

FlashBuddy Flash Card XML Specification

Version 1.2

Changelog

Date	User	Change
03/03/14	JLeidel	v.1.2: Adding required numcards field to deck
03/02/14	JLeidel	v.1.1: Adding title to deck format
03/01/14	JLeidel	v.1.0: Initial Printing

Contents

1	INTRODUCTION.....	4
1.1	OVERVIEW	4
1.2	DOCUMENT ASSUMPTIONS	4
1.3	TYPOGRAPHICAL CONVENTIONS.....	4
2	XML SPECIFICATION	5
2.1	OVERVIEW	5
2.2	CARD DECKS.....	5
2.3	CARD IDENTIFIERS	6
2.4	CARD DATA.....	6
3	SAMPLE XML FILES.....	7
3.1	SAMPLE #1: SINGLE CARD EXAMPLE.....	7
3.2	SAMPLE #2: MULTIPLE CARD EXAMPLE	7
3.3	SAMPLE #3: MULTIPLE TIMER EXAMPLE	8

1 Introduction

1.1 Overview

The following document describes the file format specification for the FlashBuddy card decks. Card decks are described using a predefined XML specification such that we may describe card deck subject matter, individual card layout [front versus back] as well as any plain text data to be included in the card's content. We selected XML in order to permit simple text compression for transferring over low bandwidth connections as well as the ubiquity of parsing on a multitude of platforms.

1.2 Document Assumptions

This document assumes that the reader is familiar with basic XML document markup language and syntax. This document does not assume any specific knowledge of the Android programming infrastructure, APIs or execution environment.

1.3 Typographical Conventions

This document contains the following typographical conventions:

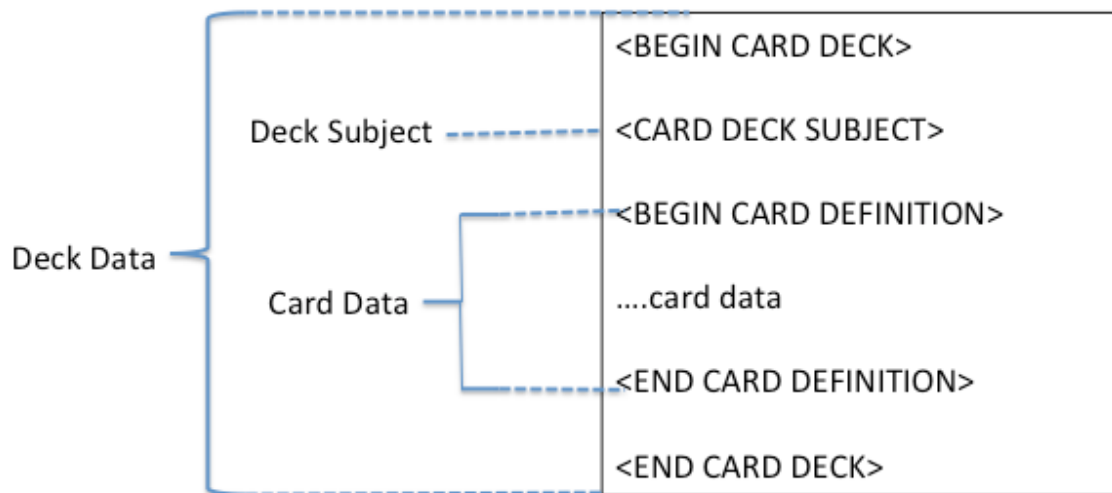
Bold Fixed Width Font is used for system-generated output and source code examples.

2 XML Specification

2.1 Overview

The following section describes the XML data format specification and layout required to build decks of one or more FlashBuddy cards. The format must be adhered to during any I/O processing in order to ensure that data is correctly rendered and/or stored to/from the FlashBuddy application.

The general XML specification follows the following form:



2.2 Card Decks

The outer-most XML section definition encapsulates the *card deck* section. This section is denoted using the **<deck>** and **</deck>** XML markup for the beginning and end of a card deck, respectively. All other data regarding the target deck must be held within. Any data found outside the deck tags is considered undefined and erroneous. Decks must include at a minimum, one card definition [Section 2.3]; the number of cards denoted by **<numcards>** and **</numcards>**; and a title denoted by **<title>** and **</title>**.

Within each deck section, the XML file can contain an optional subject specification. The subject specification is encapsulated in a beginning and ending **<subject>** and **</subject>** tag. The data contained within is any plain text value. A deck may contain at most one subject identifier. An example of a subject definition within a deck definition is as follows:

```

<deck>
<numcards>1</numcards>
<title>MyDeck</title>
<subject>Mathematics</subject>
...
  
```

`</deck>`

2.3 Card Identifiers

Card specifiers are contained within a parent deck definition. Cards cannot exist outside a beginning and ending `<deck>` and `</deck>`, respectively. The card specifier is formatted as follows:

```
<card>
...card data
</card>
```

2.4 Card Data

Each card must contain several data items. These data items cannot be omitted from the card section. These data items cannot exist outside of a card definition. The card data items are split into three areas as follows:

- Card ID: Unique card identifying integer
- Card Name: The name of a card in plain text.
- Card Timer: The maximum time to permit the user to respond, in seconds. If the time is set to zero [0], then the user has an infinite amount of time to respond.
- Question Data: The data presented to the user in the form of a question in plain text.
- Answer Data: The data representing the answer to the corresponding card questions in plain text.

An example of a single card's data region is as follows:

```
<card>
<id>5</id>
<name>Question 5</name>
<timer>0</timer>
<question>Which came first, the chicken or the
egg?</question>
<answer>Neither. This is an age old question</answer>
</card>
```

3 Sample XML Files

3.1 Sample #1: Single Card Example

```
<deck>
<numcards>1</numcards>
<title>MyDeck</title>
<subject>math</subject>
<card>
<id>1</id>
<name>Question #1</name>
<timer>0</timer>
<question>5+3=?</question>
<answer>5+3=8</answer>
</card>
</deck>
```

3.2 Sample #2: Multiple Card Example

```
<deck>
<numcards>3</numcards>
<title>MyDeck</title>
<subject>math</subject>
<card>
<id>1</id>
<name>Question #1</name>
<timer>0</timer>
<question>5+3=?</question>
<answer>5+3=8</answer>
</card>
<card>
<id>2</id>
<name>Question #2</name>
<timer>0</timer>
<question>5+4=?</question>
<answer>5+4=9</answer>
</card>
<card>
<id>3</id>
<name>Question #3</name>
<timer>0</timer>
<question>5+5=?</question>
<answer>5+5=10</answer>
</card>
</deck>
```

3.3 Sample #3: Multiple Timer Example

```
<deck>
<numcards>3</numcards>
<title>MyDeck</title>
<subject>math</subject>
<card>
<id>1</id>
<name>Question #1</name>
<timer>60</timer>
<question>5 x 3 = ?</question>
<answer>5 x 3 = 15</answer>
</card>
<card>
<id>2</id>
<name>Question #2</name>
<timer>60</timer>
<question>5 x 4 = ?</question>
<answer>5 x 4 = 20</answer>
</card>
<card>
<id>3</id>
<name>Question #3</name>
<timer>60</timer>
<question>5 x 5 = ?</question>
<answer>5 x 5 = 25</answer>
</card>
</deck>
```