ERROR_NO_SERVICE	Failed because service is currently unavailable
int	RESULT_ERROR_NULL_PDU	Failed because no pdu provided
int	RESULT_ERROR_RADIO_OFF	Failed because radio was explicitly turned off
int	STATUS_ON_ICC_FREE	Free space (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).
int	STATUS_ON_ICC_READ	Received and read (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).
int	STATUS_ON_ICC_SENT	Stored and sent (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).
int	STATUS_ON_ICC_UNREAD	Received and unread (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).
int	STATUS_ON_ICC_UNSENT	Stored and unsent (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).

Public Methods	
ArrayList<String>	divideMessage(String text) Divide a message text into several fragments, none bigger than the maximum SMS message siz
static SmsManager	getDefault() Get the default instance of the SmsManager
void	sendDataMessage(String destinationAddress, String scAddress, short destinationPort, byte[] da Send a data based SMS to a specific application port.
void	sendMultipartTextMessage(String destinationAddress, String scAddress, ArrayList<String> parts sentIntents, ArrayList<PendingIntent> deliveryIntents) Send a multi-part text based SMS.
void	sendTextMessage(String destinationAddress, String scAddress, String text, PendingIntent sentI Send a text based SMS.

[Expand] Inherited Methods
▸ From class java.lang.Object

Constants

public static final int ***RESULT_ERROR_GENERIC_FAILURE***

Since: API Level 4

Generic failure cause

Constant Value: 1 (0x00000001)

public static final int ***RESULT_ERROR_NO_SERVICE***

Since: API Level 4

Failed because service is currently unavailable

Constant Value: 4 (0x00000004)

public static final int ***RESULT_ERROR_NULL_PDU***

Since: API Level 4

Failed because no pdu provided

Constant Value: 3 (0x00000003)

public static final int ***RESULT_ERROR_RADIO_OFF***

Since: API Level 4

Failed because radio was explicitly turned off

Constant Value: 2 (0x00000002)

public static final int ***STATUS_ON_ICC_FREE***

Since: API Level 4

Free space (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).

Constant Value: 0 (0x00000000)

public static final int ***STATUS_ON_ICC_READ***

Since: API Level 4

Received and read (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).

Constant Value: 1 (0x00000001)

public static final int ***STATUS_ON_ICC_SENT***

Since: API Level 4

Stored and sent (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).

Constant Value: 5 (0x00000005)

public static final int ***STATUS_ON_ICC_UNREAD***

Since: API Level 4

Received and unread (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).

Constant Value: 3 (0x00000003)

public static final int ***STATUS_ON_ICC_UNSENT***

Since: API Level 4

Stored and unsent (TS 51.011 10.5.3 / 3GPP2 C.S0023 3.4.27).

Constant Value: 7 (0x00000007)

Public Methods

public ArrayList<String> ***divideMessage*** *(String text)*

Since: API Level 4

Divide a message text into several fragments, none bigger than the maximum SMS message size.

Parameters

text	the original message. Must not be null.

Returns

- an `ArrayList` of strings that, in order, comprise the original message

public static SmsManager ***getDefault*** *()*

Since: API Level 4

Get the default instance of the SmsManager

Returns

- the default instance of the SmsManager

public void ***sendDataMessage*** *(String destinationAddress, String scAddress, short destinationPort, byte[] data, PendingIntent sentIntent, PendingIntent deliveryIntent)*

Since: API Level 4

Send a data based SMS to a specific application port.

Parameters

destinationAddress	the address to send the message to
scAddress	is the service center address or null to use the current default SMSC
destinationPort	the port to deliver the message to
data	the body of the message to send

sentIntent	if not NULL this `PendingIntent` is broadcast when the message is successfully sent, or failed. The result code will be `Activity.RESULT_OK` for success, or one of these errors:
	`RESULT_ERROR_GENERIC_FAILURE`
	`RESULT_ERROR_RADIO_OFF`
	`RESULT_ERROR_NULL_PDU`
	For `RESULT_ERROR_GENERIC_FAILURE` the sentIntent may include the extra "errorCode" containing a radio technology specific value, generally only useful for troubleshooting.
	The per-application based SMS control checks sentIntent. If sentIntent is NULL the caller will be checked against all unknown applications, which cause smaller number of SMS to be sent in checking period.

deliveryIntent	if not NULL this `PendingIntent` is broadcast when the message is delivered to the recipient. The raw pdu of the status report is in the extended data ("pdu").

Throws

IllegalArgumentException	if destinationAddress or data are empty

public
void ***sendMultipartTextMessage*** *(String destinationAddress, String scAddress, ArrayList<String> parts, ArrayList<PendingIntent> sentIntents, ArrayList<PendingIntent> deliveryIntents)*

Since: API Level 4

Send a multi-part text based SMS. The callee should have already divided the message into correctly sized parts by calling `divideMessage`.

Parameters

destinationAddress	the address to send the message to
scAddress	is the service center address or null to use the current default SMSC
parts	an `ArrayList` of strings that, in order, comprise the original message

sentIntents	if not null, an `ArrayList` of `PendingIntent`s (one for each message part) that is broadcast when the corresponding message part has been sent. The result code will be `Activity.RESULT_OK` for success, or one of these errors:
	`RESULT_ERROR_GENERIC_FAILURE`
	`RESULT_ERROR_RADIO_OFF`

	RESULT_ERROR_NULL_PDU For RESULT_ERROR_GENERIC_FAILURE each sentIntent may include the extra "errorCode" containing a radio technology specific value, generally only useful for troubleshooting. The per-application based SMS control checks sentIntent. If sentIntent is NULL the caller will be checked against all unknown applications, which cause smaller number of SMS to be sent in checking period.
deliveryIntents	if not null, an ArrayList of PendingIntents (one for each message part) that is broadcast when the corresponding message part has been delivered to the recipient. The raw pdu of the status report is in the extended data ("pdu").

Throws

IllegalArgumentException	if destinationAddress or data are empty

public
void ***sendTextMessage*** *(String destinationAddress, String scAddress, String text, PendingIntent sentIntent, PendingIntent deliveryIntent)*

Since: API Level 4

Send a text based SMS.

Parameters

destinationAddress	the address to send the message to
scAddress	is the service center address or null to use the current default SMSC
text	the body of the message to send
sentIntent	if not NULL this PendingIntent is broadcast when the message is successfully sent, or failed. The result code will be Activity.RESULT_OK for success, or one of these errors: RESULT_ERROR_GENERIC_FAILURE RESULT_ERROR_RADIO_OFF RESULT_ERROR_NULL_PDU For RESULT_ERROR_GENERIC_FAILURE the sentIntent may include the extra "errorCode" containing a radio technology specific value, generally only useful for troubleshooting. The per-application based SMS control checks sentIntent. If sentIntent is NULL the caller will be checked against all unknown applications, which cause

	smaller number of SMS to be sent in checking period.
deliveryIntent	if not NULL this `PendingIntent` is broadcast when the message is delivered to the recipient. The raw pdu of the status report is in the extended data ("pdu").

Throws

IllegalArgumentException	if destinationAddress or text are empty