Core PAMS

By P. Schwörer

ABOUT PAMS

PAMS is a proprietary accounting system framework written in VBA. PAMS is used to power a variety of high-profile portfolio accounting systems, from hedge funds to private bank accounts.
PAMS is designed to be modular, with a number of optional modules included in the ‚core' PAMS package. These modules serve to extend and enhance the capabilities of PAMS core.

Since the behaviour of the PAMS framework is driven by its securities database, its chart of account and the active set of user defined accounting rules, PAMS can be used to manage just any type of portfolios, from a single accounting unit with underlaying trial balance (typically a fund) to accounting systems with thousands of portfolios accounts in a typical banking environment.

This PAMS Reference card provides useful information associated with the setup of a new PAMS environment, and focuses on PAMS 6, the latest mayor release available as of this writing.

PAMS ENVIRONMENT BUILDING

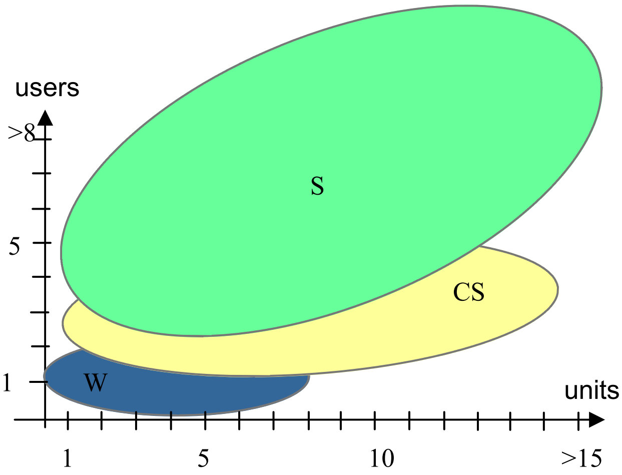
PAMS is architecture-independent, it only requires Windows based workstations running XP, Vista or Windows 7. Even if PAMS is run mostly on MS Servers, depending on your hardware, the accounting units you have to manage, and the number of PAMS users, you can setup one of the following IT infrastructures:

1. **Single clients with distribued database**
   Install PAMS on a workstation and share the folder you choose as data-folder.

2. **Client-Server**
   Just install any file or database server of your choice and define a shared folder or a database. Windows clients have to have write access to the shared folder or database.

3. **Server**
   Install a Windows Server (2003+) with the appropriate number of TSE and Microsoft Office (2003+) Licences (one for each PAMS user and one for the admin account).

If you want to install the Webreporting Plug-in (see later), please also install a WAMP server (Apache, MySQL, PHP) with ftp capability, on the same server or at an ISP.



This chart shows the typical architetures for different user/accounting-units values.
Once the choosen infrastructure as been defined, you just have to install PAMS on every workstation or on the server in the user personal folders and the default database files on a common area where users have write access authorization.

Tip

In few minutes it is possible to switch between configurations, just copying files and changing few parameters.

When you start PAMS the database engine checks the integrity of the database system and, if necessary, the user is prompted to define (browse) where the logical files are located.

In PAMS the database engine builds a virtual database at run time. So every user can run with different set of parameters. This feature seems to be complicated but allows fast switching between configurations and mixed environments.

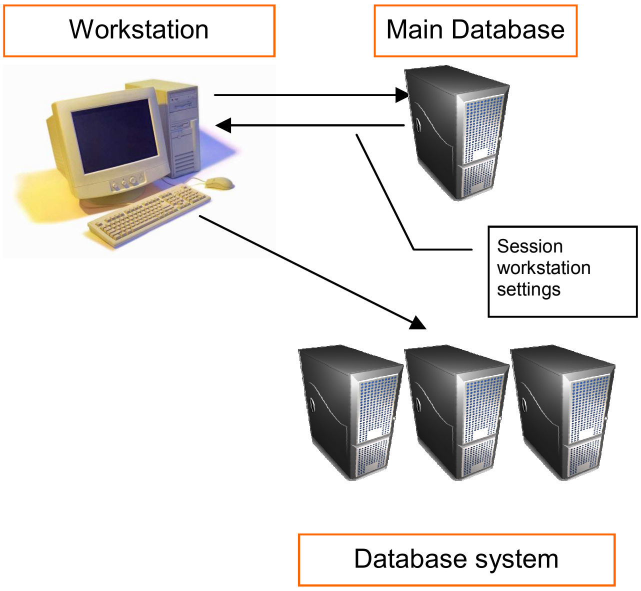
PAMS RUN TIME DATABASE BUILDING

The database engine needs six logical links:

Securities Database
Quoting Database
Main Database
Trading Database
Transactions Database(s)
Accounting Database(s)

All these logical database links can be combined in only one database or split in up to six different distribuited databases. The references to the last two logicals are stored in the main database.

The build-in database engine first reads the connection settings stored in the user settings (workstation layer) in order to reach the meta-information contained in the *logical* main database. This meta-information contains all the other needed connection settings necessary to build the database



In PAMS you can store your data with different technologies, from Microsoft SQL to open source database system (MySQL, Postgres), Microsoft Access files or even MS Excel spreadsheets, XML files or mixed environments.

Consider to already have in a linked-to-Bloomberg MS Excel Spreadsheet the pricing of some special instruments. With PAMS it is not necessary to write a script that imports this information into the database, you can simply link it into PAMS.

PAMS will then use this information, through its build-in database engine, as it where stored in your main database system.

Tip

Each table in PAMS can be linked to a different database system. This capability should be used with care, so not to lessen performances.

However it is recommended to store your data within a secured database system, and not to spread data on your network in a lot of different formats, even if this is, from a technical point of view, possible.

It is recommended to use this feature carefully, remembering that it is very useful to handle exceptions.

FIRST STEPS

Setting up an hedge fund or a bank account is not quite the same task.

An hedge fund needs a more detailled chart of accounts, a trial balance and a profit & loss analysis, on the other hand a bank account with a portfolio can be managed, from an accountant point of view, in a more simple way.

PAMS disposes of three entities that determine the behaviour of the framework.

1. Chart of accounts
2. Securities settings
3. Accounting rule templates

Some definitions in the securities database need to have a correspondence in the Chart of Accounts, and the accounting rules are based on both previously defined data sets.

For this reason, it is a good practice to begin with the definition of the chart of accounts, then define the structure of your securities database and, as last task to determine the accounting rules.

In order to deepen the learning curve, PAMS disposes of a set of predefined templates of parameters, that can be modified and fine tuned, even in a second time, to fit your needs.

Tip

Start with a preconfigured template containing a chart of accounts, a securities database and a set of accounting rules. Then apply fine tuning modifications to fit your needs.

THE CHART OF ACCOUNT

The chart of account is multi-level & multi-currency in a typical tree structure.

Each defined account disposes of a serie of switches that tell the framework what kind of account it is.

A second horizontal structure groups single accounts to account-types, building a *virtual* chart of accounts, on which are defined the accounting rules and the securities database links.

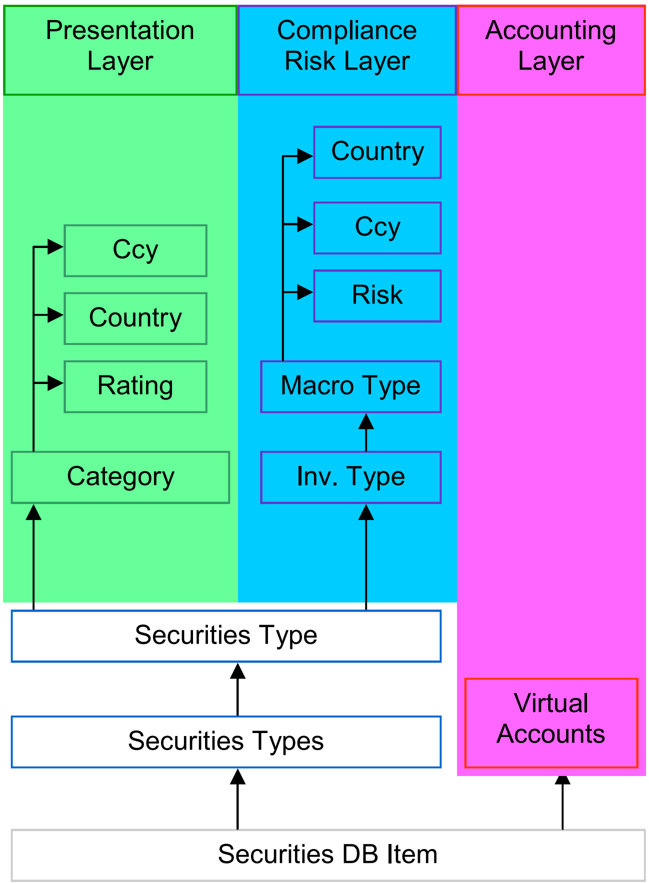
This feature permits to run the same accounting rules on different chart of accounts and lets define your chart of accounts tailored for accountants and autitors, without critical limitations for the other involved professionals (trader, clients, shareholders, etc.) .

THE SECURITIES DATABASE

The securities database stores all types of securities (stocks, options, derivatives, all kind of bonds, ...) and financial instruments (CDO, SWAPS, ...).

As the chart of accounts, it is realized with three multi-level tree structure.

The first tree structure serves for presentation purposes, the second for the risk management and the last for the accounting transaction generator.



Some practical benefits from such a structure:

You can automatically book the accounting records generated from the trading activity with the *realized* split by currency- and by price variations, always on the right P&L accounts of your choice, compute and book the *unrealized* as accrual on every date,

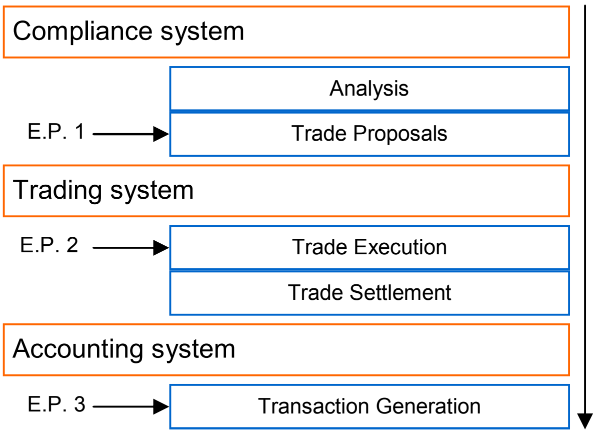
You can valuate and render securities in different ways as you treat risk: list a bond-strategy fund under *Funds* and treat it as *bond* for risk management purposes,

In other words, you will be able to produce the information to every audience (customers, shareholders, accountants, auditors, traders, risk managers) always with the appropriate layout.

THE ACCOUNTING RULES

The Accounting Rules are the *core* of the PAMS framework. They allow to strongly determine the behaviour of PAMS and they manage the internal communication between the different PAMS framework layers.

The three main logical layers of the core framework and its additional modules are:



Depending on the installed modules, a trade can be defined at every allowed entry point.

An accounting rule determines how a trade will be mapped in the accounting database, i.e. the complete list of transaction-entries that the PAMS framework generates and what happens if an information is changed above in the layer stack (when allowed).

A rule in composed by a set of instructions that binds trade informations (such as dates, security, quantity, proceed), securities definitions (involved currencies, accounts, accruals, securities details), broker and/or custodian specific definitions with system defined formulas to build computed amounts that the system use to generate the transaction.

A rule can be defined throught a GUI or as XML, and can have a global scope or be limited to a specific accounting unit and automatically linked to a security type and customer defined operation.

Example:

„Template 3130 applied when SELL BONDS“

tells PAMS to apply a specific template (*3130*) when a *BOND* is *SOLD*.

Taking a look inside the definition of a template of rules, we obtain (in this example we assume a fund accounting system) the list of steps that PAMS executes (in XML):

EXAMPLE OF A RULE TEMPLATE (XML)

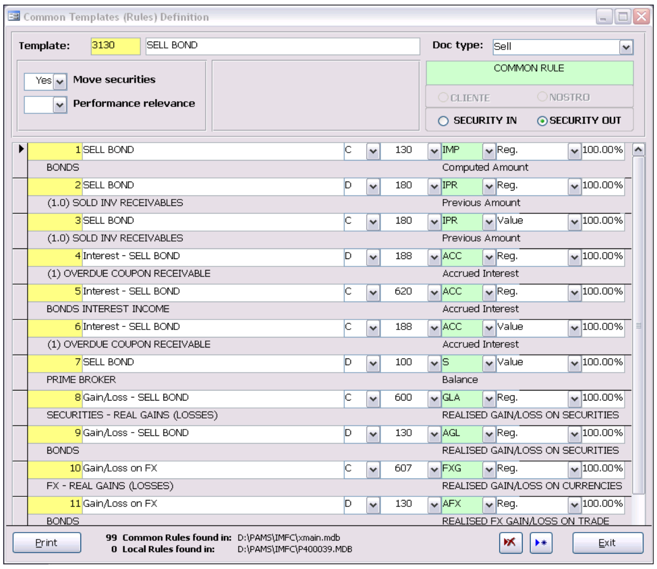
```
<template>
        <name>3130</name>
        <desciption>SELL BOND</description>
        <confirmation_template></confirmation_template>
        <involve _security>YES</involve _security>
        <step>
                <order>1</order>
                <text>BUY BONDS</text>
                <debit_credit>C</debit_credit>
                <account>130</account>
                <formula>IMP</formula>
                <date>REG</date>
                <percent>100</percent>
                <note>
                    C tells PAMS to Credit the amount,
                    130 is the balance sheet account class
                    where bonds are valuated,
                    IMP is the formula that multiply the
                    proceed with the trade quantity,
                    REG is Registration Date
                </note>
        </step>
        <step>
                <order>2</order>
                <text>BUY BONDS</text>
                <debit_credit>D</debit_credit>
                <account>180</account>
                <formula>IPR</formula>
                <date>REG</date>
                <percent>100</percent>
                <note>
                    D tells PAMS to Debit the amount,
                    180 is the balance sheet account of the
                    counterparty (broker),
                    IPR is the formula that propose the last
                    computed amount,
                    REG is Registration Date
                </note>
        </step>
        <step>
                <order>3</order>
                <text>BUY BONDS</text>
                <debit_credit>C</debit_credit>
                <account>180</account>
                <formula>IPR</formula>
                <date>VALUE</date>
                <percent>100</percent>
                <note>
                    C tells PAMS to Credit the broker account
                    in value date,
                    180 is the balance sheet account of the
                    counterparty (broker),
                    IPR is the formula that propose the last
                    computed amount,
                    REG is Value Date
                </note>
        </step>
        ... And so on ...

</template>
```

So it is possible to tells PAMS to:

1. Credit the balance sheet account where Bonds are valued (in regDate),
2. Debit with the same amount the balance sheet account of the broker (in RegDate),
3. Re-Credit the broker account in value date,
4. Calculate and Debit in Reg date the overdue coupon receivable on the right account (in RegDate)
5. Credit this amount to the P&L corresponding account (in regDate)
6. Re-Credit the overdue coupon receivable in Value Date,
7. Compute the balance of the transaction in value date and Credit the (broker-) current account,
8. (eventually, if a long position exists) Credit or Debit the Gain/ Losses in Reg Date on the corresponding P&L account (Realized on security price change),
9. (eventually, if a long position exists) Debit or Credit the Gain/ Losses in Reg Date on the corresponding Account of the balance sheet (Realized on security price change),
10. (eventually, if a long position exists) Debit or Credit the Gain/ Losses in Reg Date on the corresponding P&L account (Realized on crossrates change),
11. (eventually, if a long position exists) Credit or Debit the Gain/ Losses in Reg Date on the corresponding Account of the balance sheet (Realized on crossrates change).

SAME DEFINITION THROUGH THE GUI



Common Templates (Rules) Definition

Template: 3130 SELL BOND

Doc type: Sell

Yes Move securities

Performance relevance

COMMON RULE

○ CLIENTE ○ NOSTRO

○ SECURITY IN ● SECURITY OUT

1	SELL BOND	C	130	IMP	Reg.	100.00%
	BONDS			Computed Amount		
2	SELL BOND	D	180	IPR	Reg.	100.00%
	(1.0) SOLD INV RECEIVABLES			Previous Amount		
3	SELL BOND	C	180	IPR	Value	100.00%
	(1.0) SOLD INV RECEIVABLES			Previous Amount		
4	Interest - SELL BOND	D	188	ACC	Reg.	100.00%
	(1) OVERDUE COUPON RECEIVABLE			Accrued Interest		
5	Interest - SELL BOND	C	620	ACC	Reg.	100.00%
	BONDS INTEREST INCOME			Accrued Interest		
6	Interest - SELL BOND	C	188	ACC	Value	100.00%
	(1) OVERDUE COUPON RECEIVABLE			Accrued Interest		
7	SELL BOND	D	100	S	Value	100.00%
	PRIME BROKER			Balance		
8	Gain/Loss - SELL BOND	C	600	GLA	Reg.	100.00%
	SECURITIES - REAL GAINS (LOSSES)			REALISED GAIN/LOSS ON SECURITIES		
9	Gain/Loss - SELL BOND	D	130	AGL	Reg.	100.00%
	BONDS			REALISED GAIN/LOSS ON SECURITIES		
10	Gain/Loss on FX	C	607	FXG	Reg.	100.00%
	FX - REAL GAINS (LOSSES)			REALISED GAIN/LOSS ON CURRENCIES		
11	Gain/Loss on FX	D	130	AFX	Reg.	100.00%
	BONDS			REALISED FX GAIN/LOSS ON TRADE		

99 Common Rules found in: D:\PAMS\IMFC\xmain.mdb
0 Local Rules found in: D:\PAMS\IMFC\P400039.MDB

The first time PAMS discovers a trade activity with no defined template, it prompts the user to define a new template that will be forthwith stored in the framework database and then applied to all similar trades.

If you modify a rule, PAMS can apply the new definition to already settled trades.

This raffinate engine permits to manage complex trades without any programming or program update.

Obviously PAMS disposes of a set of predefined accounting rule templates for different environments.

TRANSACTIONS

Transactions can be inputted in any order. All trades chains are automatically rebuilt (FIFO, LIFO, AVG) as the corresponding accounting transactions. The engine assures that the correspondence between trades and accounting transactions is always guaranteed.

A trade can then be defined selecting a transaction type, a trade date and a value date, selecting a security, a quantity and a proceed and eventually an applied crossrate. All the rest (posting, calculations of realized and unrealized on security and on crossrates variations, accounts selection, and finally transaction generation) is done by the framework. In case of input errors the framework autocorrects inputted data and in case of doubt the engine prompts the user to select the appropriate account or generates a warning.

Tip Define your trades ‚*as they come*'. PAMS will take in charge all the rest, building its metadata automatically.

With the appropriate permissions, trades can be modified, deleted and partially or totally redefined.

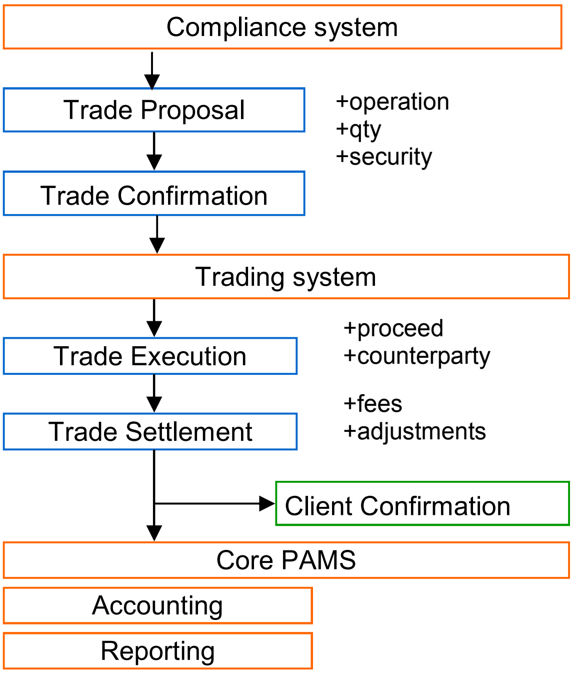
Based on a specific data structure that includes counterparty-, trade place-, and securities- specific data, an analogous engine calculates and books the accounting records for broker-, trading- and bank-fees.
This engine, coupled with the rules templates database, adds intelligence and know-how to the framework, so that even untrained users can input trades that are always consistent with the accounting books.

The PAMS framework and its additional modules build an application multi-level stack. Depending on your business activity, you can define trades at any level.
Every level adds a piece of information to build a valid transaction from a trade. You can jump at any level, just remembering you will need to input the missing information in any of these levels.

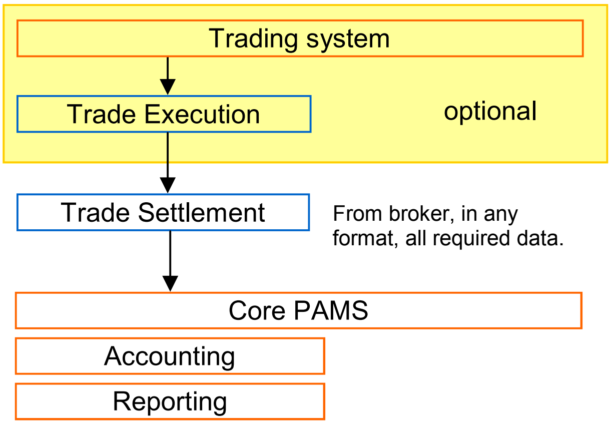
It is also important to note that if you are a bank-like institution, you can let PAMS calculate all the trading and broker fees. If you use PAMS and are ‚*on the other side*' of the trade (fees are debited to you), you will have either to adjust manually the data or import a feed from your broker, bank or institution.

Tip In PAMS you can define and compute fees to charge or to be charged. In the second case is better to reconciliate the fees manually or through a feed.

Example: Bank, managed account



Example: Fund, administration



Tip All additional modules can be installed in a second time, and you can define/ modify/delete trades at any level.

The interesting thing is that in all these different configurations, all the executable code is always the same. The only difference lies in data definitions (accounting rule templates, chart of accounts and additional modules installation).

The PAMS framework is and will always be the same.

QUERY GENERATOR

The PAMS framework disposes of an integrated query generator with graphical user interface. You can then define, store and distribute queries based on tables or on already defined queries, building complex extractions for quick results and easy reporting.

The stored queries are managed by PAMS and you can grant or deny modify or execute rights to every user, or use the queries with external third party report generators.

REPORTING

The PAMS framework disposes of more than hundred reports for portfolio valuations, portfolio analysis, accounting, NAV in different formats for customers, shareholders, fund managers, accountants, auditors and prospects.

ADDITIONAL MODULES

A wide number of additional modules have been added and constantly upgraded since 1994.

Core PAMS includes:

Securities Database
Accounting
Portfolio Valuations
PDF Printing
Query Generator
Reporting

With the core system you can manage a fund, a family office or a set of portfolio accounts.

If you have your own trading room you can add the **Order Trading & Management Module.**

If you need to keep your accounts in accordance with compliance rules, you have to install the **Portfolio-compliance Module.**

The **WebReporting Module** generates any report in batch and automatically dispatch them to a website for easy download (for customer or counterparty). With this module you also can upload in batch data feeds in XMS, TXT, CVS or XLS format.

If you are subjected to an anti-moneylaundering regulation, consider the corresponding module (**FINMA Check Module**).

Additional bank-specific modules calculate margins, trade confirmations, lombard credit checks, etc.

Additional Fund-industry specific modules are the Shareholder Management Module and the Private Equity Module.

For more informations on PAMS, please refer to the following Reference Cards:

PMS-002	Installing PAMS
PMS-003	Setting up PAMS parametrization
PMS-004	Using PAMS
PAM-011	Installing PAMS modules
PAM-012	RTQI Module
PAM-013	PDF Generator
PAM-014	Using Query Generator
PAM-015	Installing and operate WebReporting
PAM-016	Installing a B-Link Ext. Data Feed
PAM-017	Understanding Trade Confirmations
PAM-018	Installing PEM Private Equity Module
PGH-000	Building your own reporting system
PLM-000	PAMSLite vs PAMS

All IBS Reference Cards are downloadable at

http://www.ibsoftware.ch/refcards

For more information on PAMS, visit:

http://www.ibsoftware.ch

where you can find report examples and a well featured online configurator, case studies and typical projects realized from the company that created PAMS.

IBS INNOVATIVE BANKING SOFTWARE

Innovative Banking Software AG is a privately owned swiss company, established in 1986.

Its main product is PAMS and its customers are in Switzerland, Europe, Americas and Far East, ranging from small family offices, institutional banks and Fund Administrators.