

# NFC Tagger

## 1.0

Erzeugt von Doxygen 1.8.10

Die Jan 5 2016 22:26:58



# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
<b>2</b>	<b>Anwenderdokumentation</b>	<b>3</b>
<b>3</b>	<b>Hierarchie-Verzeichnis</b>	<b>5</b>
3.1	Klassenhierarchie . . . . .	5
<b>4</b>	<b>Klassen-Verzeichnis</b>	<b>7</b>
4.1	Auflistung der Klassen . . . . .	7
<b>5</b>	<b>Klassen-Dokumentation</b>	<b>9</b>
5.1	com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader Klassenreferenz . . . . .	9
5.1.1	Ausführliche Beschreibung . . . . .	9
5.1.2	Dokumentation der Elementfunktionen . . . . .	9
5.1.2.1	getAdressbookData(int position) . . . . .	10
5.1.2.2	readAllContacts(ContentResolver contentResolver) . . . . .	11
5.2	com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookWriter Klassenreferenz . . . . .	11
5.2.1	Ausführliche Beschreibung . . . . .	11
5.2.2	Dokumentation der Elementfunktionen . . . . .	11
5.2.2.1	writeContact(final Context context, final ArrayList< String > cardContent) . . . . .	11
5.3	com.ag.mk.nfccardreadwrite.cardwork.CardReader Klassenreferenz . . . . .	12
5.3.1	Ausführliche Beschreibung . . . . .	12
5.3.2	Dokumentation der Elementfunktionen . . . . .	12
5.3.2.1	readTag(Intent intent) . . . . .	12
5.4	com.ag.mk.nfccardreadwrite.cardwork.CardWriter Klassenreferenz . . . . .	12
5.4.1	Ausführliche Beschreibung . . . . .	12
5.4.2	Dokumentation der Elementfunktionen . . . . .	13
5.4.2.1	createNdefMessage(String content) . . . . .	13
5.4.2.2	writeNdefMessage(Tag tag, NdefMessage ndefMessage) . . . . .	14
5.5	com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog Klassenreferenz . . . . .	14
5.5.1	Ausführliche Beschreibung . . . . .	14
5.5.2	Beschreibung der Konstruktoren und Destruktoren . . . . .	14
5.5.2.1	ContactListDialog(MainActivity mainActivity) . . . . .	14

5.5.3	Dokumentation der Elementfunktionen	14
5.5.3.1	showDialog()	15
5.6	com.ag.mk.nfccardreadwrite.tools.ContactTools Klassenreferenz	15
5.6.1	Ausführliche Beschreibung	15
5.6.2	Beschreibung der Konstruktoren und Destruktoren	15
5.6.2.1	ContactTools(MainActivity mainActivity)	15
5.6.3	Dokumentation der Elementfunktionen	15
5.6.3.1	callContact(String name, final String number)	15
5.6.3.2	mailContact(final String email)	15
5.7	com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity Klassenreferenz	16
5.7.1	Ausführliche Beschreibung	16
5.7.2	Dokumentation der Elementfunktionen	17
5.7.2.1	onCreate(Bundle savedInstanceState)	17
5.7.2.2	onNewIntent(Intent intent)	17
5.7.2.3	onPause()	17
5.7.2.4	onResume()	17
5.8	com.ag.mk.nfccardreadwrite.tools.DataWork Klassenreferenz	18
5.8.1	Ausführliche Beschreibung	18
5.8.2	Dokumentation der Elementfunktionen	18
5.8.2.1	readMultiLineFile(String dataName)	18
5.8.2.2	readSingleLineFile(String dataName)	18
5.8.2.3	writeMultiLineFile(String dataName, List< String > informationList)	18
5.8.2.4	writeSingleLineFile(String dataName, String information)	19
5.9	com.ag.mk.nfccardreadwrite.activity.MainActivity Klassenreferenz	19
5.9.1	Ausführliche Beschreibung	20
5.9.2	Dokumentation der Elementfunktionen	20
5.9.2.1	createNdefMessage(NfcEvent event)	20
5.9.2.2	onCreate(Bundle savedInstanceState)	21
5.9.2.3	onDestroy()	21
5.9.2.4	onInit(int status)	21
5.9.2.5	onNewIntent(Intent intent)	21
5.9.2.6	onPause()	21
5.9.2.7	onResume()	21
5.9.2.8	setVCardInformationOnMainScreen(String vCardInformation)	22
5.10	com.ag.mk.nfccardreadwrite.tools.NfcTools Klassenreferenz	23
5.10.1	Ausführliche Beschreibung	23
5.10.2	Beschreibung der Konstruktoren und Destruktoren	23
5.10.2.1	NfcTools(MainActivity mainActivity, NfcAdapter nfcAdapter)	23
5.10.3	Dokumentation der Elementfunktionen	23
5.10.3.1	checkNFC()	23

5.10.3.2	checkNFCSupport()	23
5.11	com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog Klassenreferenz	24
5.11.1	Ausführliche Beschreibung	24
5.11.2	Beschreibung der Konstruktoren und Destrukturen	24
5.11.2.1	SettingsDialog(MainActivity mainActivity)	24
5.12	com.ag.mk.nfccardreadwrite.tools.VCardFormatTools Klassenreferenz	25
5.12.1	Ausführliche Beschreibung	25
5.12.2	Dokumentation der Elementfunktionen	25
5.12.2.1	extractCardInformation(String[] cardContent)	25
5.12.2.2	getFormattedVCardString(String userName, String mobileNumber, String home← Number, String eMail)	25
5.13	com.ag.mk.nfccardreadwrite.addons.Vibration Klassenreferenz	26
5.13.1	Ausführliche Beschreibung	26
5.13.2	Beschreibung der Konstruktoren und Destrukturen	26
5.13.2.1	Vibration(Vibrator vibrator)	26
5.13.3	Dokumentation der Elementfunktionen	26
5.13.3.1	vibrate()	26
5.14	com.ag.mk.nfccardreadwrite.addons.Voice Klassenreferenz	27
5.14.1	Ausführliche Beschreibung	27
5.14.2	Beschreibung der Konstruktoren und Destrukturen	27
5.14.2.1	Voice(TextToSpeech textToSpeech)	27
5.14.3	Dokumentation der Elementfunktionen	27
5.14.3.1	speakOut(String message)	27
<b>Index</b>		<b>29</b>



## Kapitel 1

### Einführung

...

..





## Kapitel 2

# Anwenderdokumentation

- Ich wiederhole mich nicht
- Ich wiederhole mich nicht



## Kapitel 3

# Hierarchie-Verzeichnis

### 3.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader . . . . .	9
com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookWriter . . . . .	11
com.ag.mk.nfccardreadwrite.cardwork.CardReader . . . . .	12
com.ag.mk.nfccardreadwrite.cardwork.CardWriter . . . . .	12
com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog . . . . .	14
com.ag.mk.nfccardreadwrite.tools.ContactTools . . . . .	15
CreateNdefMessageCallback	
com.ag.mk.nfccardreadwrite.activity.MainActivity . . . . .	19
com.ag.mk.nfccardreadwrite.tools.DataWork . . . . .	18
com.ag.mk.nfccardreadwrite.tools.NfcTools . . . . .	23
OnInitListener	
com.ag.mk.nfccardreadwrite.activity.MainActivity . . . . .	19
com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog . . . . .	24
com.ag.mk.nfccardreadwrite.tools.VCardFormatTools . . . . .	25
com.ag.mk.nfccardreadwrite.addons.Vibration . . . . .	26
com.ag.mk.nfccardreadwrite.addons.Voice . . . . .	27
AppCompatActivity	
com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity . . . . .	16
com.ag.mk.nfccardreadwrite.activity.MainActivity . . . . .	19



## Kapitel 4

# Klassen-Verzeichnis

### 4.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

<a href="#">com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader</a>	9
<a href="#">com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookWriter</a>	11
<a href="#">com.ag.mk.nfccardreadwrite.cardwork.CardReader</a>	12
<a href="#">com.ag.mk.nfccardreadwrite.cardwork.CardWriter</a>	12
<a href="#">com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog</a>	14
<a href="#">com.ag.mk.nfccardreadwrite.tools.ContactTools</a>	15
<a href="#">com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity</a>	16
<a href="#">com.ag.mk.nfccardreadwrite.tools.DataWork</a>	18
<a href="#">com.ag.mk.nfccardreadwrite.activity.MainActivity</a>	19
<a href="#">com.ag.mk.nfccardreadwrite.tools.NfcTools</a>	23
<a href="#">com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog</a>	24
<a href="#">com.ag.mk.nfccardreadwrite.tools.VCardFormatTools</a>	25
<a href="#">com.ag.mk.nfccardreadwrite.addons.Vibration</a>	26
<a href="#">com.ag.mk.nfccardreadwrite.addons.Voice</a>	27

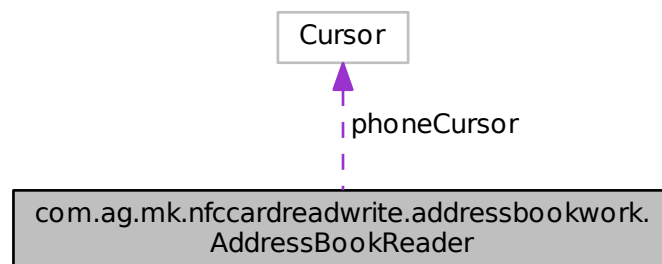


## Kapitel 5

# Klassen-Dokumentation

### 5.1 com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader Klassenreferenz

Zusammengehörigkeiten von com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader:



#### Öffentliche Methoden

- **AddressBookReader** ([MainActivity](#) mainActivity)
- void [getAdressbookData](#) (int position)
- void [readAllContacts](#) (ContentResolver contentResolver)
- ArrayList< String > **getListItems** ()

#### 5.1.1 Ausführliche Beschreibung

Diese Klasse beinhaltet alle Methoden zum Auslesen eines Kontaktes aus dem Adressbuch.

Autor

Klaus Steinhauer, Marko Klepatz

#### 5.1.2 Dokumentation der Elementfunktionen

5.1.2.1 void com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader.getAdressbookData ( int *position* )

Diese Methode sammelt beim Aufruf alle benötigten Daten zum ausgewählten Kontakt.



## Parameter

<i>position</i>	übergibt die Position aus der Namen-ListView zum lokalisieren des Kontaktes im Adressbuch
-----------------	---

### 5.1.2.2 void com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookReader.readAllContacts ( ContentResolver contentResolver )

Diese Methode liest alle Kontakte im Adressbuch aus und schreibt die Namen in die statische **listItems** Liste aus der **ContactListDialog** Klasse.

## Parameter

<i>contentResolver</i>	
------------------------	--

## Siehe auch

ContactListDialog

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- AddressBookReader.java

## 5.2 com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookWriter Klassenreferenz

### Öffentliche, statische Methoden

- static void [writeContact](#) (final Context context, final ArrayList< String > cardContent)

### 5.2.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methode zum Schreiben von Address-Daten in das Adressbuch.

## Autor

Klaus Steinhauer, Marko Klepatz

### 5.2.2 Dokumentation der Elementfunktionen

#### 5.2.2.1 static void com.ag.mk.nfccardreadwrite.addressbookwork.AddressBookWriter.writeContact ( final Context context, final ArrayList< String > cardContent ) [static]

Diese Methode generiert einen Intent welcher alle zu schreibenden Address-Daten enthält und ruft im Anschluss das Adressbuch über diesen Intent auf und übergibt die Daten.

## Parameter

<i>context</i>	übergibt den Context der rufenden Activity
<i>cardContent</i>	übergibt die zu schreibenden Daten in Form einer Array List

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- AddressBookWriter.java

## 5.3 com.ag.mk.nfccardreadwrite.cardwork.CardReader Klassenreferenz

### Öffentliche, statische Methoden

- static String [readTag](#) (Intent intent)

#### 5.3.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methode zum Auslesen der Kartendaten.

#### Autor

Oliver Friedrich, Marko Klepatz

#### 5.3.2 Dokumentation der Elementfunktionen

##### 5.3.2.1 static String com.ag.mk.nfccardreadwrite.cardwork.CardReader.readTag ( Intent *intent* ) [static]

Diese Methode liest aus dem übergebenem Intent die Daten aus, die sich auf der Karte befinden und wandelt sie in einen String um.

#### Parameter

<i>intent</i>	enthält die Daten die auf der Karte sind
---------------	--

#### Rückgabe

gibt den String mit den Daten auf der Karte zurück

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- CardReader.java

## 5.4 com.ag.mk.nfccardreadwrite.cardwork.CardWriter Klassenreferenz

### Öffentliche Methoden

- **CardWriter** (Context context)
- void [writeNdefMessage](#) (Tag tag, NdefMessage ndefMessage)
- NdefMessage [createNdefMessage](#) (String content)

#### 5.4.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methoden die für das Beschreiben eines NFC Chips benötigt werden.

#### Autor

Oliver Friedrich, Marko Klepatz

## 5.4.2 Dokumentation der Elementfunktionen

### 5.4.2.1 NdefMessage com.ag.mk.nfccardreadwrite.cardwork.CardWriter.createNdefMessage ( String *content* )

Diese Methode generiert die NDEF Message mit dem speziellen Mime Type für diese App.

## Parameter

<i>content</i>	übergibt den Inhalt der auf die Karte geschrieben werden soll
----------------	---

## Rückgabe

gibt die generierte NDEF Message zurück

5.4.2.2 `void com.ag.mk.nfccardreadwrite.cardwork.CardWriter.writeNdefMessage ( Tag tag, NdefMessage ndefMessage )`

Diese Methode schreibt die NDEF Nachricht auf den NFC Chip.

## Parameter

<i>tag</i>	übergibt das Tag-Format
<i>ndefMessage</i>	übergibt die NDEF Nachricht zum beschreiben auf den NFC Chip

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- CardWriter.java

## 5.5 com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog Klassenreferenz

### Öffentliche Methoden

- [ContactListDialog](#) ([MainActivity](#) mainActivity)
- void [showDialog](#) ()

### 5.5.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methoden zum Initialisieren und Anzeigen des Kontakt-Listen-Dialogs.

## Autor

Marko Klepatz

### 5.5.2 Beschreibung der Konstruktoren und Destruktoren

5.5.2.1 `com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog.ContactListDialog ( MainActivity mainActivity )`

Dieser Konstruktor leitet alle Initialisierungen für den [ContactListDialog](#) ein.

## Parameter

<i>mainActivity</i>	übergibt die Klasse MainActivity für den Context der GUI Elemente
---------------------	---

## Siehe auch

MainActivity  
AddressBookReader

### 5.5.3 Dokumentation der Elementfunktionen

## 5.5.3.1 void com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog.showDialog ( )

Diese Methode leitet das Auslesen der Kontakte ein und ruft die Anzeigemethode für den [ContactListDialog](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- ContactListDialog.java

## 5.6 com.ag.mk.nfccardreadwrite.tools.ContactTools Klassenreferenz

### Öffentliche Methoden

- [ContactTools](#) ([MainActivity](#) mainActivity)
- void [mailContact](#) (final String email)
- void [easterEgg](#) (String name)
- void [callContact](#) (String name, final String number)

### 5.6.1 Ausführliche Beschreibung

Diese Klasse beinhaltet verschiedene Methoden um die Kontaktdaten direkt an andere Anwendungen im Gerät zu übergeben und zu starten.

#### Autor

Marko Klepatz

### 5.6.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.6.2.1 com.ag.mk.nfccardreadwrite.tools.ContactTools.ContactTools ( [MainActivity](#) mainActivity )

##### Parameter

<i>mainActivity</i>	übergibt die MainActivity für den Context zum Ausführen von Code der nur auf dieser ausgeführt werden kann
---------------------	--

#### Siehe auch

[MainActivity](#)

### 5.6.3 Dokumentation der Elementfunktionen

#### 5.6.3.1 void com.ag.mk.nfccardreadwrite.tools.ContactTools.callContact ( [String](#) name, final [String](#) number )

Diese Methode generiert einen Intent welcher die Nummer des Kontaktes übergibt und ruft mit diesem direkt das Programm zum Anrufen von Kontakten auf und ruft diesen an.

##### Parameter

<i>name</i>	übergibt den Namen des ausgewählten Kontakts für die Sprachausgabe
<i>number</i>	übergibt die Nummer des ausgewählten Kontakts zum Anrufen

#### 5.6.3.2 void com.ag.mk.nfccardreadwrite.tools.ContactTools.mailContact ( final [String](#) email )

Diese Methode generiert einen Intent der die E-Mail Adresse des aktuell ausgewählten Kontakts übergibt und startet eine Auswahl mit allen potenziell einsetzbaren E-Mail Programmen auf dem Gerät.

## Parameter

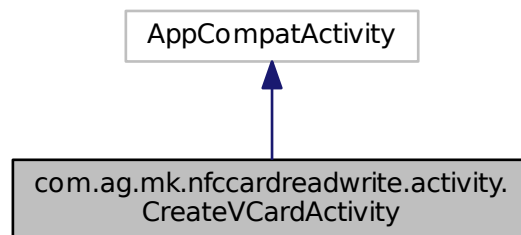
<i>email</i>	übergibt die E-Mail Adresse des aktuell ausgewählten Kontakts
--------------	---

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

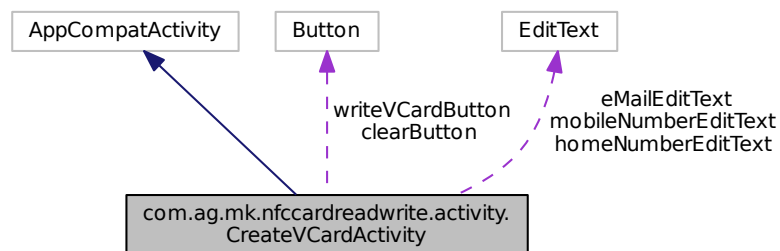
- ContactTools.java

## 5.7 com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity Klassenreferenz

Klassendiagramm für com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity:



Zusammengehörigkeiten von com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity:



### Geschützte Methoden

- void [onCreate](#) (Bundle savedInstanceState)
- void [onNewIntent](#) (Intent intent)
- void [onResume](#) ()
- void [onPause](#) ()

#### 5.7.1 Ausführliche Beschreibung

Diese Activity beinhaltet die Logik zu den GUI Elementen, die benötigt werden um NFC Medien zu beschreiben.

Zusätzlich beinhaltet sie eine Methode zum Einlesen gültiger NFC Medien.

#### Autor

Marko Klepatz, Oliver Friedrich

### 5.7.2 Dokumentation der Elementfunktionen

**5.7.2.1** void com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity.onCreate ( Bundle *savedInstanceState* )  
[protected]

Diese Methode setzt alle relevanten Eigenschaften für die GUI und leitet die Initialisierungen aller Objekte ein.

Zusätzlich nimmt sie den aufrufenden Intent von der MainActivity entgegen und prüft ob dieser Daten zum Beschreiben auf ein NFC Medium enthält und leitet diese weiter, an die GUI Objekte.

#### Parameter

<i>savedInstanceState</i>	
---------------------------	--

#### Siehe auch

NfcAdapter  
CardWriter

**5.7.2.2** void com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity.onNewIntent ( Intent *intent* ) [protected]

Diese Methode empfängt einen von dem Manifest gefilterten Intent, welcher nur dann in dieser ankommt wenn das NFC Medium, was an das Gerät gehalten wird, mit den gültigen Technologien ausgestattet ist.

Die aktuell gültigen Technologien sind in folgenden Dateien einsehbar:

**AndroidManifest.xml**

**tech.xml**

#### Parameter

<i>intent</i>	
---------------	--

**5.7.2.3** void com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity.onPause ( ) [protected]

Diese Methode deaktiviert den NFC Adapter.

**5.7.2.4** void com.ag.mk.nfccardreadwrite.activity.CreateVCardActivity.onResume ( ) [protected]

Diese Methode konfiguriert und aktiviert bei Aufruf den NFC Adapter.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- CreateVCardActivity.java

## 5.8 com.ag.mk.nfccardreadwrite.tools.DataWork Klassenreferenz

### Öffentliche Methoden

- **DataWork** (Context context)

### Öffentliche, statische Methoden

- static String [readSingleLineFile](#) (String dataName)
- static List< String > [readMultiLineFile](#) (String dataName)
- static void [writeSingleLineFile](#) (String dataName, String information)
- static void [writeMultiLineFile](#) (String dataName, List< String > informationList)

### 5.8.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methoden zum Lesen und Schreiben von Dateien für die Anwendung.

#### Autor

Marko Klepatz

### 5.8.2 Dokumentation der Elementfunktionen

#### 5.8.2.1 static List<String> com.ag.mk.nfccardreadwrite.tools.DataWork.readMultiLineFile ( String dataName ) [static]

Diese Methode liest eine Datei mit mehreren Zeilen aus und gibt den Inhalt on Form einer Liste vom Typ String zurück.

##### Parameter

<i>dataName</i>	übergibt den Namen der zu lesenden Datei
-----------------	--

##### Rückgabe

values gibt die Liste mit dem Inhalt jeder Zeile zurück

#### 5.8.2.2 static String com.ag.mk.nfccardreadwrite.tools.DataWork.readSingleLineFile ( String dataName ) [static]

Diese Methode liest eine Datei mit einer Zeile aus und gibt den Inhalt on Form eines Strings zurück.

##### Parameter

<i>dataName</i>	übergibt den Namen der zu lesenden Datei
-----------------	--

##### Rückgabe

gibt den ausgelesenen String zurück

#### 5.8.2.3 static void com.ag.mk.nfccardreadwrite.tools.DataWork.writeMultiLineFile ( String dataName, List< String > informationList ) [static]

Diese Methode schreibt eine mehrzeilige Information in eine Datei mit Umbrüchen



## Parameter

<i>dataName</i>	übergibt den Namen der zu schreibenden Datei
<i>informationList</i>	übergibt die zu schreibenden Informationen

5.8.2.4 static void com.ag.mk.nfccardreadwrite.tools.DataWork.writeSingleLineFile ( String *dataName*, String *information* )  
[static]

Diese Methode schreibt eine einzeilige Information in eine Datei ohne Umbrüche

## Parameter

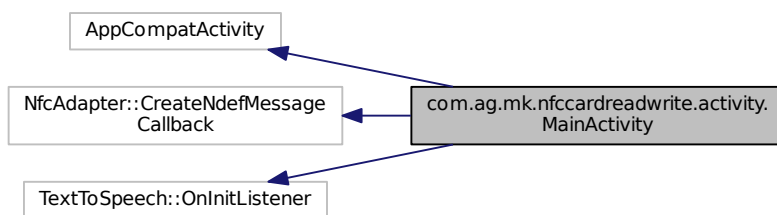
<i>dataName</i>	übergibt den Namen der zu schreibenden Datei
<i>information</i>	übergibt die zu schreibende Information

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

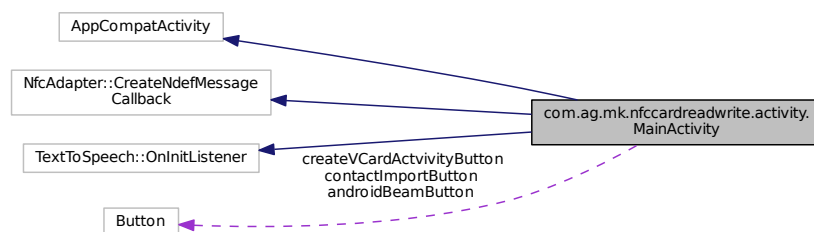
- DataWork.java

## 5.9 com.ag.mk.nfccardreadwrite.activity.MainActivity Klassenreferenz

Klassendiagramm für com.ag.mk.nfccardreadwrite.activity.MainActivity:



Zusammengehörigkeiten von com.ag.mk.nfccardreadwrite.activity.MainActivity:



### Öffentliche Methoden

- void [setVCardInformationOnMainScreen](#) (String vCardInformation)
- NdefMessage [createNdefMessage](#) (NfcEvent event)

- void [onInit](#) (int status)
- boolean [onCreateOptionsMenu](#) (Menu menu)
- boolean [onOptionsItemSelected](#) (MenuItem item)

### Statische öffentliche Attribute

- static final String **TAG** = "Nfc Card App"

### Geschützte Methoden

- void [onCreate](#) (Bundle savedInstanceState)
- void [onNewIntent](#) (Intent intent)
- void [onResume](#) ()
- void [onPause](#) ()
- void [onDestroy](#) ()

## 5.9.1 Ausführliche Beschreibung

Diese Activity beinhaltet die Logik zu den GUI Elementen die benötigt werden um NFC Medien zu lesen und anzuzeigen.

Zusätzlich werden hier das Anzeigen von Kontakten aus dem Adressbuch und das Importieren von neuen Kontakten, sowie die Android Beam-Funktion, eingeleitet. Ebenso wird hier auch die [CreateVCardActivity](#) gerufen.

Weiterführend wird immer diese Activity gerufen, wenn eine für diese App gültige NFC Technologie oder der spezielle Mime Type dieser App erkannt wird. (Der Mime Type ist in der Klasse **CardWriter** einsehbar)

Siehe auch

CardWriter

Autor

Marko Klepatz, Oliver Friedrich

## 5.9.2 Dokumentation der Elementfunktionen

### 5.9.2.1 NdefMessage com.ag.mk.nfccardreadwrite.activity.MainActivity.createNdefMessage ( NfcEvent event )

Diese Callback Methode gibt die zu sendende NDEF Message für den Android NFC Beamer zurück.

Parameter

<i>event</i>	
--------------	--

Rückgabe

gibt die zu sendende NDEF Message für den Android NFC Beamer zurück.

**5.9.2.2 void com.ag.mk.nfccardreadwrite.activity.MainActivity.onCreate ( Bundle savedInstanceState ) [protected]**

Diese Methode setzt alle relevanten Eigenschaften für die GUI und leitet die Initialisierungen aller Objekte ein.

Zusätzlich nimmt diese Methode einen von dem Manifest gefilterten Intent entgegen welcher nur dann in dieser ankommt wenn das NFC Medium, was an das Gerät gehalten wird, mit den gültigen Technologien ausgestattet ist bzw. den Mime Type für diese App besitzt.

Die aktuell gültigen Technologien sind in folgenden Dateien einsehbar:

**AndroidManifest.xml**

**tech.xml**

Parameter

<i>savedInstanceState</i>	
---------------------------	--

**5.9.2.3 void com.ag.mk.nfccardreadwrite.activity.MainActivity.onDestroy ( ) [protected]**

Diese Methode beendet beim Aufruf die TextToSpeech Vorgang/Funktion.

**5.9.2.4 void com.ag.mk.nfccardreadwrite.activity.MainActivity.onInit ( int status )**

Diese Methode weist der TextToSpeech Klasse die Sprache zu mit welcher dann die Sprachausgabe erfolgt. In diesem Fall wird immer die Standardsprache des Geräts verwendet.

Parameter

<i>status</i>	
---------------	--

**5.9.2.5 void com.ag.mk.nfccardreadwrite.activity.MainActivity.onNewIntent ( Intent intent ) [protected]**

Diese Methode empfängt einen von dem Manifest gefilterten Intent, welcher nur dann in dieser ankommt wenn das NFC Medium, was an das Gerät gehalten wird, mit den gültigen Technologien ausgestattet ist.

Die aktuell gültigen Technologien sind in folgenden Dateien einsehbar:

**AndroidManifest.xml**

**tech.xml**

Parameter

<i>intent</i>	
---------------	--

**5.9.2.6 void com.ag.mk.nfccardreadwrite.activity.MainActivity.onPause ( ) [protected]**

Diese Methode deaktiviert den NFC Adapter.

**5.9.2.7 void com.ag.mk.nfccardreadwrite.activity.MainActivity.onResume ( ) [protected]**

Diese Methode konfiguriert und aktiviert bei Aufruf den NFC Adapter.

5.9.2.8 void com.ag.mk.nfccardreadwrite.activity.MainActivity.setVCardInformationOnMainScreen ( String *vCardInformation* )

Diese Methode lädt die Daten auf die **vCardListView**.

## Parameter

<i>vCard</i> ↔ <i>Information</i>	übergibt die Informationen die sich auf dem NFC Medium befinden.
--------------------------------------	--

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- MainActivity.java

## 5.10 com.ag.mk.nfccardreadwrite.tools.NfcTools Klassenreferenz

### Öffentliche Methoden

- [NfcTools](#) ([MainActivity](#) mainActivity, [NfcAdapter](#) nfcAdapter)
- void [checkNFC](#) ()
- void [checkNFCSupport](#) ()

### 5.10.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methoden zum Überprüfen ob die NFC Technologie vorhanden und aktiviert ist.

#### Autor

Marko Klepatz

### 5.10.2 Beschreibung der Konstruktoren und Destruktoren

5.10.2.1 com.ag.mk.nfccardreadwrite.tools.NfcTools.NfcTools ( [MainActivity](#) mainActivity, [NfcAdapter](#) nfcAdapter )

## Parameter

<i>mainActivity</i>	übergibt die MainActivity für den Context
<i>nfcAdapter</i>	übergibt den NfcAdapter zur Überprüfung

#### Siehe auch

MainActivity  
NfcAdapter

### 5.10.3 Dokumentation der Elementfunktionen

5.10.3.1 void com.ag.mk.nfccardreadwrite.tools.NfcTools.checkNFC ( )

Diese Methode überprüft ob die NFC Technologie aktiviert ist und gibt eine Warnung in Form einer Toast Nachricht aus wenn sie deaktiviert ist.

5.10.3.2 void com.ag.mk.nfccardreadwrite.tools.NfcTools.checkNFCSupport ( )

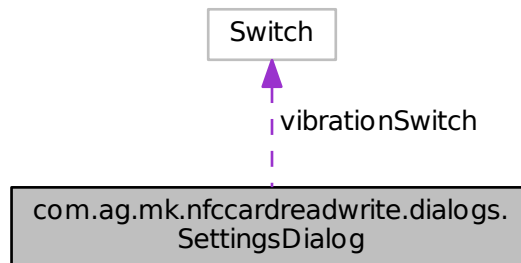
Diese Methode überprüft ob die NFC Technologie auf dem Gerät vorhanden ist und gibt eine Toast Nachricht aus wenn sie es nicht ist und beendet die App.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- NfcTools.java

## 5.11 com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog Klassenreferenz

Zusammengehörigkeiten von com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog:



### Öffentliche Methoden

- [SettingsDialog \(MainActivity mainActivity\)](#)
- void **showDialog** ()
- Switch **getVibrationSwitch** ()
- Switch **getVoiceSwitch** ()

#### 5.11.1 Ausführliche Beschreibung

Diese Klasse beinhaltet alle Methoden zum Generieren und Anzeigen des Einstellungs-Dialogs. Weiterführend werden hier auch alle Einstellungen direkt in Dateien auf dem Handy geschrieben.

Siehe auch

DataWork

Autor

Marko Klepatz

#### 5.11.2 Beschreibung der Konstruktoren und Destruktoren

##### 5.11.2.1 com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog.SettingsDialog ( MainActivity mainActivity )

Dieser Konstruktor leitet alle Initialisierungen für den [SettingsDialog](#) ein.

Parameter

<i>mainActivity</i>	übergibt die Klasse MainActivity für den Context der GUI Elemente
---------------------	---

Siehe auch

MainActivity

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- SettingsDialog.java

## 5.12 com.ag.mk.nfccardreadwrite.tools.VCardFormatTools Klassenreferenz

### Öffentliche, statische Methoden

- static ArrayList< String > [extractCardInformation](#) (String[] cardContent)
- static String [getFormattedVCardString](#) (String userName, String mobileNumber, String homeNumber, String eMail)

### 5.12.1 Ausführliche Beschreibung

Diese Klasse beinhaltet die Methoden, die für die V-Card Formatierung und das Extrahieren der Informationen aus dem V-Card Format benötigt werden.

#### Autor

Marko Klepatz, Klaus Steinhauer

### 5.12.2 Dokumentation der Elementfunktionen

#### 5.12.2.1 static ArrayList<String> com.ag.mk.nfccardreadwrite.tools.VCardFormatTools.extractCardInformation ( String[] cardContent ) [static]

Diese Methode extrahiert die Informationen aus einem V-Card String und bringt sie in ein für diese App sinnvolles Ausgabeformat.

#### Parameter

<i>cardContent</i>	übergibt die rohen Karten-Daten
--------------------	---------------------------------

#### Rückgabe

vCardInformationList gibt eine Liste mit allen extrahierten Daten zurück

#### 5.12.2.2 static String com.ag.mk.nfccardreadwrite.tools.VCardFormatTools.getFormattedVCardString ( String userName, String mobileNumber, String homeNumber, String eMail ) [static]

Diese Methode formatiert die Roh-Adress-Daten in einen V-Card-Format String.

#### Parameter

<i>userName</i>	übergibt den Namen des gewählten Kontaktes
<i>mobileNumber</i>	übergibt die Mobilnummer des gewählten Kontaktes
<i>homeNumber</i>	übergibt die Festnetznummer des gewählten Kontaktes
<i>eMail</i>	übergibt die E-Mail Adresse des gewählten Kontaktes

#### Rückgabe

gibt den V-Card-Format String zurück

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- VCardFormatTools.java

## 5.13 com.ag.mk.nfccardreadwrite.addons.Vibration Klassenreferenz

### Öffentliche Methoden

- [Vibration](#) (Vibrator vibrator)

### Öffentliche, statische Methoden

- static void **setVibration** (boolean vibration)
- static boolean **isVibration** ()
- static void [vibrate](#) ()

#### 5.13.1 Ausführliche Beschreibung

Diese Klasse ist für die Vibrationsfunktion, insofern aktiviert, für alle Klassen und Activities zuständig.

Sie muss einmal über ihren Konstruktor initialisiert werden und ist dann für alle Klassen, durch ihren statischen Charakter verfügbar.

Vibrator ist die zu initialisierende Klasse mit der die Vibrationsfunktion ausgeführt werden kann.

#### Autor

Marko Klepatz

#### 5.13.2 Beschreibung der Konstruktoren und Destruktoren

##### 5.13.2.1 com.ag.mk.nfccardreadwrite.addons.Vibration.Vibration ( *Vibrator vibrator* )

Dieser Konstruktor nimmt die in der MainActivity initialisierte Vibrator Klasse entgegen, die hier benötigt wird um die [Vibration](#) auszuführen.

#### Parameter

<i>vibrator</i>	übergibt die benötigte Vibrator Klasse zum Ausführen der <a href="#">Vibration</a>
-----------------	--

#### 5.13.3 Dokumentation der Elementfunktionen

##### 5.13.3.1 static void com.ag.mk.nfccardreadwrite.addons.Vibration.vibrate ( ) [static]

Diese Methode ist beim Aufruf dafür zuständig, dass das Gerät 25 Millisekunden vibriert.

Zusätzlich wird geprüft ob die Variable **vibration** auf true oder false gestellt ist.

true: vibriert

false: vibriert nicht

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:



- Vibration.java

## 5.14 com.ag.mk.nfccardreadwrite.addons.Voice Klassenreferenz

### Öffentliche Methoden

- [Voice](#) (TextToSpeech textToSpeech)

### Öffentliche, statische Methoden

- static void [speakOut](#) (String message)
- static boolean **isSound** ()
- static void **setSound** (boolean sound)

### 5.14.1 Ausführliche Beschreibung

Diese Klasse ist für die Sprachausgabe, insofern aktiviert, für alle Klassen und Activities zuständig.

Sie muss einmal über ihren Konstruktor initialisiert werden und ist dann für alle Klassen, durch ihren statischen Charakter verfügbar.

Vibrator ist die zu initialisierende Klasse mit der die Vibrationsfunktion ausgeführt werden kann.

#### Autor

Marko Klepatz

### 5.14.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.14.2.1 com.ag.mk.nfccardreadwrite.addons.Voice.Voice ( TextToSpeech textToSpeech )

Dieser Konstruktor nimmt die in der MainActivity initialisierte TextToSpeech Klasse entgegen, die hier benötigt wird um die Sprachausgabe auszuführen.

#### Parameter

<i>textToSpeech</i>	übergibt die benötigte TextToSpeech Klasse zum Ausführen der Sprachausgabe
---------------------	--

### 5.14.3 Dokumentation der Elementfunktionen

#### 5.14.3.1 static void com.ag.mk.nfccardreadwrite.addons.Voice.speakOut ( String message ) [static]

Diese Methode ist beim Aufruf für die Sprachausgabe zuständig.

Zusätzlich wird geprüft ob die Variable **sound** auf true oder false gestellt ist.

true: Sprachausgabe wird ausgeführt

false: Sprachausgabe wird nicht ausgeführt

#### Parameter

<i>message</i>	übergibt die Nachricht, welche über die Sprachausgabe ausgegeben wird
----------------	---

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Voice.java

# Index

callContact  
    com::ag::mk::nfccardreadwrite::tools::Contact↔  
        Tools, 15

checkNFC  
    com::ag::mk::nfccardreadwrite::tools::NfcTools, 23

checkNFCSupport  
    com::ag::mk::nfccardreadwrite::tools::NfcTools, 23

com.ag.mk.nfccardreadwrite.activity.CreateVCard↔  
    Activity, 16

com.ag.mk.nfccardreadwrite.activity.MainActivity, 19

com.ag.mk.nfccardreadwrite.addons.Vibration, 26

com.ag.mk.nfccardreadwrite.addons.Voice, 27

com.ag.mk.nfccardreadwrite.addressbookwork.↔  
    AddressBookReader, 9

com.ag.mk.nfccardreadwrite.addressbookwork.↔  
    AddressBookWriter, 11

com.ag.mk.nfccardreadwrite.cardwork.CardReader, 12

com.ag.mk.nfccardreadwrite.cardwork.CardWriter, 12

com.ag.mk.nfccardreadwrite.dialogs.ContactListDialog,  
    14

com.ag.mk.nfccardreadwrite.dialogs.SettingsDialog, 24

com.ag.mk.nfccardreadwrite.tools.ContactTools, 15

com.ag.mk.nfccardreadwrite.tools.DataWork, 18

com.ag.mk.nfccardreadwrite.tools.NfcTools, 23

com.ag.mk.nfccardreadwrite.tools.VCardFormatTools,  
    25

com::ag::mk::nfccardreadwrite::activity::CreateVCard↔  
    Activity  
        onCreate, 17  
        onNewIntent, 17  
        onPause, 17  
        onResume, 17

com::ag::mk::nfccardreadwrite::activity::MainActivity  
    createNdefMessage, 20  
    onCreate, 20  
    onDestroy, 21  
    onInit, 21  
    onNewIntent, 21  
    onPause, 21  
    onResume, 21  
    setVCardInformationOnMainScreen, 21

com::ag::mk::nfccardreadwrite::addons::Vibration  
    vibrate, 26  
    Vibration, 26

com::ag::mk::nfccardreadwrite::addons::Voice  
    speakOut, 27  
    Voice, 27

com::ag::mk::nfccardreadwrite::addressbookwork::↔  
    AddressBookReader

    getAdressbookData, 9

    readAllContacts, 11

com::ag::mk::nfccardreadwrite::addressbookwork::↔  
    AddressBookWriter

    writeContact, 11

com::ag::mk::nfccardreadwrite::cardwork::CardReader  
    readTag, 12

com::ag::mk::nfccardreadwrite::cardwork::CardWriter  
    createNdefMessage, 13  
    writeNdefMessage, 14

com::ag::mk::nfccardreadwrite::dialogs::ContactList↔  
    Dialog  
        ContactListDialog, 14  
        showDialog, 14

com::ag::mk::nfccardreadwrite::dialogs::SettingsDialog  
    SettingsDialog, 24

com::ag::mk::nfccardreadwrite::tools::ContactTools  
    callContact, 15  
    ContactTools, 15  
    mailContact, 15

com::ag::mk::nfccardreadwrite::tools::DataWork  
    readMultiLineFile, 18  
    readSingleLineFile, 18  
    writeMultiLineFile, 18  
    writeSingleLineFile, 19

com::ag::mk::nfccardreadwrite::tools::NfcTools  
    checkNFC, 23  
    checkNFCSupport, 23  
    NfcTools, 23

com::ag::mk::nfccardreadwrite::tools::VCardFormat↔  
    Tools  
        extractCardInformation, 25  
        getFormattedVCardString, 25

ContactListDialog  
    com::ag::mk::nfccardreadwrite::dialogs::Contact↔  
        ListDialog, 14

ContactTools  
    com::ag::mk::nfccardreadwrite::tools::Contact↔  
        Tools, 15

createNdefMessage  
    com::ag::mk::nfccardreadwrite::activity::Main↔  
        Activity, 20  
    com::ag::mk::nfccardreadwrite::cardwork::Card↔  
        Writer, 13

extractCardInformation  
    com::ag::mk::nfccardreadwrite::tools::VCard↔  
        FormatTools, 25

getAdressbookData

com::ag::mk::nfccardreadwrite::addressbookwork↔  
     ::AddressBookReader, 9  
 getFormattedVCardString  
     com::ag::mk::nfccardreadwrite::tools::VCard↔  
         FormatTools, 25  
  
 mailContact  
     com::ag::mk::nfccardreadwrite::tools::Contact↔  
         Tools, 15  
  
 NfcTools  
     com::ag::mk::nfccardreadwrite::tools::NfcTools, 23  
  
 onCreate  
     com::ag::mk::nfccardreadwrite::activity::CreateV↔  
         CardActivity, 17  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 20  
 onDestroy  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 21  
 onInit  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 21  
 onNewIntent  
     com::ag::mk::nfccardreadwrite::activity::CreateV↔  
         CardActivity, 17  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 21  
 onPause  
     com::ag::mk::nfccardreadwrite::activity::CreateV↔  
         CardActivity, 17  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 21  
 onResume  
     com::ag::mk::nfccardreadwrite::activity::CreateV↔  
         CardActivity, 17  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 21  
  
 readAllContacts  
     com::ag::mk::nfccardreadwrite::addressbookwork↔  
         ::AddressBookReader, 11  
 readMultiLineFile  
     com::ag::mk::nfccardreadwrite::tools::DataWork,  
         18  
 readSingleLineFile  
     com::ag::mk::nfccardreadwrite::tools::DataWork,  
         18  
 readTag  
     com::ag::mk::nfccardreadwrite::cardwork::Card↔  
         Reader, 12  
  
 setVCardInformationOnMainScreen  
     com::ag::mk::nfccardreadwrite::activity::Main↔  
         Activity, 21  
 SettingsDialog  
     com::ag::mk::nfccardreadwrite::dialogs::Settings↔  
         Dialog, 24  
  
 showDialog  
     com::ag::mk::nfccardreadwrite::dialogs::Contact↔  
         ListDialog, 14  
 speakOut  
     com::ag::mk::nfccardreadwrite::addons::Voice, 27  
  
 vibrate  
     com::ag::mk::nfccardreadwrite::addons::Vibration,  
         26  
 Vibration  
     com::ag::mk::nfccardreadwrite::addons::Vibration,  
         26  
 Voice  
     com::ag::mk::nfccardreadwrite::addons::Voice, 27  
  
 writeContact  
     com::ag::mk::nfccardreadwrite::addressbookwork↔  
         ::AddressBookWriter, 11  
 writeMultiLineFile  
     com::ag::mk::nfccardreadwrite::tools::DataWork,  
         18  
 writeNdefMessage  
     com::ag::mk::nfccardreadwrite::cardwork::Card↔  
         Writer, 14  
 writeSingleLineFile  
     com::ag::mk::nfccardreadwrite::tools::DataWork,  
         19