Creating flashcards for gCardSys

Tommy Chang

First Drafted Sunday October 25, 2009
Last Revised April 13, 2011
(Subversion *Rev* : 10)

1 Introduction

In order to create flashcards for gCardSys, you need to create two types of files.

1. Granule's Card Deck File (.dkf)
2. gCardSys's CardBoxCabinet File (.cbx)

A Card Deck file (.dkf) contains the actual flashcards. There is no limit on how many cards a .dkf can hold. A CardBoxCabinet file (.cbx) is a container file consisted of Card Deck files used and some book-keeping information (such as flashcard arrangements and the number of cardboxes).

When gCardSys saves your work, it modifies only the CardBoxCabinet (.cbx) file. Currently, once used by a CardBoxCabinet, the Card Deck file (.dkf) should be considered read-only. However, editing/modifying *existing* flashcards already in the .dkf file is okay and the changes will be loaded the next time[1].

The reason that you should not insert new flashcards or delete existing flashcards from a Card Deck file is because once initialized, a CardBoxCabinet does not expect any of its Card Deck files to grow or shrink[2].

2 Creating a Granule's Card Deck File

Granule[1] is a flashcards software written by Vladislav Grinchenko. gCardSys uses the same[3] Card Deck File (.dkf) format uses by Granule. The format is in XML and can be easily created by any ASCII text editor.

The examples below illustrates the Card Deck file format.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE deck SYSTEM "granule.dtd">
<deck>
```

[1] In the future versions, the user can simply activate the "reload" function instead.

[2] This restriction will be lifted in the future versions of gCardSys (version 0.4 and above).

[3] Although the format is the same, Granule suggests using timestamp for card's id, whereas qCardSys does not make such suggestion.

```
 4    <!-- some global attributes: -->
 5    <author>Tommy Chang</author>
 6    <description>An example of basic Deck File Format</description>
 7    <sound_path relative="yes">sounds</sound_path>
 8    <pics_path relative="yes"/>
 9
10    <!-- first card starts here: -->
11    <card id="_0000000000">
12      <front>abase (verb)</front>
13      <back>lower and make less deserving respect</back>
14      <back_example>
15  Phrase: abase and degrade yourself
16
17  Exp1: Politicians abasing themselves before wealthy businessmen.
18      </back_example>
19    </card>
20
21    <!-- second card starts here: -->
22    <card id="_0000000001">
23      <front image="pics/abash_front.png">abash (verb)</front>
24      <back>embarrass</back>
25      <back_example image="pics/abash_back.png">
26  Phrase: abashed with sudden shame
27
28  Exp1: She seemed a little abashed when they asked about her job.
29      </back_example>
30    </card>
31
32    <!-- rest of cards start here: -->
33
34  </deck>
```

- Lines 1-2 are XML declaration.
- Line 3 starts the deck.
- Lines 5-8 sets some global attributes. See granule.dtd to learn about other attributes. Not all attributes are used by gCardSys.
- Line 11-19 defines the first card.

  The card's `id` attribute always start with an underscore “_” and followed by a 10-digit number (in the case for Granule, this number should be the seconds since 1970; In gCardSys, the number can be anything as long as it is unique.)

  Although the content of the card doesn't have a strict format, it should be noted that some special xml-related symbol are not allowed. They include angle brackets: > and <

  Also, it's customary (but optional) to have a category string enclosed inside parenthesis inside the `<front> </front>` field. For example:

  ```
  <front>hello (noun)</front>
  ```

  There is a single space between the word “hello” and its category “(noun)”.

- Similarly, lines 22-30 defines the 2nd card. Note the 2nd card specifics the locations of front and back_example images, which are optional.

- Line 34 matches line 3 and closes the entire deck definition.

- Lines 4, 10, 21, and 32 are comments. Comments are always ignored. Blank lines 9, 20, 31, and 33 are also ignored.

- In general, you can put anything between the tag pairs: `<front>` and `</front>`; `<back>` and `</back>`; `<back_example>` and `</back_example>`. Note, white spaces and blank lines within the tag pair are preserved and they will show up in your card.

3 Creating a gCardSys CardBoxCabinet File

gCardSys uses the CardBoxCabinet file format (.cbx) to store your gCardSys data. These data consist of data for tracking your progress. An initial .cbx file needs to be created first. The following example shows how to create an initial .cbx file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CardBoxCabinet SYSTEM "cardBoxCabinet.dtd">
<CardBoxCabinet promoteCountThres="3"
                box1Study="30" higherBoxesReview="45">
  <!-- list of deck files to use: -->
  <CardDeckFile file="decks/abc.dkf"/>
  <CardDeckFile file="decks/efgh.dkf"/>
  <CardDeckFile file="decks/ijkl.dkf"/>
  <CardDeckFile file="decks/mnop.dkf"/>
  <CardDeckFile file="decks/qrst.dkf"/>
  <CardDeckFile file="decks/uvwxyz.dkf"/>

  <!-- content of 1st cardbox: -->
  <CardBox expPeriodHrs="0">
    <UnVisitedCards file_id="0" card_id="_all"/>
    <UnVisitedCards file_id="1" card_id="_all"/>
    <UnVisitedCards file_id="2" card_id="_all"/>
    <UnVisitedCards file_id="3" card_id="_all"/>
    <UnVisitedCards file_id="4" card_id="_all"/>
    <UnVisitedCards file_id="5" card_id="_all"/>
  </CardBox>

  <!-- content of 2nd cardbox: -->
  <CardBox expPeriodHrs="12">
  </CardBox>

  <!-- content of 3nd cardbox: -->
  <CardBox expPeriodHrs="24">
  </CardBox>

  <!-- content of 4th cardbox: -->
  <CardBox expPeriodHrs="48">
  </CardBox>

```

```
35    <!-- content of 5th cardbox: -->
36    <CardBox expPeriodHrs=”96”>
37    </CardBox>
38
39    <!-- the trash cardbox: -->
40    <CardBox expPeriodHrs=”−1”>
41    </CardBox>
42
43  </CardBoxCabinet>
```

- Lines 1-2 are XML declaration.

- Lines 3-4 starts the CardBoxCabinet. The optional attributes:

  – `promoteCountThres` specifies the number of promotions a card in cardbox 1 must receive in order to be really promoted to cardbox 2. If not specified, the default value is 3.

  – `box1Study` specifies the number of minutes to spend learning new cards from cardbox 1. If not specified, the default value is 30.

  – `higherBoxesReview` specifies the number of minutes to spend reviewing old cards from all higher cardboxes combined. Default value is 45.

- Lines 6-11 lists all the Granule Deck Files (.dkf) to use in this CardBoxCabinet. There is no limit on how many .dkf files you can use in a .cbx file. However, the order of the list is important, as it determines the `file_id` within the .cbx file (see content of cardbox in next item).

- Lines 14-21 defines the content of cardbox 1. The attributes `file_id` and `card_id` are mandatory. Together they identify an unique card from the corresponding .dkf file. For example, `file_id="0"` refers to the first .dkf file, or `decks/abc.dkf` in our example.

  The `card_id` attribute matches a card's `id` attribute in the .dkf file. However, to save space, it does not need to have the leading 0s. For example `<card id="_0000002682">` in the .dkf file is referred by `card_id="_2682"` in the .cbx file.

  When the `card_id` attribute takes the value of `"_all"`, as in `card_id="_all"`, it refers to *all* cards from the corresponding .dkf file.

  The attribute `expPeriodHrs` defines the cardbox's expiration time in hours. Typically we set cardbox 1 to be always expired (i.e., `expPeriodHrs="0"`).

- Lines 24-37 defines empty cardboxes 2 to 5. The corresponding expiration time are 12, 24, 48, and 96 hours. You can experiment with these settings and there is no limit on the number of cardboxes you can have.

- Lines 40-41 specifies a special trash cardbox. This is indicated by the -1 expiration time as in `expPeriodHrs="-1"`. gCardSys allows you to explicitly move a card into this trash cardbox and the card will never be presented again.

The creation of this trash cardbox is optional, gCardSys will create one for you if you did not create it initially.

- Finally, empty field such as ’`<back></back>`’ is allowed, but `<front>` must not be empty.

When you save your work, the .cbx file will get updated. The .dkf files will not be modified. This way, you can share .dkf files in different .cbx files.

References

[1] http://granule.sourceforge.net/