

# Bank of CUCC (.pl)

Stuart Bennett  
sb476@cam.ac.uk

June 18, 2013

Copyright © 2013 Stuart Bennett

Typeset using the  $\LaTeX$  document presentation system in the Palatino and Computer Modern fonts.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation and Setup</b>	<b>3</b>
2.1	Installation . . . . .	3
2.2	Setup . . . . .	4
<b>3</b>	<b>The Admin Control Panel</b>	<b>6</b>
3.1	Account management . . . . .	6
3.1.1	User accounts . . . . .	6
3.1.2	Virtual accounts . . . . .	7
3.1.3	Address alternatives configuration . . . . .	7
3.1.4	Personal attribute configuration . . . . .	7
3.1.5	Attribute group configuration . . . . .	8
3.2	Events . . . . .	8
3.2.1	Events . . . . .	8
3.2.2	Fee templates . . . . .	10

3.2.3	Event types . . . . .	10
3.3	Charges and expenses . . . . .	11
3.3.1	Charge/credit/expense configuration . . . . .	11
3.3.2	Currency configuration . . . . .	12
3.4	General configuration . . . . .	13
3.4.1	Installation configuration . . . . .	13
<b>4</b>	<b>User-visible interfaces</b>	<b>15</b>
4.1	The user account view . . . . .	15
4.2	Quick expenses forms . . . . .	15
4.3	Transaction Groups (TGs) . . . . .	16
4.3.1	Header fields . . . . .	16
4.3.2	Tabulated fields . . . . .	17
4.3.3	Transfer Pots . . . . .	17

# Chapter 1

## Introduction

The Bank of CUCC (henceforth, 'boc') is a software package designed to ease the life of an unfortunate charged with keeping track of expenses and distributing costs for and between a number of people. An example of same might be the treasurer of a club or society, but equally might be someone keeping track of expenditure for a group of people on holiday.

boc is designed to permit data to be entered relatively easily, to allow the current state of individuals' tabs to be accessible and to have all day-to-day operations conducted through a web interface (with concurrency of users). Secondary, but important, features are fairly strong security, sufficient abstraction to be flexible in application, version controlled data-store, and ease of auditing.

A few warnings to make your boc experience less irritating:

- boc validates input as strongly as it can, and will complain if it doesn't like something. Unfortunately it won't refill forms for you. Hitting your browser's back button will probably allow you to recycle your previously input values, if you wish.
- boc doesn't mind tabbed use per-se, but it does take exclusive locks out on files being edited. Thus attempting several tabs editing one thing will

not work. Closing a tab which is editing a file will result in that file being uneditable until the lock expires.

## Chapter 2

# Installation and Setup

### 2.1 Installation

Installation is minimal, and straightforward for anyone with some experience of installing CGI applications.

1. Put the code (boc.pl, \*.pm) in a directory where the web-server will accept boc.pl a CGI executable programme.
2. Place the contents of the 'templates' directory somewhere. It is simplest for this to be a subdirectory 'templates' of the directory containing boc.pl.
3. Create a data-store directory, with read/write permission by the web-server. This directory need not (and indeed, probably shouldn't) be in a web-visible path.
4. Place the top-level configuration file in the same directory: an example is included with the software, but the style is one key/value pair per line, separated by a tab.
5. Set the value of the 'Root' key in the configuration file to the absolute path of the data-store directory.

6. Set the value of the 'TemplateDir' key in the configuration file to the path of the templates directory. If the suggestion configuration is used, this will be 'templates'.
7. Install necessary Perl modules. A list is given below, all are available from CPAN:
  - CGI::Session
  - Crypt::Cracklib
  - Crypt::PasswdMD5
  - File::Slurp
  - Git::Wrapper
  - HTML::Entities
  - HTML::Template
  - HTML::Template::Pro
  - JSON::XS
  - MIME::Lite
  - Text::Password::Pronounceable
  - Time::ParseDate
  - URI::Escape
  - UUID::Tiny
8. Browse to the appropriate URL to execute boc.pl. Errors will be displayed for any missing Perl modules; a screen inviting the creation of administrator account otherwise.

## 2.2 Setup

Due to the flexible nature of the software a prescriptive guide to setup is not appropriate. The below will describe a minimal use-case. All setup is conducted through the web interface.



1. Supply a username and password for the (first) administrator account.  
You now have a minimally functional boc installation...
2. Log in, using the just-supplied account details.
3. Go to 'Admin control panel', then 'Edit installation configuration'.
4. See description of this configuration page in subsection 3.4.1. Make changes as appropriate, and save. Setting the system names and style file URL are *highly* recommended.
5. Create some users via 'Manage people' (subsection 3.1.1) and/or non-personal accounts via 'Manage virtual accounts' (subsection 3.1.2).<sup>1</sup>
6. (Optional) setup a default and presentation currency, via 'Edit units', described in subsection 3.3.2.

---

<sup>1</sup>You may like to set up address alternatives and the attributes system first (subsection 3.1.3 and subsection 3.1.4), but this is not always necessary

## Chapter 3

# The Admin Control Panel

### 3.1 Account management

#### 3.1.1 User accounts

The account management page lists all known users accounts, gives a brief overview of their properties and permits their addition, modification, deletion and the setting/clearing of passwords.

When adding or editing a user account a short reference name (lowercase, 3-10 letters) is required, as well as a proper display name, email address, real-world address (or address alternative, subsection 3.1.3). Individual attributes can also be set on an account (see subsection 3.1.4 and subsection 3.1.5).

In order to set an account password the system email configuration must be completed and correct (subsection 3.4.1), as passwords are auto-generated and sent to the email address associated with the account.

### 3.1.2 Virtual accounts

Virtual accounts are accounts not associated with a user – nominal ring-fenced funds, or real accounts (such as a bank account).

The account management page lists all known virtual accounts, gives a brief overview of their properties and permits their addition, modification and deletion.

When adding or editing a virtual account a short reference name (lowercase, 3-10 letters) is required, as well as a proper display name.

A virtual account can be a ‘negated’ account: a prime example being a bank account. Transactions between regular and ‘negated’ accounts suffer a sign inversion on the negated account: whereas a swap of 5 between regular accounts a and b would result in a diminishing by 5 and b increasing by 5, a withdrawal of 5 from regular account a ‘to’ negated account n results in *both* a and n diminishing by 5. The use of this becomes clear when a is a user account and n is the bank account: if 5 pounds are given in reality to user a, both the credit on their tab and the bank account balance must decrease.

### 3.1.3 Address alternatives configuration

When adding/editing a user account it is a requirement that some *address* information is specified. The address alternatives system permits the creation of drop-down menus whose options can be used instead of typing an address.

### 3.1.4 Personal attribute configuration

The personal attribute configuration page allows user-defined attributes to be created, which can then subsequently be associated with appropriate user accounts. Such attributes are ‘Has’ or ‘Is’ prefixed. For example, to permit recording of whether a user has a car we could have a ‘Has’ attribute called ‘ACar’, or for whether someone can drive an ‘Is’ attribute called ‘Driver’.

### 3.1.5 Attribute group configuration

Attributes can either be user-defined (subsection 3.1.4) or system-defined.

The system attributes are as follows:

**IsAdmin** any user with this has full administrative privileges within the boc web application

**IsAuthed** any user with this has logged in with a password

**IsPleb** all users have this attribute

**MayAddEditTGs** any user with this may change Transactions Groups arbitrarily

**MayEditOwnEvents** any user with this may change Events for which they are the recorded organizer

**MayStewardEvents** any user with this may add, delete or edit the basic details of any Events

Attributes can be combined logically to create a powerful and expressive capability system, both permitting one attribute to imply several others, and also the combination of several attributes to imply a different attribute.

## 3.2 Events

### 3.2.1 Events

Events are essentially a system for creating templated forms for a non-trivial set of related transactions. By creating such forms, not only does it make it easier for the boc overseer to use, but it allows the possibility of those organizing the events inputting the data directly (and having half a chance of doing it correctly).

Since events can be delegated in this way, a mechanism to prevent further changes after auditing is provided: by using an event's lock button changes may only be made by an administrator; unlocking is the inverse of locking.

Apart from adding a new event, events are edited from the detailed event view (obtained clicking an event's name).

### **Event headings (basic details)**

*Accessible by users with MayStewardEvents.*

Creating an event requires the provision of some basic details: an event name, a date on which the event starts, the event's duration, a nominated organizer (selected from those with user accounts on the boc system), and optionally an Event Type/Fee Template.

The choice of Event Type/Fee Template is quite important for the system to work satisfactorily. The choice of Event Type determines the columns presented in the Event form (a choice of 'none' means all configured fees/expenses are presented). The choice of Fee Template determines the default charges that will be applied to event participants, see subsection 3.2.2 for more details of this capability.

### **Event participants**

*Accessible by organizers with MayEditOwnEvents.*

Accounts may be associated with/disassociated from an Event using checkboxes. More recently active user accounts are listed first (according to the threshold in subsection 3.4.1). A button permits creation of new accounts, should they not already exist.

Values for a newly added participant will be automatically filled according to any fee template associated with the Event.

## **Event values**

*Accessible by organizers with MayEditOwnEvents.*

Values may be filled in as appropriate. The 'Custom Fee' column permits arbitrary charging to the Event's default account, should it be desired.

'Splits' allow direct currency swaps between Event participants, for instance if one has incurred an expense on behalf of another. The expense is entered in the appropriate column, and shares allocated to those owing. Any positive (non-zero) value will work for a simple swap. Descriptions of each split/swap should be entered below, to avoid confusion.

### **3.2.2 Fee templates**

Fee templates are only available when some commodity units are defined (subsection 3.3.2).

Fee templates (FTs) allow default charges to be applied to event participants as they are added to an Event. The charges applied may depend on what attributes are set on the event participant's account.

FTs can be specific to one Event Type, or general. This is selected at the time of FT creation from the drop-down box.

FTs must have a name. This will be prefixed with the Event Type, if set.

### **3.2.3 Event types**

Event types are only available when some expense categories are defined (section 3.3.1).

Event types (ETs) allow the display of an event to be customized, showing only the desired expense columns in the desired order.

An alternative default linked account may be set, overriding that set in the

general expense configuration (section 3.3.1).

For each known fee/expense the order in which to display the column (or disable it) is configurable, as is whether modifications to the column's values require confirmation, and optionally what descriptive text should be presented at the column heading (again, overriding that from the general expense configuration).

## **3.3 Charges and expenses**

### **3.3.1 Charge/credit/expense configuration**

This configuration is heavily involved in the templating of expenses: if this configuration is missing or invalid the Events system and quick-expenses forms will not be accessible.

#### **Default event account**

This is the account against which event proceeds and expenses credit/debit, if no other account takes precedence.

#### **Charges, credits and expenses**

Here one can define expense categories for use in Events and in the quick-expense forms. Expenses can either be pre-existing commodity units, or user-defined categories. The word 'expense' is used generally in this description, but obviously a negative expense is a credit.

All expenses have a 2-4 letter code and a descriptive name (these are pre-filled for commodity units), a Boolean flag and a linked account.

**Is Boolean?** makes it a yes/no expense, rather than being able to take a wider range of quantities.

**Linked account** is the account to/from which the expense will be credited/debited.

This must be set in order for the expense to be valid.

User-defined (non-commodity) expenses have two additional flags:

**Drain linked account?** specifies that the amount of the expense will be the value of the linked account shared across the event participants in proportion to the quantities entered

**Expensable?** makes the expense available for use in the quick-expense forms. It cannot be used in combination with the Boolean or drain flags.

### 3.3.2 Currency configuration

#### Unit definition

This page permits the definition of the units used by boc. These can be both real currencies (Pounds Sterling, Euros, etc.), and commodities – a day’s equipment hire for instance; any thing to which a per-unit value can be attached (even a negative value, in the case of a discount).

Only so many units may be defined at a time – more slots will become available after saving should you wish to add more than you have available rows for at one time.

A unit definition requires a description (‘Pounds Sterling’, ‘Lamp hire/days’), a 2-4 letter code (‘GBP’, ‘LAMP’), and determination of whether the unit is a currency or a commodity.

One currency (a non-commodity) should be set as the ‘Anchor currency’: this is the currency to which other currencies are referred. Changing this will require all exchange rates to be re-entered appropriately.

One unit should be set as the ‘Presentation default’: this is the unit all values are converted to and displayed in. Can be changed arbitrarily.



To save the unit definitions, click 'Proceed to set exchange rates'.

### **Exchange rate configuration**

Each defined unit requires a rate to relate it to the others. Commodities can be defined in terms of any currency – the drop-down defines which.

Each rate needs a date which defines the date from which the rate is effective (inclusive of the given date). Any transaction using a currency on a date before a rate for that currency was defined will use the first available date/rate for that unit.

Note that commodity rates are *Currency Per Unit*, whereas currencies are *Currency Per Anchor Currency*.

## **3.4 General configuration**

### **3.4.1 Installation configuration**

**System short name** : an abbreviated, preferably single word name for the installation (e.g. 'boc')

**System long name** : a full name for the system (e.g. 'Bank of CUCC')

**System style file location** : URL for the style file loaded by all boc pages

**timeout.js URL** : URL to the file timeout.js. boc will use the copy distributed in the templates directory if the URL is not supplied, but this will cause *two* boc.pl invocations for every page load, and hence is disagreeably slow.

**Allow passwordless user logins?** : whether users can log-in to their accounts when they don't have a password set

**System email address** : the email address from which mail sent by the system should appear to have come

**SMTP relay host** : hostname of a functional SMTP relay. Or use 'sendmail' if the web-server is also running a mail server.

**Boundary for 'recent' people in event participation** : the threshold number of days for people to be counted as having recently attended an Event

## Chapter 4

# User-visible interfaces

### 4.1 The user account view

The user account view is the first page any user sees after logging in, and a very similar page is displayed when looking at another account (by clicking on the account's name). It displays details of all transactions known to be associated with that account, itemized by Transaction Group, and links to any Events for which the user is responsible.

A number of buttons are available, the exact number varying depending on the user's permissions. In the minimal case, apart from the pervading 'Logout' button, the Accounts summary and lists of all Transaction Groups and Events (if configured) are available.

### 4.2 Quick expenses forms

*Accessible by users with MayAddEditTGs.*

From the user account view, a number of convenient forms to enter swaps, splits and expenses may be available.

If some expenses with the *Expensable* flag are defined in the fee configuration (section 3.3.1), one may ‘Add simple communal expense’ or ‘Add multi-creditor/category communal expense’. As the names imply, the former is for the case of one creditor and one expense type, the later for the more involved case.

Likewise, the same simple/harder divide applies to the ‘Add person-to-person swap’ and ‘Add a set of expenses split across a number of people’ forms.

‘Add credit/debit to real-world account’ is *only available to administrators*, primarily as a quick way of entering bank transactions.

### 4.3 Transaction Groups (TGs)

Transaction Groups underpin the whole boc accounting system – all other forms reduce to a TG. From an account view, or from the list of all TGs, a TG may be inspected. *For users with MayAddEditTGs it may also be edited; this capability also permits the creation of new TGs (from the list of all TGs).*

A transaction group (TG) is a record of a logically related financial event where funds move from m accounts to n accounts. It is generally quite flexible in what it can represent, but for clarity this should not be abused into commingling transactions distinct in time, place or nature.

#### 4.3.1 Header fields

**Name** a representative and descriptive name for the transactions, this is what the TG’s purpose will be summarized by in other displays

**Date** when the transaction(s) happened. Preferably in DD.MM.YYYY, but more imaginative forms may be correctly parsed. It is important to get this right so that correct currency conversions may be applied. Also, TGs tend to be sorted by date.

**Omitted** set if the TG's effects should not be included in the overall account balances

### 4.3.2 Tabulated fields

Typically each row describes one transaction, with the following fields:

**Creditor** to whom (/what subsection 4.3.3) the Amount column applies

**Amount** the amount to credit to (or, for a negative value, debit from) the Creditor. '\*' uses the negated balance of the Creditor account, thereby draining it.

**Currency** the currency in which the Amount is expressed

**Debt shares** the proportions to allocate to each of the accounts to finance the Amount. Must be positive. Shares are calculated by taking the ratio of an account's share to the sum of all shares. This allows true monetary values to be used if so desired, assuming they sum to the Amount. Accounts with italicized headings are Negated accounts, where the account change happens in the same sense (credit/debit) as that of the Creditor.

**Transfer pot** see below

**Description** a particular description for the transaction row, be it describing the Creditor, Amount, or Debt share

Any field appearing with a red background is one whose saved contents do not now validate for some reason. Such fields must be amended to permit saving of the TG.

### 4.3.3 Transfer Pots

Transfer pots (TPs) are a powerful way to link several rows to make a transaction with multiple Creditors. A TG may have up to nine TPs, 1-9.

Any rows which have the same number (1-9) in the Transfer Pot field have their debt-shares combined. The Amount to be split across the shares is the nett Amount following summation of the combined rows' Amounts.

A numbered Transfer Pot may also be set as a Creditor. In this circumstance the Amount must be empty or '\*', as this simply specifies that all of the row's debt-shares are to be added to those already in the appropriate pot.