



## **CBA Excel Link**

© 2006 Greentree International Ltd

## **CBA Excel Link**

© 2006 Greentree International Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: November 2006

Publisher: Greentree International Ltd

---

# Table of Contents

<b>Part I Overview</b>	<b>1</b>
<b>Part II Installation</b>	<b>1</b>
<b>Part III Menu Options</b>	<b>1</b>
<b>Part IV Functions</b>	<b>2</b>
1 cbaCVPeriod .....	2
2 cbaCVYear .....	3
3 cbaGL .....	3
4 cbaGLAccBal .....	4
5 cbaGLBud .....	5
6 cbaGLCurPeriod .....	6
7 cbaGLCurYear .....	6
8 cbaGLName .....	7
9 cbaHide .....	7
<b>Part V Security</b>	<b>8</b>
	<b>0</b>

## 1 Overview

CBAXLink is an add-in for Excel that has been developed to allow direct access to the CBA database in order to retrieve General Ledger account information and balances. Thus allowing for financial reports to be written in Excel.

This add-in is compatible with Excel 2000 (Office 2000) and above. It can not be used with older versions of Excel.

## 2 Installation

The components for CBAXLink consist of CBAXLink.xla and CBAXLink.dll and these components need to be installed using the installer provided.

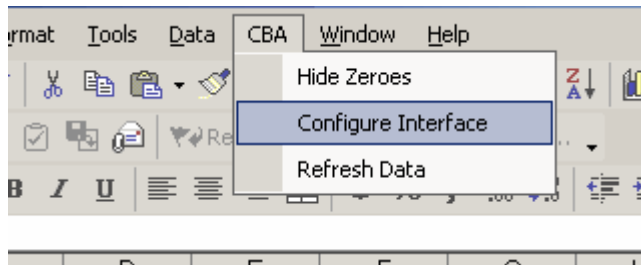
To installation needs to be run on each workstation that will utilise CBAXLink. If multiple users use the workstation and they all want to utilise CBAXLink, it will need to be installed for each user.

### To Install

1. Login as the User that will be using CBAXLink on the workstation. This user will need to have Administrator rights to be able to complete the installation.
2. Run `x:\CBA2000.v10\CBAXLink\CBAXLink1101.exe`
3. Start **Excel** and from the **Tools** menu select **Add-ins**. In the list of Add-ins, tick Cbaxlink.

## 3 Menu Options

After the CBAXLink add-in has been installed a **CBA** menu option will be available from with Excel.



### Hide Zeros

This option can be toggled on or off and will look for cells that contain the [cbaHide](#) function and hide or display rows accordingly.

### Configure Interface

This option allows for selection of which database is to be used for retrieving data. The path defined needs to contain the CBA General Ledger files. This option will not need to be used unless there is more than one CBA database that reports are being run on.

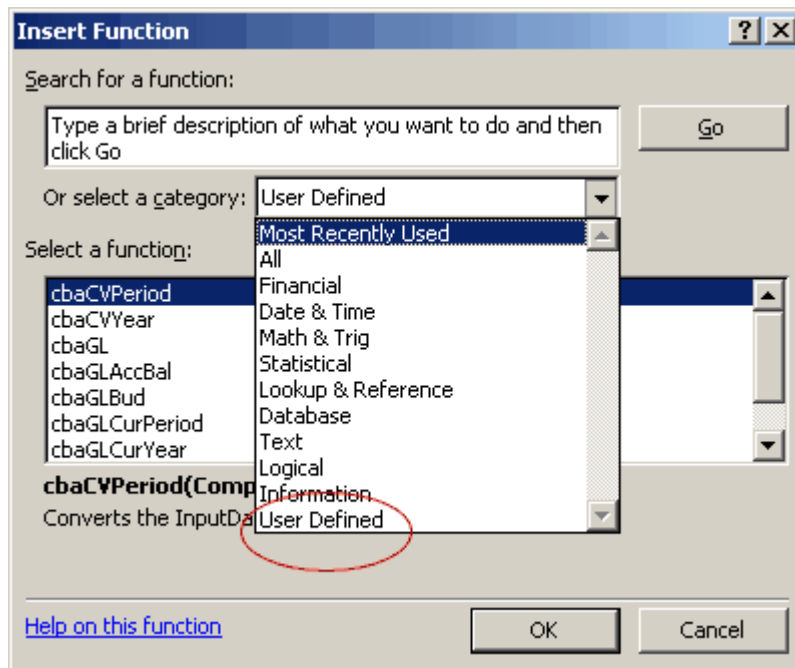
### Refresh Data

This option forces Excel to re-process the functions and return the most up to date data. Normally this option would only need to be used when the CBA data has been changed while the spreadsheet has been open.

## 4 Functions

These functions are only available once the CBAXLink add-in has been activated Excel.

Access to these functions is available by choosing to see the User Defined functions.



### 4.1 cbaCVPeriod

This function converts a date into a CBA Calendar Period. The period number is returned and can be used in subsequent functions that get the account balances.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

**InputDate** (mandatory) - the date to be converted. This date may be in any valid Excel date format and must be entered with text delimiters, e.g. "01/11/2005"

**RelativePeriod** (optional) - the number of periods (+ or -) to move from the period the date being converted falls into.

Using a CBA calendar for Company 01, based on calendar months and covering 01/04/2005 (being period 1) to 31/03/2006 (being period 12);

`cbaCVPeriod("01","30/11/2005")` returns 8

`cbaCVPeriod("01","30/11/2005",-1)` returns 7

## 4.2 cbaCVYear

This function converts a date into a CBA Calendar Year. The year number is returned and can be used in subsequent functions that get the account balances.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

**InputDate** (mandatory) - the date to be converted. This date may be in any valid Excel date format and must be entered with text delimiters, e.g. "01/11/2005"

**RelativePeriod** (optional) - the number of periods (+ or -) to move from the period the date being converted falls into.

Using CBA calendars for Company 01, based on calendar months, where year 2005 is 01/04/2004 (being period 1) to 31/03/2005 (being period 12) and year 2006 is 01/04/2005 to 31/03/2006;

```
cbaCVYear("01","30/11/2005") returns 2006  
cbaCVYear("01","30/11/2005",-12) returns 2005
```

## 4.3 cbaGL

This function returns the value of any of the fields that are available in CBA on the GL Masterfile record. It is useful to find information that can effect how you want other cells formatted e.g. finding the account's balance sign or short code.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

**Account** (mandatory) - a valid GL account number. Wildcards are not allowed to be used as this argument must refer to a single account.

**FieldName** (mandatory) - a valid field name from the GLMASTER file.

Account MAS.AK.SLS.1010 is defined as having a description of SALES - DESKS & CHAIRS, along with a short code of SLS and balance sign of CR.

```
cbaGL("01","mas.ak.sls.1010","short_code") returns SLS  
cbaGL("01","mas.ak.sls.1010","sign") returns CR  
cbaGL("01","mas.ak.sls.1010","desc") returns SALES - DESKS & CHAIRS
```



*If this function is used to return the account description, it will be returned in exactly the format it is entered in with CBA, i.e. all caps. If you prefer to have the account description returned using normal sentence caps .i.e. upper and lower case, use the **cbaGLName** function instead.*

## 4.4 cbaGLAccBal

This function returns the account balances in either the ledger (base) currency or the currency the account is held in (foreign). It is also used to return quantity balances for accounts that have been designated as quantity accounts.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

**FromAccount** (mandatory) - a valid GL account number. Wildcards can be used to accumulate the balances of multiple accounts. Segments may contain a single wildcard character ( \* or ?).

Accounts MAS.AK.SLS.1010, MAS.WN.SLS.1010, MFG.WN.FTY.1010 & MFG.SY.FTY.1010 all contain balances however it is desired to report these accumulated balances as a single value.

The From Account would be entered as \*.\*1010

**FromYear** (mandatory) - a valid CBA calendar year that you wish to start from when retrieving balances.

**FromPeriod** (mandatory) - the first period number, of the above year, that you wish to start from when retrieving balances. If you wish to start from the Opening Balance (for a balance sheet account) this is considered to be period zero.

**ToAccount** (optional) - *if left blank it will be assumed to be the same as the FromAccount.* If you wish to report on a range of accounts, enter the last account in the range. Segments may contain a single wildcard character ( \* or ?).

Accounts MAS.AK.SLS.1010, MAS.WN.SLS.1010, MFG.WN.FTY.1010, MFG.SY.FTY.1020 & MAS.AK.SLS.1020 all contain balances however it is desired to report these accumulated balances as a single value.

The From Account would be entered as \*.\*1010

The ToAccount would be entered as \*.\*1020

**ToYear** (optional) - *if left blank it will be assumed to be the same as the FromYear.* If you wish to report on a range of years, enter the last year in the range.

**ToPeriod** (optional) - *if left blank it will be assumed to be the same as the FromPeriod.* If you wish to report on a range of periods, enter the last period in the range.

**IncludeFlags** (optional) - if you have utilised User Flags on your CBA GL accounts you may choose to only include accounts which match certain user flags. There are up to nine user flags that can be used and when entering them in this function they are separated by a semi colon.

User Flags have been set up where by an X in position 1 indicates regular local sales, X in position 2 indicates seasonal local sales, X in position 3 indicates regular export sales and X in position 4 = seasonal export sales. Sales Accounts fall in the range of 1000 to 1099.

To get a balance for only sales accounts flagged as "regular local sales" you would enter

From Account = \*.\*1000

ToAccount = \*.\*1099

IncludeFlags = X

To get a balance for only sales accounts flagged as "seasonal local sales" you would enter

```
From Account = *.*1000
ToAccount = *.*.1099
IncludeFlags = ;X
```

To get a balance for sales accounts flagged as "seasonal sales" (being both local and export) you would enter

```
From Account = *.*1000
ToAccount = *.*.1099
IncludeFlags = ;X;;X
```

**ExcludeFlags** (optional) - if you have utilised User Flags on your CBA GL accounts you may choose to exclude accounts which match certain user flags. There are up to nine user flags that can be used and when entering them in this function they are separated by a semi colon.

**BalType** (optional) - *if left blank it will be assumed to be "V", meaning value.* If you wish to return the quantity enter a "Q".

**FCBal** (optional) - *if left blank it will be assumed to be FALSE, meaning it will return the ledger (base) currency value.* If you wish to return the value in the currency of the account enter TRUE.

## 4.5 cbaGLBud

This function returns the account budget balances in either the ledger (base) currency or the currency the account is held in (foreign). It is also used to return quantity budgets for accounts that have been designated as quantity accounts.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

**FromAccount** (mandatory) - a valid GL account number. Wildcards can be used to accumulate the balances of multiple accounts. Segments may contain a single wildcard character ( \* or ?).

Accounts MAS.AK.SLS.1010, MAS.WN.SLS.1010, MFG.WN.FTY.1010 & MFG.SY.FTY.1010 all contain budgets however it is desired to report these accumulated budgets as a single value.

The From Account would be entered as \*.\*1010

**Year** (mandatory) - a valid CBA budget year.

**FromPeriod** (mandatory) - the first period number, of the above year, that you wish to start from when retrieving balances. If you wish to start from the Opening Balance (for a balance sheet account) this is considered to be period zero.

**BudgetNumber** (mandatory) - a valid budget number that relates to the budget year selected.

**ToAccount** (optional) - *if left blank it will be assumed to be the same as the FromAccount.* If you wish to report on a range of accounts, enter the last account in the range. Segments may contain a single wildcard character ( \* or ?).

Accounts MAS.AK.SLS.1010, MAS.WN.SLS.1010, MFG.WN.FTY.1010, MFG.SY.FTY.1020 & MAS.AK.SLS.1020 all contain budgets however it is desired to report these accumulated budgets as a single value.

The From Account would be entered as \*.\*1010

The ToAccount would be entered as \*.\*.1020

**ToPeriod** (optional) - if left blank it will be assumed to be the same as the FromPeriod. If you wish to report on a range of periods, enter the last period in the range.

**IncludeFlags** (optional) - if you have utilised User Flags on your CBA GL accounts you may choose to only include accounts which match certain user flags. There are up to nine user flags that can be used and when entering them in this function they are separated by a semi colon.

User Flags have been set up where by an X in position 1 indicates regular local sales, X in position 2 indicates seasonal local sales, X in position 3 indicates regular export sales and X in position 4 = seasonal export sales. Sales Accounts fall in the range of 1000 to 1099.

To get a balance for only sales accounts flagged as "regular local sales" you would enter  
From Account = \*.\*.1000  
ToAccount = \*.\*.1099  
IncludeFlags = X

To get a balance for only sales accounts flagged as "seasonal local sales" you would enter  
From Account = \*.\*.1000  
ToAccount = \*.\*.1099  
IncludeFlags = ;X

To get a balance for sales accounts flagged as "seasonal sales" (being both local and export) you would enter  
From Account = \*.\*.1000  
ToAccount = \*.\*.1099  
IncludeFlags = ;X;;X

**ExcludeFlags** (optional) - if you have utilised User Flags on your CBA GL accounts you may choose to exclude accounts which match certain user flags. There are up to nine user flags that can be used and when entering them in this function they are separated by a semi colon.

**BalType** (optional) - if left blank it will be assumed to be "V", meaning value. If you wish to return the quantity enter a "Q".

**FCBal** (optional) - if left blank it will be assumed to be FALSE, meaning it will return the ledger (base) currency value. If you wish to return the value in the currency of the account enter TRUE.

## 4.6 cbaGLCurPeriod

This function returns the current period number as set in CBA via GL Company Maintenance.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

## 4.7 cbaGLCurYear

This function returns the current year number as set in CBA via GL Company Maintenance.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

## 4.8 cbaGLName

This function returns the description of the GL account as defined in CBA via Account Maintenance. In CBA the account description is held in capital letters but this function will return the description using upper and lowercase i.e. normal sentence caps.

The function arguments are

**Company** (mandatory) - a valid CBA Company Code. If your company code is numeric you need to enter it in text delimiters, e.g. "01"

**Account** (mandatory) - a valid GL account number. Wildcards are not allowed to be used as this argument must refer to a single account.

Account MAS.AK.SLS.1010 is defined as having a description of SALES - DESKS & CHAIRS, along with a short code of SLS and balance sign of CR.

`cbaGL("01","mas.ak.sls.1010")` returns Sales - Desk & Chairs



*If you prefer to have the account description returned exactly as it appears in CBA, i.e. all caps, use the **cbaGL** function, with a *FieldName* of Desc.*

## 4.9 cbaHide

This function is used to control the hiding of rows that contain all zero values. It can also control whether or not a total row should be hidden based on whether the previous range is all hidden.

The function arguments are

**Name** (mandatory) - a unique name that will identify this range. The name may be alphanumeric and may be just a single character.

**HideRange** (mandatory) - a range of cells that will be checked for rows that contain all zero values. Any rows that do contain all zero values will be hidden when the option **Hide Zeroes** is selected from the CBA menu on the Excel menu bar.

**TotalsRange** (optional) - if you wish to hide a total row, based on whether all rows in the HideRange are hidden, enter the range of cells that make up this total row.

**In the following example rows 2 & 3 will be hidden. If all three rows (2, 3, & 4) contained all zeroes then all 3 rows would be hidden and row 6, containing the totals, would also be hidden.**

IF =cbaHide(1,A2:D4,A6:D6)

	A	B	C	D	E	F	G
1		Current	YTD	Budget			
2	Account 1	0	0	0			
3	Account 2	0	0	0			
4	Account 3	2500	20000	25000			
5							
6	Total	2500	20000	25000			

**Function Arguments**

cbaHide

**Name** 1 = 1

**HideRange** A2:D4 = {"Account 1",0,0,0;"

**TotalsRange** A6:D6 = {"Total ",2500,20000(

= "Hide is Off"

## 5 Security

CBAXLink does not have any awareness of the CBA user security and therefore does not do any validation of access to the database.

Any restriction of access must be done via Excel in the form of password protecting workbooks, individual worksheets or cells.