

## UNIVERSITATEA TITU MAIORESCU

FACULTATEA: INFORMATICĂ

DEPARTAMENT: INFORMATICĂ

Programa de studii: INFORMATICĂ

DISCIPLINA: INTELIGENȚĂ ARTIFICIALĂ

**IA - Testul de evaluare nr. 15****NFC pentru telefonie mobilă**

Grupa	Numele și prenumele	Semnătură student	Notă evaluare

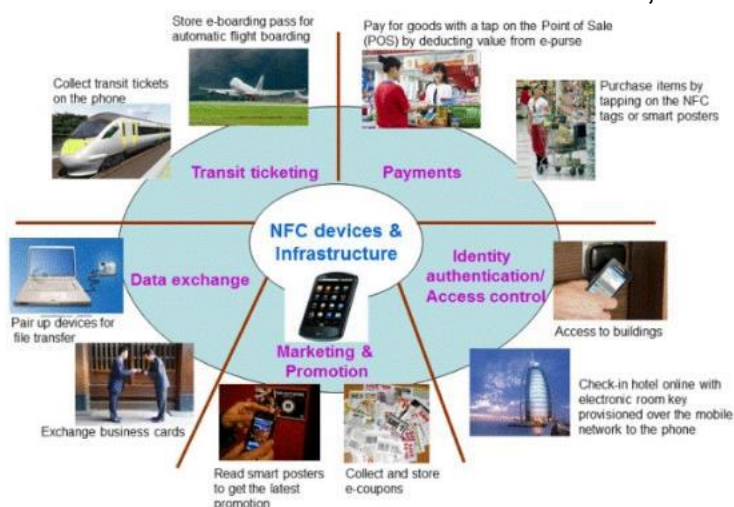
Data: \_\_\_\_ / \_\_\_\_ / \_\_\_\_

Conf.dr.ing.

Lucian Ștefăniță GRIGORE

Ș.L.dr.ing.

Dan-Laurențiu GRECU



## Cuprins

1.	NOȚIUNI DE BAZĂ NFC .....	3
1.1	Sistemul de Tag Dispatch .....	3
1.1.1	Cum sunt mapate tag-urile NFC ca MIME și URI-uri .....	3
1.1.2	Cum sunt expediate Tag-urile NFC la Aplicații .....	5
2.	SOLICITAREA de ACCES a NFC în ANDROIDMANIFEST .....	7
3.	FILTRAREA INTENȚIILOR pentru NFC.....	8
3.1	ACTION_NDEF_DISCOVERED .....	8
3.2	ACTION_TECH_DISCOVERED .....	8
3.3	ACTION_TAG_DISCOVERED.....	10
3.4	Obținerea de informații despre intenții .....	10
4.	CREAREA de TIPURI COMUNE de ÎNREGISTRĂRI NDEF.....	11
4.1	TNF_ABSOLUTE_URI.....	11
4.2	TNF_MIME_MEDIA.....	11
4.3	TNF_WELL_KNOWN cu RTD_TEXT .....	12
4.4	TNF_WELL_KNOWN cu RTD_URI.....	12
4.5	TNF_EXTERNAL_TYPE.....	13
4.6	Înregistrarea aplicațiilor Android .....	14
5.	TRANSMITEREA MESAJELOR NDEF către alte DISPOZITIVE .....	16

## 1. NOȚIUNI DE BAZĂ NFC

Sarcinile de bază NFC ce vor fi efectuate de un Android sunt descrise în continuare. Se explică modul în care pentru a trimite și primi date NFC sub formă de mesaje NDEF și descrie cadru API-urilor Android care acceptă aceste caracteristici. Pentru mai multe subiecte avansate, inclusiv o discuție de lucru cu date non-NDEF, vezi (**Error! Reference source not found.**).

Există două cazuri majore de utilizări atunci când se lucrează cu date NDEF și Android:

- citirea datelor dintr-o etichetă NDEF NFC;
- emiterea de mesaje NDEF de la un dispozitiv la altul cu (Android Beam™ - 0).

Citirea datelor dintr-o etichetă NDEF NFC este efectuată cu *sistemul tag-ul de expediere*, care analizează etichetele NFC descoperite, clasifică în mod corespunzător datele, și începe o aplicație care este interesată de datele clasificate. O cerere care vrea să se ocupe de tag-ul NFC scanat poate *declara un filtru intenție* și să solicite să se ocupe de datele respective.

Caracteristica Android Beam™ este că permite unui dispozitiv să trimită un mesaj NDEF pe un alt dispozitiv prin atingerea fizică a dispozitivelor respective. Această interacțiune oferă o modalitate mai ușoară de a trimite date decât alte tehnologii wireless, cum ar fi Bluetooth, deoarece cu NFC, nu este necesar un dispozitiv descoperire sau împerecherea manuală. Conexiunea este pornită automat atunci când două dispozitive se asociază în aceeași gamă. Android Beam™ este disponibil printr-un set de API-uri NFC, astfel încât orice aplicație poate transmite informații între dispozitive. De exemplu, contactele, browser-ul, și aplicațiile YouTube folosesc Android Beam™ pentru a partaja contactele, paginile web, și clipurile video cu alte dispozitive.

### 1.1 Sistemul de Tag Dispatch

Dispozitivele cu Android sunt, de obicei în căutarea de tag-uri NFC când ecranul este deblocat, cu excepția cazului în NFC este dezactivată în meniul Settings al dispozitivului. Când un dispozitiv Android descoperă o etichetă NFC, comportamentul dorit este de a avea activitatea cea mai adecvată cu intenția de a interacționa cu un utilizator pentru aplicația căutată.

Deoarece dispozitivele pot scana etichete NFC la un interval foarte scurt, este probabil ca utilizatorii să efectueze selecția manuală a unei activități, ceea ce i-ar obliga să mute dispozitivul mai departe de etichetă și astfel conexiunea să se întrerupă. De aceea ar trebui ca dezvoltarea activității să se ocupe numai de etichetele NFC pentru a preveni ca Activity Chooser să apară.

În acest scop, Android oferă un sistem special de expediere tag-uri care analizează etichetele NFC scanate, le analizează, și încearcă să localizeze aplicații care sunt compatibile cu datele scanate. Acest lucru se realizează prin:

1. analiza tag-ului NFC și apoi configurând tipul de MIME sau de URI care identifică sarcina utilă din tag;
2. încapsularea de tip MIME sau URI și sarcina utilă într-o singură iterație. Aceste prime două etape sunt descrise în *How NFC tags are mapped to MIME types and URIs*.
3. Începerea unei activități pe baza intenției declarate. Acest lucru este descris *How NFC Tags are Dispatched to Applications*.

#### 1.1.1 Cum sunt mapate tag-urile NFC ca MIME și URI-uri

Înainte de a începe scrierea de aplicații NFC, este important să se înțeleagă diferitele tipuri de etichete NFC, modul în care sistemul tag de expediere analizează etichetele NFC, și cum se oprește expedierea tag-ului atunci când detectează un mesaj NDEF. Tag-urile NFC vin într-o gamă largă de tehnologii și pot avea, de asemenea, datele scrise în mai multe moduri diferite. Android are cel mai mare suport pentru aplicarea standardului NDEF, care este definit de către NFC Forum<sup>1</sup>.

---

<sup>1</sup> <http://nfc-forum.org/>

Datele NDEF sunt încapsulate într-un mesaj (NdefMessage<sup>2</sup>) care conține una sau mai multe înregistrări (NdefRecord<sup>3</sup>). Fiecare înregistrare NDEF trebuie să fie bine formată, conform caietului de sarcini pentru tipul de înregistrare care se dorește a fi creat. Suportul Android, este de asemenea compatibil cu alte tipuri de tag-uri care nu conțin date NDEF, car cu care se poate lucra folosind clasele din pachetul android.nfc.tech<sup>4</sup>. Lucrul cu aceste tag-uri pentru alte tipuri de scriere implică propria stivă de protocol pentru a comunica cu tag-urile respective, așa că se recomandă utilizarea NDEF atunci când este posibil pentru a ușura dezvoltarea și obținerea unui suport maxim pentru dispozitivele cu Android.

**Notă:** Pentru a descărca specificațiile complete de NDEF, se recomandă site-urile [Download NFC Forum Specificatii](#)<sup>5</sup> și [Crearea tipuri comune de înregistrări NDEF](#)<sup>6</sup> pentru exemple de cum eă pot construi înregistrările NDEF.

În următoarele secțiuni se descrie mai detaliat modul în care se ocupă de tag-urile formate NDEF, Android. Când un dispozitiv Android scanează o etichetă NFC care conține date NDEF formate, se analizează mesajul și se încearcă identificarea datelor de tip MIME sau URI. Pentru a face acest lucru, sistemul citește prima [NdefRecord](#) din interiorul [NdefMessage](#) pentru a determina modul de interpretare a întregului mesaj NDEF (un mesaj NDEF poate avea mai multe înregistrări NDEF). Într-un mesaj NDEF bine formatat, primul [NdefRecord](#) conține următoarele câmpuri:

- **3-bit TNF (Tip Nume Format)**

Indică modul de interpretare a domeniului de tip lungime variabilă. Valorile valide sunt descrise în descrie în tabelul 1.

- **Tipul de lungime variabilă**

Descrie tipul de înregistrare. Dacă utilizați TNF\_WELL\_KNOWN, utilizați acest câmp pentru a specifica Record Type Definition (CDT). Valorile valide CDT sunt descrise în tabelul 2.

- **ID lungime variabilă**

Un identificator unic pentru înregistrare. Acest câmp nu este folosit de multe ori, dar dacă este necesar pentru a identifica în mod unic o etichetă, se poate crea un ID pentru eticheta respectivă.

- **Lungime variabilă sarcină utilă**

Sarcina utilă reală de date pe care doriți să citiți sau scrie. Un mesaj NDEF poate conține mai multe înregistrări NDEF, așa ca nu își asumă sarcina utilă completă este în prima înregistrare NDEF a mesajului NDEF.

Sistemul de expediere a tag-urilor utilizează câmpurile de tip TNF pentru a încerca să mapeze un mesajul NDEF ca tipar de MIME sau URI. Dacă va avea succes, se încapsulează aceste informații în interiorul unui ACTION\_NDEF\_DISCOVERED intenție împreună cu sarcina utilă actuală. Cu toate acestea, există cazuri când sistemul de expediere a etichetelor nu poate determina tipul de date bazat pe prima înregistrare NDEF. Acest lucru se întâmplă atunci când datele NDEF nu pot fi mapate ca MIME sau URI, sau în cazul în care eticheta NFC nu conține date NDEF pentru a începe acest lucru. În astfel de cazuri, un obiect Tag care are informații despre tehnologiile de etichetare și despre sarcina utilă sunt încapsulate în interiorul unei locații intenție ACTION\_TECH\_DISCOVERED.

Tabelul 1. descrie modul în care sistemul de expediere tag-uri pentru maparea domeniilor TNF și pentru tipurile MIME sau URI. De asemenea, descrie care TNFs nu pot fi mapate la un tip MIME sau URI. În aceste cazuri, sistemul de expediere tag revine la ACTION\_TECH\_DISCOVERED.

De exemplu, în cazul în care sistemul de tag-ul de expediere întâlnește o înregistrare de tip TNF\_ABSOLUTE\_URI, această mapare a câmpului tip de lungime variabilă a acestei înregistrări

<sup>2</sup> <http://developer.android.com/reference/android/nfc/NdefMessage.html>

<sup>3</sup> <http://developer.android.com/reference/android/nfc/NdefRecord.html>

<sup>4</sup> <http://developer.android.com/reference/android/nfc/tech/package-summary.html>

<sup>5</sup> [http://www.nfc-forum.org/specs/spec\\_license](http://www.nfc-forum.org/specs/spec_license)

<sup>6</sup> <http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#creating-records>

într-un URI. Sistemul expediere a tag-urilor încapsulează URI în câmpul de date al unei intenții ACTION\_NDEF\_DISCOVERED împreună cu alte informații despre etichetă, cum ar fi sarcina utilă. Pe de altă parte, în cazul în care se confruntă cu o înregistrare de tip TNF\_UNKNOWN, se creează o intenție care încapsulează în locul lor tehnologii TAG.

**Tabelul 1.** TNFs acceptate și mapările lor

Denumire (TNF)	Tip	Format	Mapping
TNF_ABSOLUTE_URI			URI bazat pe tip de teren.
TNF_EMPTY			Revine la ACTION_TECH_DISCOVERED.
TNF_EXTERNAL_TYPE			URI bazat pe urna în domeniul de tip. Urnă este codificat în câmpul de tip NDEF într-o formă prescurtată: <code>&lt;DOMAIN_NAME&gt;:&lt;SERVICE_NAME&gt;</code> . Android Hărți acest la un URI sub forma: <code>vnd.android.nfc://ext/&lt;DOMAIN_NAME&gt;:&lt;SERVICE_NAME&gt;</code> .
TNF_MIME_MEDIA			Tip MIME bazat pe teren de tip.
TNF_UNCHANGED			Invalid în prima înregistrare, astfel încât revine la ACTION_TECH_DISCOVERED.
TNF_UNKNOWN			Revine la ACTION_TECH_DISCOVERED.
TNF_WELL_KNOWN			Tip MIME sau URI în funcție de înregistrare Tipul Definition (CDT), pe care le setați în câmpul de tip. Vezi tabelul 2. Pentru mai multe informații cu privire disponibile RTD și mapările lor.

**Tabelul 2.** RTD acceptate pentru TNF\_WELL\_KNOWN și mapările lor

Înregistrare Tip Definition (CDT)	Mapping
RTD_ALTERNATIVE_CARRIER	Revine la ACTION_TECH_DISCOVERED.
RTD_HANDOVER_CARRIER	Revine la ACTION_TECH_DISCOVERED.
RTD_HANDOVER_REQUEST	Revine la ACTION_TECH_DISCOVERED.
RTD_HANDOVER_SELECT	Revine la ACTION_TECH_DISCOVERED.
RTD_SMART_POSTER	URI bazat pe parsarea sarcina utilă.
RTD_TEXT	Tip MIME de <code>text</code> / plain.
RTD_URI	URI bazat pe sarcină utilă.

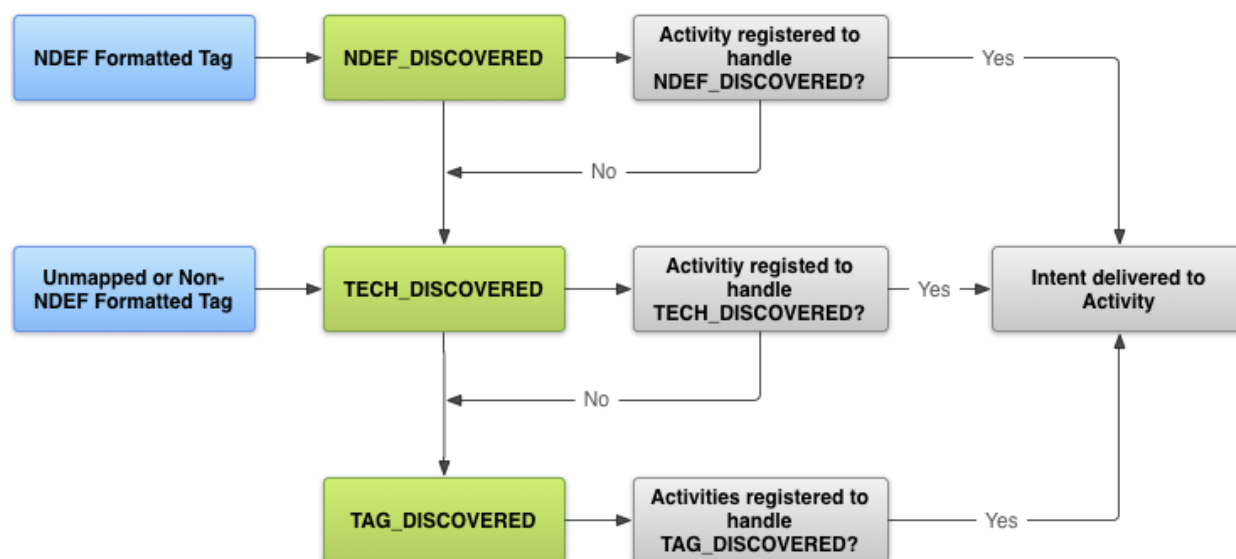
### 1.1.2 Cum sunt expediate Tag-urile NFC la Aplicații

Când sistemul de expediere al tag-urilor se face pentru crearea unei intenții care încapsulează tag-ul NFC și informațiile de identificare, acesta trimite intenția unei cereri interesate care filtrează datele pentru intenția specificată. Dacă mai mult de o cerere se repartizează unei intenții, Chooser Activity este prezentat astfel încât utilizatorul poate selecta Activity. Sistemul de expediere tag-uri definește trei intenții, care sunt enumerate în ordinea descrescătoare a priorităților:

1. ACTION\_NDEF\_DISCOVERED: Această intenție este folosită pentru a porni o activitate atunci când o etichetă care conține o sarcină utilă NDEF este scanată și este de tip recognized.

Aceasta este cea mai mare prioritate ca intenție, și sistemul de expediere a tag-ului încearcă să înceapă o activitate cu această intenție, înainte de orice altă intenție, ori de câte ori este posibil.

2. **ACTION\_Tech\_DISCOVERED**: Dacă se activează înregistrarea să se ocupe de intenția **ACTION\_NDEF\_DISCOVERED**, sistemul de expediere tag-uri încearcă să pornească o aplicație cu această intenție. Această intenție este, de asemenea, pornită direct (fără a porni mai întâi **ACTION\_NDEF\_DISCOVERED**) în cazul în care tag-ul, care este scanat conține date NDEF care nu pot fi mapate ca MIME sau URI, sau în cazul în care eticheta nu conține date NDEF, dar este o tehnologie de tag-uri cunoscută.
3. **ACTION\_TAG\_DISCOVERED**: Este pornit dacă activitățile generate au intențiile **ACTION\_NDEF\_DISCOVERED** sau **ACTION\_Tech\_DISCOVERED**.



**Figura 1.** Tag System Dispatch

Modul de bază în care funcționează sistemul de expediere tag-uri:

1. Începerea unei activități cu intenția pentru care a fost creată de sistemul de expediere a etichetelor atunci când se analizează eticheta NFC (**ACTION\_NDEF\_DISCOVERED** sau **ACTION\_Tech\_DISCOVERED**).
2. Dacă nu filtru de activitate pentru intenția care, se încearcă să pornească o Activity cu următoarea cea mai mică prioritate de intenție (**ACTION\_Tech\_DISCOVERED** sau **ACTION\_TAG\_DISCOVERED**) până se aplică o filtrare pentru intenție sau până când sistemul de expediere tag-uri a încercat toate posibilele intenții.
3. Dacă nici o cerere de filtrare pentru oricare dintre intențiile promovate, nu face nimic.

Ori de câte ori este posibil, se pot folosi mesajele de intenție NDEF și **ACTION\_NDEF\_DISCOVERED**, și anume care sunt cele mai specifice din cele trei. Această intenție permite ca cererea solicitată să înceapă în cel mai potrivit moment, altele celelalte două intenții, oferind utilizatorului o variantă mai bună, deja verificată.

## 2. SOLICITAREA de ACCES a NFC în ANDROIDMANIFEST

Înainte de a putea accesa NFC de pe un dispozitiv hardware și să trateze în mod corespunzător punctele NFC, aceste elemente trebuie declarate în fișierul `AndroidManifest.xml`:

- NFC `<uses permission>` elemente pentru a accesa hardware-ul:

```
<uses-permission android:name="android.permission.NFC" />
```

- Versiunea minimă SDK care poate suporta. Nivelul API 9 suportă doar expedierea tag-urilor limitate prin `ACTION_TAG_DISCOVERED`, și oferă acces doar la mesaje NDEF prin `EXTRA_NDEF_MESSAGES` suplimentare. Nici un fel de proprietăți ale tag-urilor sau operațiunilor I/O care sunt accesibile. Nivelul API 10 include ca suport un reader / writer, precum și un prim-plan pentru expedierea NDEF, iar nivelul API 14 oferă o modalitate mai ușoară de expediere a mesajelor NDEF la alte dispozitive cu Android Beam™ și metode mai simple pentru a crea înregistrări de NDEF.

```
<uses-sdk android:minSdkVersion="10"/>
```

- Elementul `Uses-feature` ca cerere apare în Google Play doar pentru dispozitivele care au hardware compatibil NFC:

```
<uses-feature android: name="android.hardware.nfc" android: required="true" />
```

În cazul în care solicitarea utilizează funcționalitatea NFC, dar ea nu este foarte importantă (vitală) pentru solicitarea respectivă, se pot omite puteți omite `Uses-feature-element` și se va verifica pentru valabilitatea NFC prin rutina de verificare pentru a vedea dacă `getDefaultAdapter()` este nulă.



### 3. FILTRAREA INTENȚIILOR pentru NFC

Pentru a porni aplicația, atunci când o etichetă NFC trebuie scanată, solicitarea poate filtra pentru una, două, sau toate cele trei intenții ale NFC în AndroidManifest. Cu toate acestea, de obicei se realizează filtrarea pentru ACTION\_NDEF\_DISCOVERED, în momentul în care intenția de control începe ca urmare a solicitării. Intenția ACTION\_TECH\_DISCOVERED este o rezervă pentru ACTION\_NDEF\_DISCOVERED atunci când nu se filtrează cererile de ACTION\_NDEF\_DISCOVERED sau atunci când sarcina utilă nu este NDEF. Filtrarea pentru ACTION\_TAG\_DISCOVERED este de obicei prea generală pentru a o filtra. Multe aplicații vor filtra pentru ACTION\_NDEF\_DISCOVERED sau ACTION\_TECH\_DISCOVERED mai înainte de ACTION\_TAG\_DISCOVERED, astfel încât solicitarea are o probabilitate scăzută de a începe. ACTION\_TAG\_DISCOVERED este disponibil doar în ultimă instanță pentru cererile de a filtra în cazurile în care nu sunt instalate alte aplicații care să se ocupe de ACTION\_NDEF\_DISCOVERED sau ACTION\_TECH\_DISCOVERED.

Deoarece implementările NFC pentru tag-uri variază și nu sunt de multe ori sub controlul utilizatorului, lucru care nu este întotdeauna posibil, reprezintă un motiv pentru care se pot rezerva la celelalte două intenții atunci când este necesar. Când există un control asupra tipurilor de tag-uri și date scrise, este recomandat să se utilizeze NDEF pentru a formata etichetele. Următoarele secțiuni descriu cum să se filtreze fiecare tip de intenție.

#### 3.1 ACTION\_NDEF\_DISCOVERED

Pentru a filtra intențiile pentru ACTION\_NDEF\_DISCOVERED, declara intenția către filtrul reprezentativ, împreună cu tipul de date care se dorește să fie filtrat. În următorul exemplu se prezintă filtre pentru intențiile ACTION\_NDEF\_DISCOVERED cu tip MIME de `text/plain`:

```
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />
  <category android:name="android.intent.category.DEFAULT" />
  <data android:mimeType="text/plain" />
</intent-filter>
```

Următorul exemplu de filtrare pentru un URI sub formă de <http://developer.android.com/index.html>.

```
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />
  <category android:name="android.intent.category.DEFAULT" />
  <data android:scheme="http"
        android:host="developer.android.com"
        android:pathPrefix="/index.html" />
</intent-filter>
```

#### 3.2 ACTION\_TECH\_DISCOVERED

În cazul în care se pune problema filtrării activităților pentru intenția ACTION\_TECH\_DISCOVERED, trebuie creat un fișier de resurse XML care specifică tehnologiile acceptate într-un set de `tech-list`. Activitatea este considerată o competiție în cazul în care o `tech-list` este un subset al tehnologiilor care sunt acceptate de tag-urile respective, care se poate obține apelând `getTechList()`.



De exemplu, în cazul în care tag-ul, care este scanat acceptă MifareClassic, NdefFormatable, și NFCA, setul `tech-list` trebuie să specifice toate trei, două, sau una dintre tehnologiile (și nimic altceva) pentru ca activitatea să fie acoperită.

Următorul exemplu definește toate tehnologiile. Există posibilitatea să se elimine cele care nu este nevoie. Fișier este salvat în folder-ul `<proiect rădăcină> / RES / XML`.

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2" >
  <tech-list>
    <tech> android.nfc.tech.IsoDep </tech>
    <tech> android.nfc.tech.NfcA </tech>
    <tech> android.nfc.tech.NfcB </tech>
    <tech> android.nfc.tech.NfcF </tech>
    <tech> android.nfc.tech.NfcV </tech>
    <tech> android.nfc.tech.Ndef </tech>
    <tech> android.nfc.tech.NdefFormatable </tech>
    <tech> android.nfc.tech.MifareClassic </tech>
    <tech> android.nfc.tech.MifareUltralight </tech>
  </tech-list>
</resources>
```

Se pot specifica, de asemenea, mai multe set-uri de `tech-list`. Fiecare dintre set-urile de `tech-list` este considerat independent, iar activitatea este considerată o competiție în cazul în care un singur set `tech-list` este un subset al tehnologiilor care sunt returnate de `getTechList()`. Aceasta oferă semantica **ȘI** și **SAU** pentru tehnologiile potrivite. Următorul exemplu se potrivește etichetelor care pot fi suport tehnologiile NFCA și Ndef sau pot fi suport pentru tehnologiile NfcB și Ndef:

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2" >
  <tech-list>
    <tech> android.nfc.tech.NfcA </tech>
    <tech> android.nfc.tech.Ndef </tech>
  </tech-list>
</resources>

<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2" >
  <tech-list>
    <tech> android.nfc.tech.NfcB </tech>
    <tech> android.nfc.tech.Ndef </tech>
  </tech-list>
</resources>
```

În fișierul `AndroidManifest.xml`, se specifică fișierul de resurse creat în elementul din interiorul `<meta-data> <activitatea>` ca în următorul exemplu:

```
<activity>
...
<intent-filter>
  <action android:name="android.nfc.action.TECH_DISCOVERED" />
</intent-filter>

<meta-data android:name="android.nfc.action.TECH_DISCOVERED"
```

```

    android:resource="@xml/nfc_tech_filter" />
    ...
</activity>

```

Mai multe informații despre lucrul cu tehnologii tag și intenții ACTION\_TECH\_DISCOVERED, se poate consulta [Working with Supported Tag Technologies](#) în documentul [AdvancedNFC](#).

### 3.3 ACTION\_TAG\_DISCOVERED

Pentru a filtra intenția ACTION\_TAG\_DISCOVERED se va folosi următorul filtru:

```

<intenție-filtru>
    <acțiune Android: name = "android.nfc.action.TAG_DISCOVERED" />
</intenție-Filter>

```

### 3.4 Obținerea de informații despre intenții

În cazul în care o activitate începe din cauza unei intenții NFC, se pot obține informații despre intenția de etichetare NFC scanată. Intențiile pot conține următoarele dotări în funcție de tag-ul, care a fost scanat:

- EXTRA\_TAG (necesar): un obiect Tag care reprezintă eticheta scanată;
- EXTRA\_NDEF\_MESSAGES (opțional): O serie de mesaje NDEF ale unui tag. Acest plus este obligatoriu pentru intențiile ACTION\_NDEF\_DISCOVERED;
- EXTRA\_ID (opțional): ID-ul de nivel scăzut al etichetei.

Pentru a obține aceste extra, se verifică dacă activitatea a fost lansată cu unul dintre intențiile NFC pentru a se asigura că o etichetă a fost scanată, iar apoi de a obține un extras privind intenția. Următorul exemplu verifică pentru intenția ACTION\_NDEF\_DISCOVERED și mesajele de NDEF, devine un plus intenție.

```

public void onResume () {
    super . onResume ();
    ...
    if ( NfcAdapter . ACTION_NDEF_DISCOVERED . equals ( getIntent (). getAction () ) ) {
        Parcelable [] rawMsgs = intent . getParcelableArrayExtra ( NfcAdapter .
EXTRA_NDEF_MESSAGES );
        if ( rawMsgs != null ) {
            msgs = new NdefMessage [ rawMsgs . length ];
            for ( int i = 0 ; i < rawMsgs . length ; i ++ ) {
                msgs [ i ] = ( NdefMessage ) rawMsgs [ i ];
            }
        }
    }
    //process matrice
    mesaje}

```

Alternativ, se poate obține o intenție obiect [Tag](#), care va conține sarcina utilă și va permite enumerarea tehnologiilor TAG-lui:

```

Tag tag = intenție. GetParcelableExtra ( NfcAdapter. EXTRA_TAG );

```

#### 4. CREAREA de TIPURI COMUNE de ÎNREGISTRĂRI NDEF

Această secțiune descrie modul de creare a tipurilor comune de înregistrări NDEF pentru a putea scrie etichete NFC sau de expedierea de date cu Android Beam™. Începând cu Android 4.0 (nivel API 14), este disponibilă metoda de [createUri\(\)](#), pentru a putea crea în mod automat înregistrări URI. Începând cu Android 4.1 (nivel API 16), [createExternal\(\)](#) și [createMime\(\)](#) sunt disponibile pentru a putea crea MIME și înregistrări NDEF, externe. Această metodă este utilă pentru a evita greșelile atunci când se creează manual înregistrări NDEF.

Toate aceste exemple de înregistrare NDEF trebuie să fie în prima înregistrare NDEF a mesajului NDEF pe care îl scrie o etichetă sau Bluetooth.

##### 4.1 TNF\_ABSOLUTE\_URI

**Notă:** Se recomandă utilizarea `RTD_URI` tipul loc de `TNF_ABSOLUTE_URI`, pentru că este mult mai eficient.

Se poate crea o înregistrare NDF cu `TNF_ABSOLUTE_URI` în modul următor:

```
NdefRecord uriRecord = new NdefRecord (
    NdefRecord . TNF_ABSOLUTE_URI ,
    "http://developer.android.com/index.html" . getBytes ( Charset . forName ( "US-ASCII" )),
    new byte [ 0 ], new byte [ 0 ]);
```

Filtrul anterior de intenție pentru înregistrare NDEF, ar arăta astfel:

```
<intent-filter>
  <action android:name = "android.nfc.action.NDEF_DISCOVERED" />
  <category android:name = "android.intent.category.DEFAULT" />
  <data android:scheme = "http"
        android:host = "developer.android.com"
        android:pathPrefix = "/index.html" />
</intent-filter>
```

##### 4.2 TNF\_MIME\_MEDIA

Se poate crea o înregistrare NDF cu `TNF_MIME_MEDIA` în următoarele moduri:

Utilizarea metodei [createMime\(\)](#):

```
NdefRecord mimeRecord = NdefRecord . CreateMime ( "application /
vnd.com.example.android.beam",
    "Beam me up, Android". GetBytes (caractere. ForName ("US-ASCII")));
```

Crearea manuală a [NdefRecord](#):

```
NdefRecord mimeRecord = new NdefRecord (
    NdefRecord . TNF_MIME_MEDIA ,
    "application/vnd.com.example.android.beam" . getBytes ( Charset . forName ( "US-ASCII"
)),
    new byte [ 0 ], "Beam mă, Android ". getBytes (caractere. forName (US-ASCII" ")));
```

Filtrul intenție pentru înregistrările NDEF anterioare ar arăta astfel:

```

<intent-filter>
  <action android:name = "android.nfc.action.NDEF_DISCOVERED" />
  <category android:name = "android.intent.category.DEFAULT" />
  <data android:mimeType = "application/vnd.com.example.android.beam"
/>
</intent-filter>

```

### 4.3 TNF\_WELL\_KNOWN cu RTD\_TEXT

Puteți crea o înregistrare NDEF cu [TNF\\_WELL\\_KNOWN](#) în modul următor:

```

public NdefRecord createTextRecord ( String payload , Locale locale , boolean encodeInUtf8
) {
  byte [] langBytes = locale . getLanguage (). getBytes ( Charset . forName ( "US-ASCII" ));
  Charset utfEncoding = encodeInUtf8 ? Charset . forName ( "UTF-8" ) : Charset . forName
( "UTF-16" );
  byte [] textBytes = payload . getBytes ( utfEncoding );
  int utfBit = encodeInUtf8 ? 0 : ( 1 << 7 );
  char status = ( char ) ( utfBit + langBytes . length );
  byte [] data = new byte [ 1 + langBytes . length + textBytes . length ];
  data [ 0 ] = ( byte ) status ;
  System . arraycopy ( langBytes , 0 , data , 1 , langBytes . length );
  System . arraycopy ( textBytes , 0 , data , 1 + langBytes . length , textBytes . length );
  NdefRecord record = new NdefRecord ( NdefRecord . TNF_WELL_KNOWN ,
  NdefRecord . RTD_TEXT , new byte [ 0 ], data );
  return record ;
}

```

Filtrul intenție ar arăta astfel:

```

<intent-filter>
  <action android:name = "android.nfc.action.NDEF_DISCOVERED" />
  <category android:name = "android.intent.category.DEFAULT" />
  <data android:mimeType = "text/plain" />
</intent-filter>

```

### 4.4 TNF\_WELL\_KNOWN cu RTD\_URI

Se poate crea o înregistrare NDEF cu TNF\_WELL\_KNOWN în următoarele moduri:

Utilizarea metodei (String) createUri:

```

NdefRecord rtdUriRecord1 = NdefRecord . CreateUri ( "http://example.com" );

```

Utilizarea metodei (Uri) createUri:

```

Uri uri = new uri ( "http://example.com" );
NdefRecord rtdUriRecord2 = NdefRecord . CreateUri ( uri );

```

Crearea [NdefRecord](#) manual:

```

1 pentru URI Prefixul
octet utilă [0] = 0x01; // prefixe http: // www. la URI
sistemul. arraycopy (uriField, 0, sarcina utilă, 1, uriField. lungime); // adaugă URI la

```

Filtrul intenție pentru înregistrările NDEF anterioare ar arata astfel:

```

<intent-filter>
  <action android:name = "android.nfc.action.NDEF_DISCOVERED" />
  <category android:name = "android.intent.category.DEFAULT" />
  <data android:scheme = "http"
    android:host = "example.com"
    android:pathPrefix = "" />
</intent-filter>

```

#### 4.5 TNF\_EXTERNAL\_TYPE

Se poate crea o înregistrare NDEF cu TNF\_EXTERNAL\_TYPE în următoarele moduri:  
Utilizarea metodei () createExternal:

```

byte [] sarcină utilă, // atribuiți de date
String domeniu = "com.example"; // obicei aplicației numele pachetului
String de tip = "externalType";
NdefRecord extRecord = NdefRecord.createExternal (domeniu, tip, sarcină utilă);

```

Crearea manuală a NdefRecord:

```

byte [] payload ;
...
NdefRecord extRecord = new NdefRecord (
  NdefRecord.TNF_EXTERNAL_TYPE, "com.example:externalType", new byte [ 0 ],
  payload );

```

Filtrul intenție pentru înregistrările anterioare NDEF ar arată astfel:

```

<intent-filter>
  <action android:name = "android.nfc.action.NDEF_DISCOVERED" />
  <category android:name = "android.intent.category.DEFAULT" />
  <data android:scheme = "vnd.android.nfc"
    android:host = "ext"
    android:pathPrefix = "/com.example:externalType" />
</intent-filter>

```

Utilizarea TNF\_EXTERNAL\_TYPE pentru mai multe implementări generice de tag-uri NFC reprezintă un suport mai bun pentru ambele dispozitive Android și non-Android.

**Notă:** există un format canonic de urnă pentru TNF\_EXTERNAL\_TYPE:

urna: NFC:

EXT: example.com:

externalType,

cu toate acestea, caietul de sarcini NFC Forum RTD declară urna:

NFC:

EXT: cap arte din urna trebuie omise din cazierul NDEF.

Deci, tot ce este necesar pentru a oferi domeniul (example.com în exemplu) și tipul (externalType separate de două puncte în exemplu). La expedierea TNF\_EXTERNAL\_TYPE, Android transformă urna:

NFC:

EXT:

example.com:

externalType urnă a unui vnd.android.nfc:

//ext/example.com:

externalType URI,

care este ceea ce filtrul intenție declară în exemplul dat.

#### 4.6 Înregistrarea aplicațiilor Android

Android 4.0 (nivel API 14), beneficiază de o aplicație de înregistrare Android (AAR) care oferă o certitudine mai mare atunci când o etichetă NFC este scanată. RAD are numele pachetului unei solicitări înregistrate în interiorul unui NDEF. Se poate adăuga un AAR la orice înregistrare NDEF pentru mesajul NDEF, deoarece Android caută întregul mesaj NDEF pentru RAA. Dacă găsește o AAR, începe aplicarea pe baza numelui pachetului din interiorul AAR. În cazul în care cererea nu este prezentă pe dispozitiv, Google Play este lansat automat pentru a descărca aplicația. Rapoartele anuale de activitate sunt utile pentru a împiedica alte aplicații de filtrare pentru aceeași intenție și de expediere tag-uri specifice potențial în curs de desfășurare. Rapoartele anuale de activitate sunt acceptate numai la nivel de aplicație, din cauza constrângerii numelui pachetului, și nu la nivel de activitate cu filtrarea intenției. La nivel de activitate se recomandă [use intent filters](#).

În cazul în care o etichetă conține un AAR, sistemul de expediere tag-uri lucrează, astfel:

1. se încearcă începerea activității folosind un filtru de intenție ca de obicei. Dacă activitatea care se potrivește cu intenția corespunde, de asemenea, RAA, începe activitatea.
2. în cazul în care activitatea de filtrare pentru intenție nu se potrivește cu RAA, în cazul în care mai multe activități se ocupă de intenție, sau în cazul în care nu se ocupă de activitate de intenție nici o aplicație sau activitate, trebuie pornită aplicația specificată de AAR.
3. în cazul în care nici o aplicație nu poate începe cu RAD, se accesează Google Play pentru a descărca aplicația de bază RAA.

**Notă:** Se poate înlocui RAA a sistemului de expediere intenției cu sistem de expediere în prim plan, ceea ce permite o activitate de prim-plan pentru a avea prioritate la o etichetă NFC descoperită. Cu această metodă, activitatea este în prim-plan și abia apoi trece peste RAA și sistemul de expediere intenție.

Dacă se filtrează tag-urile scanate care nu conțin o RAD, se poate declara un filtre intenție. Acest lucru este util în cazul în care solicitarea implică și tag-uri care nu conțin o AAR. De exemplu, garantarea că o solicitare accesează tag-uri de proprietate care se implementează la fel ca etichetele desfășurate de către terți. RAA sunt specifice dispozitivelor Android 4.0 sau mai noi, astfel încât la implementarea tag-urilor, se pot utiliza combinații de RAA și tipuri MIME / URI-uri pentru ca suportul să poată fi folosit pentru o gamă mai largă de dispozitive. În plus, atunci când se implementează etichete NFC, ele trebuie scrise astfel încât etichetele NFC să poată activa suportul

pentru cât mai multe dispozitive (Android și nu numai). Acest lucru se poate face prin definirea unui tip de relativ unic MIME sau URI pentru a putea fi distins mai ușor de mai multe aplicații.

Android oferă un API simplu pentru a crea un AAR, [createApplicationRecord \(\)](#). Pentru acest lucru trebuie încorpora RAD în [NdefMessage](#). Faptul că nu se utilizează prima înregistrare în [NdefMessage](#), cu excepția cazului când RAA este singura înregistrare în [NdefMessage](#). Acest lucru se datorează faptului că sistemul Android verifică prima înregistrare a unei [NdefMessage](#) pentru a determina tipul de MIME sau URI al tag-ului, care este folosit pentru a crea un filtru intenție pentru aplicații. Urmatorul cod arată cum se creează un AAR:

```
NdefMessage msg = new NdefMessage (  
    new NdefRecord [] {  
        ...,  
        NdefRecord . createApplicationRecord ( "com.example.android.beam" ) }  
    )
```



## 5. TRANSMITEREA MESAJELOR NDEF către alte DISPOZITIVE

Android Beam™ permite schimbul simplu peer-to-peer de date între două dispozitive cu Android. Solicitarea ca datele să circule de la un alt dispozitiv trebuie să fie o prioritate, iar dispozitivul care primește datele nu trebuie să fie blocat. Atunci când dispozitivul vine în contact suficient de aproape cu un dispozitiv de recepție acesta se iluminează, dispozitivul beaming afișează "Atingeți pentru a Beam" UI. Utilizatorul poate alege atunci sau nu expedierea/primirea mesajului către dispozitivul receptor.

**Notă:** Primele expedieri NDEF au fost disponibile la nivel API 10, care oferă o funcționalitate similară cu Android Beam™. Aceste API-uri de atunci au fost depreciate, dar sunt disponibile ca suport pentru dispozitivele mai vechi. Vezi [enableForegroundNdefPush \(\)](#)<sup>7</sup> pentru mai multe informații.

Android Beam™ se poate activa pentru o aplicație apelând la una dintre cele două metode:

- `setNdefPushMessage ()`: acceptă o `NdefMessage` pentru a seta mesajul ca fascicul de date. În mod automat mesajul se transferă între două dispozitive aflate în imediata apropiere unul față de altul.
- `setNdefPushMessageCallback ()`: Acceptă un apel invers care conține un `createNdefMessage ()`, care se numește atunci când un dispozitiv este în direcția de emisie (a fasciculului) de date. Utilizarea [setNdefPushMessageCallback](#) permite crearea unui mesaj NDEF numai atunci când este necesar.

O activitate poate expedia un singur mesaj NDEF la un moment dat, astfel încât [setNdefPushMessageCallback \(\)](#) are prioritate față de [setNdefPushMessage \(\)](#) în cazul în care sunt setate ambele. Pentru a utiliza Android Beam™, următoarele orientări generale trebuie să fie îndeplinite:

- activitatea care transmite datele trebuie să fie în prim-plan. Ambele dispozitive trebuie să aibă ecranele deblocate;
- datele de expediere trebuie încapsulate într-un obiect [NdefMessage](#);
- dispozitivul NFC care primește datele trebuie să se constituie suport pentru protocolul de expediere NDEF `com.android.npp` sau NFC Forum lui SNEP (Simple Protocol NDEF Exchange). `Com.android.npp` este necesar protocolul pentru dispozitivele la nivel de API 9 (Android 2.3), API 13 (Android 3.2). `Com.android.npp` și SNEP sunt necesare atât la nivelul API 14 (Android 4.0) cât și la cele care urmează.

**Notă:** În cazul în care activitatea permite Android Beam™ și este în prim-plan, sistemul standard de expediere este dezactivat. Cu toate acestea, în cazul în care activitatea permite, de asemenea dispecerizarea, atunci se pot scana în continuare tag-urile care se potrivesc filtrelor intenție stabilite. Pentru a activa Android Beam™:

1. se creează un `NdefMessage` care conține `NdefRecord` care expediază datele către celălalt dispozitiv.
2. sunetul `setNdefPushMessage ()`, cu un `NdefMessage` sau apelul `setNdefPushMessageCallback` trece într-o `NfcAdapter.CreateNdefMessageCallback` obiect în `onCreate ()`. Aceste metode necesită cel puțin o activitate care să activeze cu Android Beam™, împreună cu o listă opțională a altor activități.

În mod normal `setNdefPushMessage ()` se utilizează în orice moment atunci când trebuie transmis același mesaj NDEF, când cele două dispozitive sunt în aceeași gamă. Se poate utiliza `setNdefPushMessageCallback` atunci când solicitarea are legătură cu contextul actual al cererii și dorește să transmită un mesaj NDEF în funcție de ceea ce utilizatorul face în aplicația.

Următorul exemplu arată cum solicită o activitate simplă `NfcAdapter.CreateNdefMessageCallback` în `onCreate ()` metoda de activitate (a se vedea `AndroidBeamDemo` pentru proba completă). Acest exemplu prezintă metode pentru crearea unei înregistrări MIME:

```
package com.example.android.beam;

import android.app.Activity;
import android.content.Intent;
import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.nfc.NfcAdapter.CreateNdefMessageCallback;
import android.nfc.NfcEvent;
import android.os.Bundle;
import android.os.Parcelable;
import android.widget.TextView;
import android.widget.Toast;
import java.nio.charset.Charset;

public class Beam extends Activity implements CreateNdefMessageCallback {
    NfcAdapter mNfcAdapter;
    TextView textView;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TextView textView = (TextView) findViewById(R.id.textView);
        // Check for available NFC Adapter
        mNfcAdapter = NfcAdapter.getDefaultAdapter(this);
        if (mNfcAdapter == null) {
            Toast.makeText(this, "NFC is not available", Toast.LENGTH_LONG).show();
            finish();
            return;
        }
        // Register callback
        mNfcAdapter.setNdefPushMessageCallback(this, this);
    }
    @Override
```

```

public NdefMessage createNdefMessage(NfcEvent event) {
    String text = ("Beam me up, Android!\n\n" +
        "Beam Time: " + System.currentTimeMillis());
    NdefMessage msg = new NdefMessage(
        new NdefRecord[] { createMime(
            "application/vnd.com.example.android.beam", text.getBytes())
        /**
         * The Android Application Record (AAR) is commented out. When a device
         * receives a push with an AAR in it, the application specified in the AAR
         * is guaranteed to run. The AAR overrides the tag dispatch system.
         * You can add it back in to guarantee that this
         * activity starts when receiving a beamed message. For now, this code
         * uses the tag dispatch system.
         */
        //,NdefRecord.createApplicationRecord("com.example.android.beam")
    });
    return msg;
}

@Override
public void onResume() {
    super.onResume();
    // Check to see that the Activity started due to an Android Beam
    if (NfcAdapter.ACTION_NDEF_DISCOVERED.equals(getIntent().getAction())) {
        processIntent(getIntent());
    }
}

@Override
public void onNewIntent(Intent intent) {
    // onResume gets called after this to handle the intent
    setIntent(intent);
}

/**
 * Parses the NDEF Message from the intent and prints to the TextView
 */
void processIntent(Intent intent) {
    textView = (TextView) findViewById(R.id.textview);
}

```

```

Parcelable[] rawMsgs = intent.getParcelableArrayExtra(
    NfcAdapter.EXTRA_NDEF_MESSAGES);

// only one message sent during the beam
NdefMessage msg = (NdefMessage) rawMsgs[0];
// record 0 contains the MIME type, record 1 is the AAR, if present
textView.setText(new String(msg.getRecords()[0].getPayload()));
}
}

```

Acest cod comentarii efectuează o AAR, care se poate elimina. Dacă se activează RAA, cererea specificată în RAA primește întotdeauna mesajul Android Beam™. În cazul în care cererea nu este prezentă, Google Play este pornit pentru a descărca aplicația. Prin urmare dacă RAA este utilizat, următoarele nu sunt necesare din punct de vedere tehnic pentru dispozitivele Android 4.0 sau mai noi:

```

<intent-filter>
    <action android:name = "android.nfc.action.NDEF_DISCOVERED" />
    <category android:name = "android.intent.category.DEFAULT" />
    <data android:mimeType = "application/vnd.com.example.android.beam" />
</intent-filter>

```

Cu acest filtru intenție, `com.example.android.beam` solicitarea poate porni atunci când este scanată o etichetă NFC sau primește un fascicul de date Android, cu un AAR de tip `com.example.android.beam`, sau atunci când un mesaj NDEF formatat conține rezultatele MIME de tip `application /vnd.com.example.android.beam`. Chiar dacă rapoartele anuale de activitate garantează asupra unei aplicații, sunt recomandate filtrele intenție, pentru că acestea permit ca o activitate la alegere să înceapă aplicație în loc de a începe, conform protocoalelor, mereu cu activitățile principale din cadrul pachetului specificat de către un AAR. Rapoartele anuale de activitate nu sunt foarte precise. De asemenea, deoarece unele dispozitive Android nu acceptă RAA, ar trebui, încorporate informații de identificare, în prima înregistrare a mesajelor NDEF și filtrul pentru NDEF. Vezi [Creating Common Types of NDEF records](#) pentru mai multe informații cu privire la modul de creare a înregistrărilor.