EPL - Expert PIMS Language

V 2.0

Technical Reference Manual

Introduction

This module comes as an extension of the PIMS2 business software. Its aim is to allow various peripheral devices and software applications to retrieve business data directly from PIMS2.

EPL can be used in e-mails, in mobile SMS as well as in Excel to query PIMS2 business application. Usages vary from retrieval of simple business information like account balance or stock position to the writing of complex analytical or consolidated reports. And while the structural form of the query itself may vary from one device to the other, its functionalities and commands remain the same whatever the hosting application or device may be.

EPL, in many aspects, can be considered as a small script language that consists of commands and parameters. These commands were kept short on purpose in order to allow for fast writing specially when using devices such as mobile SMS.

This reference manual will focus mainly on the usage of EPL with Microsoft Excel. Other usages of EPL will be described in separate appendices and will be added in future releases of this manual.

One last word before getting technical: EPL is in **permanent evolution**. New features and functionalities are often added to EPL. Please make sure you get the latest version of this manual
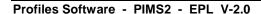
EPL Requirements

EPL is an option of PIMS2 business software, so first you have to make sure that this option is installed in your system before being able to use it.

Also, it is important to note that whatever the device using EPL to query PIMS2, the latter should be running in order to be able to process the received queries.

Other technical requirements vary from one device to the other. Since our current focus is on Excel, Microsoft Excel version 97 or above must be installed on the PC.

Finally, we assume that the user is familiar with the usage of Microsoft Excel.

Hint: *In Excel, it is necessary to have the following setting activated: from Tools\options\calculation page, make sure the Option "Automatic" is checked.*

EPL Syntax

As mentioned above, an EPL query is a sequence of commands and options that explain to PIMS what information is required. Actually we have:

- one module identity

- one or more commands (usually one)

- one or more parameters pertinent to the command issued

In the following descriptions of EPL commands:

- compulsory parameters will be enclosed in angle brackets **< >**

- optional one will be enclosed in square brackets **[ ]**

Do not type these brackets in your commands. They are only shown in this document for clarity.

```
Module , Command1 , [Command2] , [Param1] , [Param2] , …
```

The Module directive is used to specify the "field of interest" for which subsequent command will be issued. Since EPL can be used to get any kind of information throughout PIMS2, this directive specifies *Company, Account, Currency, Stock-Item, etc…*

In Excel, we will use a variant of this syntax. Any cell may contain an EPL query, the same way it may contain regular Excel formulas. In each EPL cell, the query will be expressed as a "PIMS formula", which in turn contains the EPL script:

```
="PIMS(Module, Command1, [Command2], [Param1], [Param2], …)"
```

Note that, unlike a standard Excel formula, the EPL formula is embraced between two quotation marks. These are compulsory and prevent Excel from trying to interpret the formula by itself.

EPL Parameters:

Before moving on to the list of EPL commands, a word about the [*Params*] contained in the formula.

Each parameter may be either expressed literally, or may refer to another Excel named cell by its name.

For example: if one of the parameters is a date, the command can be issued in the following way:

="PIMS(ACT, BAL, 53, **31/12/2002**)"

or alternatively, we can issue it with a named parameter:

="PIMS(ACT, BAL, 53, **@mydate**)"

where **mydate** is the name given to another cell on the Excel sheet or book. Of course, this cell would contain a date as expected by the parameter being replaced.

Any parameter can be replaced by a named cell if necessary.

Complex cell-name parameters can be used too: EPL accepts such expression as:

@mycell1 + @mycell2	in this case the command will be applied twice, first for @mycell1 and second for @mycell2, then both results will be added together as the result of the command
@mycell1 .. @mycell2	in that case the command will be applied for the entire range extending from @mycell1 till @mycell2, all results will be added together as the result of the command

PROFILES SOFTWARE

EPL Commands

Generic commands

`PIMS(COMP, NAME, [#CompCode])`

Returns the full name of the company PIMS2 is currently logged on.

[#CompCode]: If this optional parameter is provided as a valid code for a known company in PIMS2, the full name of that company will be returned instead.

Note that the hash sign in front of the company code parameter is compulsory

---

`PIMS(COMP, ADR, [#CompCode])`

Same as above, but returns the full address of the company

[#CompCode]: same as previously

`PIMS(CUR, NAME, <CODE>)`

Returns the full name of the currency defined as *Code* in PIMS. Since currencies definition is common across the system, specifying a company is not supported. Both main currency code and international code are acceptable (see PIMS help for more details).

`PIMS(CUR, RATE, <CODE>, [Date])`

Returns the rate of currency specified in the compulsory parameter *Code*. Both main currency code and international code are acceptable (see PIMS help for more details)

[Date]: If parameter is provided, rate will be as of that date. If omitted, current date will be considered as default. Format for this parameter: dd/mm/yyyy

---

`PIMS(CUR, MINRATE, <CODE>)`

Returns the smallest rate available in PIMS for currency *Code*. Reminder: PIMS considers the minimum rate over a period covering the past 12 months (see PIMS help file). Both main currency code and international code are acceptable.

`PIMS(CUR, MAXRATE, <CODE>)`

Same as above, but returns the largest rate available.

`PIMS(TAFQIT, <LANG>, <AMOUNT>, <CURRENCY>)`

Returns amount and currency expressed in full words. Parameter *<LANG>* specifies the language used for expressing the result, values acceptable so far are: **AR** (Arabic), **FR** (French), **EN** (English).

<CURRENCY> is the code of currency related to the amount.

PROFILES
SOFTWARE

User input commands

Sometimes, an ad-hoc information is necessary for processing the EPL template. This information is rarely the same from one processing to the other, and needs to be fed by the operator each time the template is processed according to the needs. Such information would typically be a "starting date", a "ratio" or a "rate"..

Instead of opening the EPL template, change all values that need editing, and save the template again, a function has been provided to enforce the user to fill in the relevant information required for the template processing

```
PIMS(ASKUSER, <MESSAGE>, <DATATYPE>)
```

Will pause processing and expect a user input. Note that all similar functions throughout the template will be grouped and result in one single input form only, with multiple values to be fed by the user.

<MESSAGE>: Free explanatory text clarifying to the user what is the role of value being requested.

<DATATYPE>: type of value being requested. To be chosen among

- DATE: The user would be forced to input a date
- AMOUNT: The user would be forced to input an amount with 2 decimals *(not implemented yet)*
- COUNT: The user would be forced to input a value with no decimals *(not implemented yet)*
- *more data types to be expected soon*

Accounting, properties commands:

`PIMS(ACT, NAME, <CODE>, [#CompCode])`

Returns the full name of chart account specified in the compulsory parameter *Code*.

[*#CompCode*]: same as previously.

`PIMS(ACT, ALIAS, <CODE>, [#CompCode])`

Returns the <u>Alias</u> code of chart account specified in the compulsory parameter *Code*. If this code is not available, *Code* itself is returned. More information about account aliases in PIMS help.

[*#CompCode*]: same as previously.

`PIMS(AUX, NAME, <CODE>, [#CompCode])`

Same as above, but applicable to auxiliary accounts.

`PIMS(AUX, ADR, <CODE>, [#CompCode])`

Returns the full address of auxiliary account specified in the compulsory parameter *Code*.

[*#CompCode*]: same as previously.

`PIMS(AUX, PROPERTY, <CODE>, <PROPERTY>, [#CompCode])`

Returns the value of auxiliary property specified in the compulsory parameters *Code* and *Property*.

[*#CompCode*]: same as previously.

**PROFILES
SOFTWARE**

Accounting, balances commands:

```
PIMS(ACT, BAL, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Returns the balance of chart account specified in the compulsory parameter *Code*. If balance is debiting, value returned will be negative.

Complex *<CODE>* allowed: Often you may want to compute a result for a combination of codes instead of doing it for a single code. Complex codes are enclosed between square brackets and may contain additions of accounts as well as ranges of accounts. Additions are marked with the "**+**" sign while ranges are designed by two consecutive dots "**..**". For example:

PIMS (ACT, BAL, [53+51]) will return the combined balance of both accounts contained in the brackets.

PIMS (ACT, BAL, [51..58]) will return the combined balance of all accounts between 51 and 58 inclusive. In case these accounts are not terminal accounts, only terminal accounts that are parented by accounts between 51 and 58 inclusive will be included in the balance computation.

PIMS (ACT, BAL, [51 + 55..58]) concatenation of complex codes are also allowed.

[*Date*]: If parameter is provided, balance will be as of that date. If omitted, current date will be considered as default. Format for this parameter: dd/mm/yyyy.

In case a **future** date is provided, the function returns the **forecast** balance of the account. Any future transaction stored in the scheduler with a date within the provided parameter, and involving this account, will be taken into consideration for balance computation.

[*Currency*]: if provided, balance will be restricted to movements in the currency specified. If omitted, global balance (all currencies intermixed) will be provided as one single counter-value. Default counter-value will be provided in PIMS2 base currency. Format for this parameter is any valid PIMS2 currency code.

[*CounterValue*]: Currency of counter-value. If provided, balance will be expressed in that currency. If omitted, base currency of PIMS2 system will be used for conversion. Format for this parameter is any of two values: **CVLOCAL** or **CVBASE**. Other values specific to your PIMS2 may be accepted. For example if your base or local currencies are the Euro or Saudi Rial, acceptable extra values are **CVEUR** or **CVSAR**

[*#CompCode*]: same as previously.

```
PIMS(ACT, BALIFDB, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but balance is returned **only if debiting**. Otherwise a value of 0 (zero) is returned. Future dates are also supported as explained above.

```
PIMS(ACT, BALIFCR, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but balance is returned **only if crediting**. Otherwise a value of 0 (zero) is returned. Future dates are also supported as explained above.

```
PIMS(ACT, BALVAL, <CODE>, [Dateval], [Currency], [CounterValue], [#CompCode])
```

Same as *PIMS(ACT, BAL, ..)* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

```
PIMS(ACT, BALVALIFDB, <CODE>, [Dateval], [Currency], [CounterValue], [#CompCode])
```

Same as *PIMS(ACT, BALIFDB, ..)* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

PROFILES SOFTWARE

```
PIMS(ACT, BALVALIFCR, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as *PIMS(ACT, BALIFCR, ..)* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

```
PIMS(ACT, BALDB, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but value returned is the sum of all Debiting balances related to this account only.

In other words, if **ACT** is a grouping account, the value returned is the sum of all accounts grouped on **ACT** and having debiting balances.

If on the other hand **ACT** is a father account, the value returned will be the sum of all its terminal children having debiting balances.

```
PIMS(ACT, BALCR, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but value returned is the sum of all Crediting balances related to this account only.

In other words, if **ACT** is a grouping account, the value returned is the sum of all accounts grouped on **ACT** and having crediting balances.

If on the other hand **ACT** is a father account, the value returned will be the sum of all its terminal children having crediting balances.

```
PIMS(AUX, BAL, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but applicable to auxiliary accounts.

```
PIMS(AUX, BALIFDB, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but applicable to auxiliary accounts.

```
PIMS(AUX, BALIFCR, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but applicable to auxiliary accounts.

```
PIMS(AUX, BALVAL, <CODE>, [Dateval], [Currency], [CounterValue], [#CompCode])
```

Same as *PIMS(AUX, BAL, ..)* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

```
PIMS(AUX, BALVALIFDB, <CODE>, [Dateval], [Currency], [CounterValue], [#CompCode])
```

Same as *PIMS(AUX, BALIFDB, ..),* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

```
PIMS(AUX, BALVALIFCR, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as *PIMS(AUX, BALIFCR, ..)* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

```
PIMS(AUX, BALDB, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but applicable to auxiliary accounts. If **AUX** is a father account, the value returned will be the sum of all its terminal children having debiting balances.

PROFILES SOFTWARE

```
PIMS(AUX, BALCR, <CODE>, [Date], [Currency], [CounterValue], [#CompCode])
```

Same as above, but applicable to auxiliary accounts. If **AUX** is a father account, the value returned will be the sum of all its terminal children having crediting balances.

```
PIMS(ACT, FRCST, <CODE>, [Date], [CounterValue], [#CompCode])
```

Returns the **forecast** balance of chart account specified in the compulsory parameter *Code.*

This function should not be used any more. It is kept for backward compatibility*.* You should use **BAL** function instead, and provide a future date in the parameters.

```
PIMS(AUX, FRCST, <CODE>, [Date], [CounterValue], [#CompCode])
```

Same as above, but applicable to auxiliary accounts.

If **AUX** is a father account, the forecast value returned will be the result of forecasting all its transactable children.

This function should not be used any more. It is kept for backward compatibility*.* You should use **BAL** function instead, and provide a future date in the parameters.

Accounting, movements commands:

```
PIMS(ACT, MVT, <CODE>, <Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
    [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
    [JTjournaltypeFrom[:journaltypeTo]])
```

Returns the movement between two dates for chart account specified in parameter <CODE>. If movement is debiting, value returned will be negative.

Complex **<CODE>** allowed: same as previously

<Date1>: Compulsory parameter. It expresses the start date for the period over which movement of account will be studied. Format for this parameter: dd/mm/yyyy

[Date2]: It expresses the end date for the period over which movement will be studied. If parameter is omitted, current date be considered as default. Format for this parameter: dd/mm/yyyy

[Currency]: same as previously.

[CounterValue]: same as previously

[#CompCode]: same as previously

[&BranchCodeFrom:BranchCodeTo]: optional, if used will return only the movement of this account when transacted in a branch which falls in this range. A single branch code could be supplied [&BranchCode].

[CCcostcenterFrom:costcenterTo]: optional, if used will return only part of the movement of this account when transacted for a cost-center which falls in this range. A single cost-center code could be supplied [CCcostcenterCode].

[JTjournaltypeFrom:journaltypeTo]: optional, if used will return only part of the movement of this account when transacted in a voucher specifically issued with that journal type falling into the supplied range of types. A single journal type code could be supplied [JTjournaltype].

PROFILES
SOFTWARE

```
PIMS(ACT, MVTVAL, <CODE>, [Dateval1],[Dateval2], [Currency], [CounterValue],
     [#CompCode], [JTjournaltypeFrom[:journaltypeTo]])
```

Same as *PIMS(ACT, MVT, ..)* except that the date parameter addresses the **value date** of the transactions instead of the **issuing date**.

```
PIMS(ACT, MVTIFDB, <CODE>, <Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but movement is returned **only if debiting**. Otherwise a value of 0 (zero) is returned

```
PIMS(ACT, MVTIFCR, <CODE>, <Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but movement is returned **only if crediting**. Otherwise a value of 0 (zero) is returned

```
PIMS(ACT, MVTDB, <CODE>, <Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but value returned is the sum of **debiting movements** only, and not the net movement result.

```
PIMS(ACT,MVTCR,<CODE>,<Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but value returned is the sum of **crediting movements** only, and not the net movement result.

```
PIMS(AUX,MVT,<CODE>, <Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but applicable to auxiliary accounts.

```
PIMS(AUX,MVTIFDB,<CODE>,<Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but applicable to auxiliary accounts.

```
PIMS(AUX,MVTIFCR,<CODE>,<Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
     [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but applicable to auxiliary accounts.

```
PIMS(AUX,MVTDB,<CODE>,<Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
     [&BranchCodeFrom[:BranchCodeTo]], [[JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but applicable to auxiliary accounts.

**PROFILES
SOFTWARE**

```
PIMS(AUX,MVTCR,<CODE>,<Date1>, [Date2], [Currency], [CounterValue], [#CompCode],
    [&BranchCodeFrom[:BranchCodeTo]], [CCcostcenterFrom[:costcenterTo]],
    [JTjournaltypeFrom[:journaltypeTo]])
```

Same as above, but applicable to auxiliary accounts.

**PROFILES
SOFTWARE**

Stock, properties commands:

```
PIMS(STK, NAME, ITM, <CODE>)
```

Returns the full name of stock item specified in the compulsory parameter *Code*.

```
PIMS(STK, NAME, DPT, <CODE>)
```

Returns the full name of stock department specified in the compulsory parameter *Code*.

```
PIMS(STK, NAME, GRP?, <CODE>)
```

Returns the full name of stock group specified in the compulsory parameter *Code*. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

Stock, stock-position commands:

```
PIMS(STK, ONHANDQTY, ITM, <CODE>, [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]], [DEPdepartmentFrom[:departmentTo]])
```

Returns the quantity on-hand at a date for stock item(s) code specified in parameter <CODE>.

Complex *<CODE>* allowed: same as previously

[Date]: Optional parameter. It expresses the date at which stock on-hand position is required. If parameter is omitted, current date be considered as default. Format for this parameter: dd/mm/yyyy

 [#CompCode]: same as previously

 [WWwarehouseFrom:warehouseTo]: optional, if used will return on-hand quantity available in warehouses which falls in this range. A single warehouse code could be supplied [WWwarehouseCode]. If not provided, quantity on-hand available in all warehouses will be returned.

[DEPdepartmentFrom:departmentTo]: optional. Even though a single item code belongs to a specific department, and since we're allowing for complex <CODE>, there is a possibility to compute quantity on-hand for multiple items. If this parameter is provided, it will restrict computation to items belonging to the range of departments provided. A single department type code could be supplied [DEPdepartmentCode].

```
PIMS(STK, ONHANDQTY, DPT, <CODE>, [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]])
```

Same as above. Returns the quantity on-hand at a date for stock item(s) belonging to department code specified in parameter <CODE>.

```
PIMS(STK, ONHANDQTY, GRP?, <CODE>, [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]])
```

Same as above. Returns the quantity on-hand at a date for stock item(s) belonging to classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

PROFILES SOFTWARE

```
PIMS(STK, ONHANDVALAVG, ITM, <CODE>, [Currency], [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]], [DEPdepartmentFrom[:departmentTo]])
```

Returns the valuation at average cost of quantity on-hand at a date for stock item(s) code specified in parameter <CODE>.

Complex *<CODE>* allowed: same as previously

[*Date*]: Optional parameter. same as previously

[**Currency**]: can hold any of 'MGTCURR', 'CVLOCAL', 'CVBASE' and will provide valuation in management currency, local currency counter-value, or base currency counter-value respectively. If omitted, default value will be 'CVBASE'. If a complex code is provided, with multiple items, a parameter of 'MGTCURR' will be rejected.

[*#CompCode*]: same as previously

 [*WWwarehouseFrom:warehouseTo*]: same as previously.

[*DEPdepartmentFrom:departmentTo*]: same as previously

```
PIMS(STK, ONHANDVALAVG, DPT, <CODE>, [Currency], [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]])
```

Same as above. Returns valuation at average cost of on-hand quantity for stock item(s) belonging to department code specified in parameter <CODE>.

note that **[Currency]** parameter, if provided, can any of 'CVLOCAL', 'CVBASE' only.

```
PIMS(STK, ONHANDVALAVG, GRP?, <CODE>, [Currency], [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]])
```

Same as above. Returns valuation at average cost of on-hand quantity for stock item(s) belonging to classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

note that **[Currency]** parameter, if provided, can any of 'CVLOCAL', 'CVBASE' only.

```
PIMS(STK, ONHANDVALRPL, ITM, <CODE>, [Currency], [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]], [DEPdepartmentFrom[:departmentTo]])
```

Same as preceding functions, but item(s) valuated at replacement cost.

```
PIMS(STK, ONHANDVALRPL, DPT, <CODE>, [Currency], [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]])
```

Same as preceding functions, but item(s) valuated at replacement cost.

```
PIMS(STK, ONHANDVALRPL, GRP?, <CODE>, [Currency], [Date], [#CompCode],
    [WWwarehouseFrom[:warehouseTo]])
```

Same as preceding functions, but item(s) valuated at replacement cost.

```
PIMS(STK, OWNEDQTY, ITM, <CODE>, [Date], [#CompCode],
    [DEPdepartmentFrom[:departmentTo]])
```

Returns the quantity owned at a date for stock item(s) code specified in parameter <CODE>.

parameters: same as previously

PROFILES
SOFTWARE

```
PIMS(STK, OWNEDQTY, DPT, <CODE>, [Date], [#CompCode])
```

Same as above. Returns the quantity owned at a date for stock item(s) belonging to department code specified in parameter <CODE>.

```
PIMS(STK, OWNEDQTY, GRP?, <CODE>, [Date], [#CompCode])
```

Same as above. Returns the quantity owned at a date for stock item(s) belonging to classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

```
PIMS(STK, OWNEDVALAVG, ITM, <CODE>, [Currency], [Date], [#CompCode],
    [DEPdepartmentFrom[:departmentTo]])
```

Returns the valuation at average cost of quantity owned at a date for stock item(s) code specified in parameter <CODE>.

parameters: see previous ONHANDVALAVG function.

```
PIMS(STK, OWNEDVALAVG, DPT, <CODE>, [Currency], [Date], [#CompCode])
```

Same as above. Returns valuation at average cost of owned quantity for stock item(s) belonging to department code specified in parameter <CODE>.

note that **[Currency]** parameter, if provided, can any of 'CVLOCAL', 'CVBASE' only.

```
PIMS(STK, OWNEDVALAVG, GRP?, <CODE>, [Currency], [Date], [#CompCode])
```

Same as above. Returns valuation at average cost of owned quantity for stock item(s) belonging to classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

note that **[Currency]** parameter, if provided, can any of 'CVLOCAL', 'CVBASE' only.

```
PIMS(STK, OWNEDVALRPL, ITM, <CODE>, [Currency], [Date], [#CompCode],
    [DEPdepartmentFrom[:departmentTo]])
```

Same as preceding functions, but item(s) valuated at replacement cost.

```
PIMS(STK, OWNEDVALRPL, DPT, <CODE>, [Currency], [Date], [#CompCode])
```

Same as preceding functions, but item(s) valuated at replacement cost.

```
PIMS(STK, OWNEDVALRPL, GRP?, <CODE>, [Currency], [Date], [#CompCode])
```

Same as preceding functions, but item(s) valuated at replacement cost.

**PROFILES
SOFTWARE**

Stock, pending stock-position commands:

```
PIMS(STK, QTYONSUPORD, ITM, <CODE>, [Date], [#CompCode], [AUXauxiliaryCode],
     [DEPdepartmentFrom[:departmentTo]])
```

Returns the pending ordered to supplier quantity for item(s) specified by <CODE>.

[*Date*]: Optional parameter. if provided, will limit the quantity computed to orders expected to be delivered within this date (value date of the supplier order). Otherwise, entire pending quantity will be returned.   // not implemented yet //

[*AUXauxilairyCode*]: Optional parameter. if provided, will limit the quantity computed to orders made to supplier specified with this code. Otherwise, orders to any supplier would be taken into account.

other parameters: same as previously

```
PIMS(STK, QTYPCHNOTREC, ITM, <CODE>, [Date], [#CompCode], [AUXauxiliaryCode],
     [DEPdepartmentFrom[:departmentTo]])
```

Returns the quantity purchased, but not received yet for item(s) specified by <CODE>.
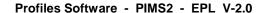
parameters: same as previously

```
PIMS(STK, QTYONCLIORD, ITM, <CODE>, [Date], [#CompCode], [AUXauxiliaryCode],
     [DEPdepartmentFrom[:departmentTo]])
```

Returns the pending client ordered quantity for item(s) specified by <CODE>.

[*Date*]: Optional parameter. if provided, will limit the quantity computed to orders expected to be delivered within this date (value date of the supplier order). Otherwise, entire pending quantity will be returned.   // not implemented yet //

[*AUXauxilairyCode*]: Optional parameter. if provided, will limit the quantity computed to orders made by client specified with this code. Otherwise, orders of all clients would be taken into account.

other parameters: same as previously

**PROFILES
SOFTWARE**

Stock, statistics commands:

```
PIMS(STK, SALQTY, ITM, <CODE>, <Date1>, [Date2], [#CompCode], [&BranchCode],
    [DEPdepartmentFrom[:departmentTo], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the net quantity sold of stock item(s) code specified in parameter <CODE>, for the period specified

Complex *<CODE>* allowed: same as previously

<Date1>: It expresses the start-date at which sales computation must start. Format for this parameter: dd/mm/yyyy

[Date2]: Optional parameter. It expresses the end-date at which sales computation must end. If parameter is omitted, current date be considered as default. Format for this parameter: dd/mm/yyyy

[#CompCode]: If this optional parameter is provided as a valid code for a known company in PIMS2, computation will occur for that specific company. Current company will be considered as default.

[&BranchCode]: If this optional parameter is provided as a valid code for a defined branch in PIMS2, computation will occur for that specific company. All branches will be considered if omitted.

 [SMsalesmanFrom:salesmanTo]: optional, if used will return only part of sales movement that were achieved by a salesman who's code falls in this range. A single salesman code could be supplied [SMsalesmanCode].

 [DEPdepartmentFrom:departmentTo]: optional. Even though a single item code belongs to a specific department, and since we're allowing for complex <CODE>, there is a possibility to compute sales quantity for multiple items. If this parameter is provided, it will restrict computation to items belonging to the range of departments provided. A single department type code could be supplied [DEPdepartmentCode].

[AUXauxiliaryFrom:auxiliaryTo]: optional, if used will return only part of sales movement related to an auxiliary account that falls in this range. A single auxiliary code could be supplied [AUXauxiliaryCode].

[ZNzoneFrom:zoneTo]: optional, if used will return only part of sales movement related to an auxiliary accounts belonging to zones that fall in this range. A single zone code could be supplied [ZNzoneCode].

```
PIMS(STK, SALQTY, DPT, <CODE>, <Date1>, [Date2], [#CompCode], [&BranchCode],
    [SMsalesmanFrom[:salesmanTo]], [AUXauxiliaryFrom[:auxiliaryTo]],
    [ZNzoneFrom[:ZoneTo]]))
```

Returns the net quantity sold of stock item(s) belonging to code of department specified in parameter <CODE>, for the period specified

parameters: same as previously

```
PIMS(STK, SALQTY, GRP?, <CODE>, <Date1>, [Date2], [#CompCode], [&BranchCode],
    [SMsalesmanFrom[:salesmanTo]], [AUXauxiliaryFrom[:auxiliaryTo]],
    [ZNzoneFrom[:ZoneTo]])
```

Returns the net quantity sold of stock item(s) for the period specified, belonging to code of classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

parameters: same as previously

PIMS2, Profiles Integrated Management System
support@profiles-software.com

PROFILES SOFTWARE

```
PIMS(STK, SALVAL, ITM, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
    [DEPdepartmentFrom[:departmentTo], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the net sales value of stock item(s) code specified in parameter <CODE>, for the period specified

parameters: same as previously

[Currency]: can hold any of 'CVLOCAL' or 'CVBASE' and will provide valuation in local currency counter-value, or base currency counter-value respectively. If omitted, default value will be 'CVBASE'.

[CCcostcenterFrom:costcenterTo]: optional, if used will return only part of sales movement when transacted using cost-center which falls in this range. A single cost-center code could be supplied [CCcostcenterCode].

[AUXauxiliaryFrom:auxiliaryTo]: optional, if used will return only part of sales movement related to an auxiliary account that falls in this range. A single auxiliary code could be supplied [AUXauxiliaryCode].

[ZNzoneFrom:zoneTo]: optional, if used will return only part of sales movement related to an auxiliary accounts belonging to zones that fall in this range. A single zone code could be supplied [ZNzoneCode].

```
PIMS(STK, SALVAL, DPT, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the net sales value of stock item(s) belonging to department code specified in parameter <CODE>, for the period specified

parameters: same as previously

```
PIMS(STK, SALVAL, GRP?, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the net sales value of stock item(s) for the period specified, belonging to code of classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

parameters: same as previously

```
PIMS(STK, SALVAL, SMAN, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
    [DEPdepartmentFrom[:departmentTo], [AUXauxiliaryFrom[:auxiliaryTo]],
    [ZNzoneFrom[:ZoneTo]])
```

Returns the net sales value achieved by salesman(men) specified in parameter <CODE>, for the period specified

parameters: same as previously

PROFILES SOFTWARE

```
PIMS(STK, SALCOSTAVG, ITM, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
    [DEPdepartmentFrom[:departmentTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the cost of goods sold of stock item(s) code specified in parameter <CODE>, for the period specified. Valuation based on average cost.

parameters: same as previously

[Currency]: can hold any of 'MGTCURR', 'CVLOCAL', 'CVBASE' and will provide valuation in management currency, local currency counter-value, or base currency counter-value respectively. If omitted, default value will be 'CVBASE'. If a complex code is provided, with multiple items, a parameter of 'MGTCURR' will be rejected.

```
PIMS(STK, SALCOSTAVG, DPT, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the cost of goods sold of stock item(s) belonging to department code specified in parameter <CODE>, for the period specified. Valuation based on average cost.

parameters: same as previously

[Currency]: can hold any of 'CVLOCAL' or 'CVBASE' and will provide valuation in local currency counter-value, or base currency counter-value respectively. If omitted, default value will be 'CVBASE'.

```
PIMS(STK, SALCOSTAVG, GRP?, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the cost of goods sold of stock item(s) belonging to code of classification group code specified in parameter <CODE>, for the period specified. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS. Valuation based on average cost.

parameters: same as previously

```
PIMS(STK, SALCOSTAVG, SMAN, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
    [DEPdepartmentFrom[:departmentTo]], [AUXauxiliaryFrom[:auxiliaryTo]],
    [ZNzoneFrom[:ZoneTo]])
```

Returns the cost of goods sold by salesman(men) specified in parameter <CODE>, for the period specified. Valuation based on average cost.

parameters: same as previously

```
PIMS(STK, SALCOSTRPL, ITM, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
    [DEPdepartmentFrom[:departmentTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Same as preceding functions, but cost of goods sold valuated at replacement cost.

```
PIMS(STK, SALCOSTRPL, DPT, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
    [&BranchCode], [CCcostcenterFrom[:costcenterTo]], [SMsalesmanFrom[:salesmanTo]],
    [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Same as preceding functions, but cost of goods sold valuated at replacement cost.

PROFILES SOFTWARE

```
PIMS(STK, SALCOSTRPL, GRP?, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
     [&BranchCode], [CCcostcenterFrom[:costcenterTo]], [SMsalesmanFrom[:salesmanTo]],
     [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Same as preceding functions, but cost of goods sold valuated at replacement cost.

```
PIMS(STK, SALCOSTRPL, SMAN, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
     [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
     [DEPdepartmentFrom[:departmentTo], [AUXauxiliaryFrom[:auxiliaryTo]],
     [ZNzoneFrom[:ZoneTo]])
```

Same as preceding functions, but cost of goods sold valuated at replacement cost.

```
PIMS(STK, PCHQTY, ITM, <CODE>, <Date1>, [Date2], [#CompCode], [&BranchCode],
     [DEPdepartmentFrom[:departmentTo], [AUXauxiliaryFrom[:auxiliaryTo]],
     [ZNzoneFrom[:ZoneTo]])
```

Returns the net quantity purchased of stock item(s) code specified in parameter <CODE>, for the period specified

parameters: same as previously

```
PIMS(STK, PCHQTY, DPT, <CODE>, <Date1>, [Date2], [#CompCode], [&BranchCode],
     [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]]))
```

Returns the net quantity purchased of stock item(s) belonging to code of department specified in parameter <CODE>, for the period specified

parameters: same as previously

```
PIMS(STK, PCHQTY, GRP?, <CODE>, <Date1>, [Date2], [#CompCode], [&BranchCode],
     [SMsalesmanFrom[:salesmanTo]], [AUXauxiliaryFrom[:auxiliaryTo]],
     [ZNzoneFrom[:ZoneTo]])
```

Returns the net quantity purchased of stock item(s) for the period specified, belonging to code of classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

parameters: same as previously

```
PIMS(STK, PCHVAL, ITM, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
     [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
     [DEPdepartmentFrom[:departmentTo], [AUXauxiliaryFrom[:auxiliaryTo]],
     [ZNzoneFrom[:ZoneTo]])
```

Returns the net purchase value of stock item(s) code specified in parameter <CODE>, for the period specified

parameters: same as previously

```
PIMS(STK, PCHVAL, DPT, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
     [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
     [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the net purchase value of stock item(s) belonging to department code specified in parameter <CODE>, for the period specified

parameters: same as previously

**PROFILES
SOFTWARE**

```
PIMS(STK, PCHVAL, GRP?, <CODE>, <Date1>, [Date2], [Currency], [#CompCode],
     [&BranchCode], [CCcostcenterFrom[:costcenterTo]],
     [AUXauxiliaryFrom[:auxiliaryTo]], [ZNzoneFrom[:ZoneTo]])
```

Returns the net purchase value of stock item(s) for the period specified, belonging to code of classification group code specified in parameter <CODE>. Note: the '?' sign can be any of 1, 2, 3, 4, or 5, representing the five classification groups available in PIMS.

parameters: same as previously

PROFILES SOFTWARE

Sample EPL commands:

The following is a list of sample commands and brief description of the returned value.

PIMS(COMP, NAME)	Full name of current company
PIMS(COMP, NAME, #TRADE01)	Full name of company code TRADE01
PIMS(ACT, NAME, 53)	Full name of chart account 53 in current company
PIMS(AUX, ADR, C0001, #TRADE01)	Full address of auxiliary C0001 extracted from company TRADE01 whatever current company PIMS2 is logged on
PIMS(ACT, BAL, 53)	consolidated balance of chart 53 as of current date in base currency counter-value.
PIMS(ACT, BAL, 53, CVBase)	consolidated balance of account 53 as of current date in base currency counter-value.
PIMS(ACT, BAL, 53, CVUSD)	consolidated balance of account 53 as of current date in base currency counter-value (here base is USD)
PIMS(ACT, BAL, 53, CVLocal)	consolidated balance of account 53 as of current date in local currency counter-value.
PIMS(ACT, BAL, 6 + 7, CVLocal)	Consolidated combined balance of classes 6 and 7 as of current date in local currency counter-value
PIMS(ACT, BAL, 53, CVLBP)	consolidated balance of account 53 as of current date in local currency counter-value (here local is LBP)
PIMS(ACT, BAL, 53, #TRADE01)	consolidated balance of chart 53 as of current date in base currency counter-value. Balance to be extracted from company TRADE01 whatever current company PIMS2 is logged to
PIMS(ACT, BAL, 53, USD)	balance of chart 53 as of current date for transactions in USD only
PIMS(ACT, BAL, 53, YEN)	balance of chart 53 as of current date for transactions in YEN only
PIMS(ACT, BAL, 53, USD, 31/12/2001)	balance of chart 53 as of 31/12/2001 for transactions in USD only
PIMS(ACT, BALIFDB, 53, 31/12/2001, CVLocal)	consolidated balance of account 53 as of 31/12/2001 in local currency counter-value, if this balance is debiting. Otherwise zero value.
PIMS(AUX, BAL, C0001)	consolidated balance of auxiliary C0001 as of current date in base currency counter-value.
PIMS(AUX, BAL, C0001, USD)	balance of auxiliary C0001 as of current date for transactions in USD only.
PIMS(AUX, BAL, C0001, 31/12/2001)	consolidated balance of auxiliary C0001 as of 31/12/2001 in base currency counter-value.
PIMS(AUX, BAL, C0001..C9999, 31/12/2001)	consolidated combined balance of all terminal auxiliaries between C0001 and C9999 as of 31/12/2001 in base currency counter-value
PIMS(AUX, BAL, C0001, 31/12/2001, #TRADE01)	consolidated balance of auxiliary C0001 as of 31/12/2001 in base currency counter-value. Balance to be extracted from company TRADE01 whatever current company PIMS2 is logged to

PIMS(ACT, MVT, 53, 01/01/2001, 31/12/2001)	consolidated movement for chart account 53 for the year 2001 in base currency counter-value
PIMS(ACT, MVT, 53, CVLOCAL, 01/01/2001, 31/12/2001)	consolidated movement for chart account 53 for the year 2001 in local currency counter-value
PIMS(ACT, MVTDB, 53, 01/01/2001, 31/12/2001)	consolidated debiting movement for chart account 53 for the year 2001 in base currency counter-value
PIMS(ACT, MVT, 53, CVLOCAL, 01/03/2003)	consolidated movement for chart account 53 from beginning of March 2003 and until current date in local currency counter-value
PIMS(ACT, MVT, 53, CVLBP, 01/01/2001, 31/12/2001)	consolidated movement for chart account 53 for the year 2001 in local currency counter-value (here local is LBP)
PIMS(ACT, MVT, 53, 01/01/2001, 31/12/2001, #TRADE01)	consolidated movement for chart account 53 for the year 2001 in base currency counter-value. Movement to be extracted from company TRADE01 whatever current company PIMS2 is logged to
PIMS(ACT, MVTIFCR, 53, 01/01/2001, 31/12/2001)	consolidated movement for chart account 53 for the year 2001 in base currency counter-value, if this movement is crediting. Otherwise zero value.
PIMS(AUX, MVT, C0001, 01/01/2001, 31/12/2001)	consolidated movement for auxiliary C0001 for the year 2001 in base currency counter-value
PIMS(AUX, MVT, C0001, @firstdate, CVLOCAL)	consolidated movement for auxiliary C0001 from starting date as per cell named "firstdate" and until now in local currency counter-value
PIMS (ACT, MVT, 53, 01/01/2012, 04/04/2012, USD, CCA1:A4)	consolidated movement for chart account 53 for the year 2012 in base currency counter-value. Movement to be extracted where only cost centers A1 to A4 are involved.
PIMS (ACT, MVT, 53, 01/01/2012, 04/04/2012, USD, JTFISCAL:REAL)	consolidated movement for chart account 53 for the year 2012 in base currency counter-value. Movement to be extracted where only journal types FISCAL to REAL are involved.

Using dates and date formats:

Excel will comply with the regional settings you've instructed Windows to use. You can tell Windows what date format to use by opening your control panel and selecting the "Regional and Language options". Whatever format you specify for the display of dates, Excel will abide by this format setting, and display dates accordingly.

However, due to temporary technical limitations, the current version of EPL is unable to comply with all possible formats, and accepts the "**dd/mm/yyyy**" date format **only**. The "*dd*" part represents the day, "*mm*" the month and "*yyyy*" the year expressed in four digits. Any attempt to run EPL with other date formats may result in error messages or even yield wrong results.

Also be aware that an EPL Excel sheet prepared on a PC that complies with this format will not be processed if copied onto another PC with different date format settings.

We are currently working on lifting this limitation.

**PROFILES
SOFTWARE**