Opera 3 Reporter Reports

Training Courseware

Pegasus Software Ltd
Accreditation Training Courseware

February 2011

Copyright © Pegasus Software Limited, 2011

Manual published by:

Pegasus Software Limited
Orion House
Orion Way
Kettering
Northamptonshire
NN15 6PE

www.pegasus.co.uk

All rights reserved. No part of this manual or any accompanying programs may be copied, sold, loaned or in any way disposed of by way of trade or for any kind of profit or incentive without the express permission of Pegasus Software Limited in writing, or as specified in the licence agreement which accompanies this manual and associated programs.

Whilst Pegasus Software Limited takes reasonable steps to ensure that the contents of this manual are up to date, and correctly describe the functioning of the programs, neither Pegasus Software Limited nor any of its Partners give any warranty that the manual is error free, nor that the programs will perform all functions stated in the manual. Pegasus Software Limited has a policy of continual improvement of its software and accordingly this manual may describe features which are no longer available in the current release of the software available to customers, or which are proposed for a future release of the software. Under no circumstances will Pegasus Software Limited or any of its Partners have any liability to any customer arising out of any error, omission or inaccuracy in this manual.

All trademarks are acknowledged.

Pegasus Opera 3 is a trademark.

Accreditation Training Courseware

Contents

Introduction ---------- **II**
About Opera 3 ..... **II**
About this Guide ..... **II**
Introducing the module ..... **II**
Feedback ..... **II**
Module Menu Structure ---------- **III**
Glossary (of Menu Structure) ---------- **IV**
The Module ---------- **1**
1 Overview ---------- **2**
1.1 Modify or Delete **2**
1.2 Create **2**
1.3 Printing **2**
1.4 About the Database **3**
2 Creating & Editing Reports ---------- **4**
2.1 Copying a Report Template **12**
2.2 Report Expressions **13**
2.2.1 Expression Builder 14
3 Running a Report ---------- **18**
3.1 Appending a Report to the Menu **19**
Appendix ---------- **21**
Appendix A – Pegasus Pre-Defined Expressions **21**
FAQS 41
Appendix B - Report Base & Linked Tables **42**
Other Guides and Help ---------- **48**
Readme File **48**
Release Guide **48**
Installation Guide **48**
Demonstration Data Guide **48**
Implementation Guide **48**
System Requirements Guide **48**
Help **48**
Document Management & Executive Dashboards Guides **48**

Accreditation Training Courseware

Introduction

About Opera 3

Opera 3 is the most versatile and user-friendly accounting, supply chain management, payroll & HR, manufacturing and reporting solution from Pegasus Software. A modular solution, Opera 3 enables you to select the components that match your current business model, whilst making it possible to add further functionality as requirements change. This means that the capability or capacity of a system can be seamlessly expanded whilst safeguarding the investment.

For more information about Opera 3, select the Solutions link on the Pegasus web site.

About this Guide

This Accreditation Training Courseware offers an accompanying guide to the accreditation training course.

If you are new to Opera 3, read the Getting Started section of the Opera 3 Help that covers the application in more detail. You can access this from the Windows Start menu, or by using the F1 Key or Help menu when you have logged in to the product.

Opera 3 is available for both Visual FoxPro and SQL Server database platforms. This Accreditation Training Courseware applies to both database platforms, however, there are certain elements that are only included in one or the other database, and as such, the relevant sections or paragraphs are clearly identified with the following labels:

> **[SQL]** – only relevant if you are running the application on an SQL Server database.
>
> **[VFP]** – only relevant if you are running the application on a Visual FoxPro database.

Sections that are not identified with these labels are relevant to both database types.

For more information, see the section Functional Differences between Opera 3 Enterprise & Opera 3 Enterprise SQL section in the release guide.

Introducing the module

Within most modules are standard report commands, most of which provide some criteria selection procedures. Although these standard reports can satisfy most requirements, you may have other reports you want to produce based on the data contained within the database. You can use the Reporter module to design your own reports. The Reporter folder contains two commands:

- Report Manager
- Reports

You use the Report Manager command to create and manage report designs. The Reports command provides a list of all the reports you have designed. You use this list to select and run the reports you want, which extract the information stored in the company database. The report designs you create are available for selection from any of the companies created in the system.

Feedback

Pegasus is interested to hear your views on the documents we provide. To give your feedback, send an email to training@pegasus.co.uk. Please type “O3Training_Accreditation_REP40ReporterReportsR2.0.docx Accreditation Training Courseware” in the subject box.

Please note that product support is not offered through the above email address.

Accreditation Training Courseware

Module Menu Structure

The following sections will cover the functionality of the options in this module as outlined in the image below which shows each menu option contained in this module.

Accreditation Training Courseware

Glossary (of Menu Structure)

Navigator Folder	Command	Purpose
Reporter	Report Manager	Launches the Report Manager screen where new reports may be created or existing ones edited as required.
	Reports	Offers a list of created reports available to run.

Notes

Accreditation Training Courseware

The Module

The Reporter module is designed to allow the user to generate their own reports quickly and easily, and to save the report definition so that it can be printed at any time, and can also be used as an export tool, to pass data to third party products.

The module has the following characteristics:

Ease of Use

By use of cursor keys and mouse control, reports can be designed by making selections from drop-down lists, rather than needing any special programming knowledge or reporting experience. New reports can also be created using existing reports as a template.

Access to All Data

Where there is a logical link between modules, facilities are provided for accessing more than one module's data, on the same report.

Flexibility

Sequences and formats can be defined, and calculations can be done.

Efficient

Reports can be printed off in a batch.

Designs are shared across companies.

Security

Users are only provided with access to these database files, which are compatible with their own menu modules.

Extendibility

The database can be extended to include tables and fields used by third party applications.

The Report Module is accessible from the Opera 3 Navigator and can be protected from use by denying access on a user-by-user basis to this menu item.

When users have restricted access to modules the Reporter module will reflect the restrictions. Reporter will make available only the modules selected in the user profile. If the user has limited access within the modules, the user will see all the tables associated with the module. There is no way to restrict individual tables.

Notes

Accreditation Training Courseware

1 Overview

Overview

To learn about the Reporter module it is best to create your own reports from scratch. However, some example reports have been provided to give you some idea of the type of reports you can produce that are not available as standard reports in the application.

Use the Report Manager command to view or amend the detail of these reports, and the Reports command to produce example output.

Code	Report Name	Explanation
NLBUD	Budget Manager Report	Provides range options for budget manager at runtime (based on Nominal Reporting Codes) and produces a report of accounts and cost centre actual values and budgets on a quarterly basis.
SL1A	Salesmen's Commission	Provides date range selection at runtime and produces a report of sales invoices by representative (based on the Territories codes in the Sales module) with a 10% commission calculation.
SL1B	Sales Transaction List	Provides an account range and transaction type selection at runtime and produces a report of the selected type of

1.1 Modify or Delete

To modify or delete an existing report:

1. Select the Report Manager from the main Naviagtion Tree.
2. From the menu select Report Manager.
3. Select the list tab and choose the report to modify from the list.

1.2 Create

To design a new reports:

1. Select the Report Manager from the main Naviagtion Tree
2. From the menu select Report Manager
3. Select the 'Create New' icon or use CNTRL + N to create a blank report.

1.3 Printing

To print an existing report:

1. Select the Reports Manager from the main Naviagtion Tree.
2. From the menu select Reports.
3. Select the report to print from the list.

1.4 About the Database

Notes

When you design a report, you must identify the data elements in your company database on which the report content is based. Before you start to create your first report design, you may wish to take a little time to learn about the principles of the database, and the terminology we have used in the documentation. The application stores your company information in a database. The database is broken down into module databases that, in general, correspond to the modules you use.

Within each of these modular categories, the main files corresponding to the modules are separated. The selection you make determines what is available, within the design of the report, that is already stored in the database. This is known as the Report Base. You are restricted to the selection of one report base, though you can add custom fields calculated or derived from other data—you cannot create one report design that utilises data from more than one report base. Having selected the report base, you can choose from a list of fields. Each field has a name that identifies the information it holds and they are grouped under headings that represent the tables in the database. In each of the tables are lists of the individual fields that hold the data that comprises the information used in a report design.

Typically, one field represents one column of information on a report, although you are not obliged to include a selected field in the report output; you may be selecting it for another purpose, such as for selection only, or to be used in the calculation of another column of information. Regardless of the report base and table you select, there is always the following available:

Customised Fields

- Formula
- Construct

You select these whenever you want to add a piece of information in your report design that is derived rather than existing in the database. For example, you might want to add a column on a report that calculates a commission as a percentage of an existing sales value. Constructs are similar to formulae except that they apply to text fields, while formulae apply to numeric fields.

.

Notes

Accreditation Training Courseware

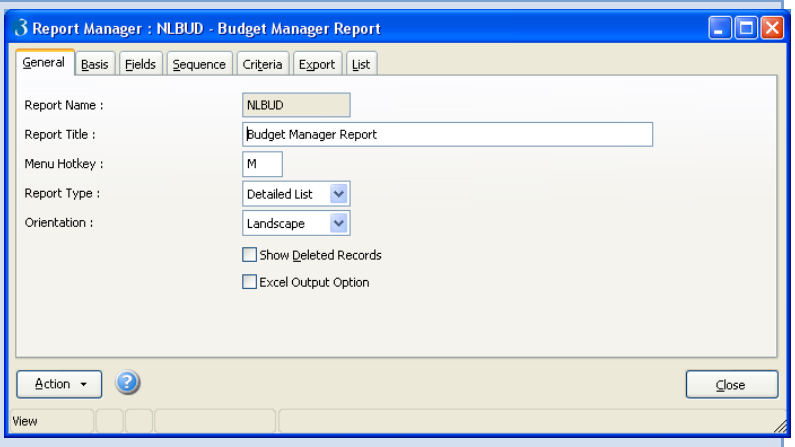
2 Creating & Editing Reports

The creation of report designs involves following a few straightforward steps, some of which are optional. You begin with the Report Manager form. This contains a number of tab dialogs, each covering a stage in the design process.

To Create a Report Design

1. Open the Reporter folder, and then click Report Manager.
2. On the Record menu, click New.
3. On the General tab, complete the boxes provided.
4. Click the Basis tab and choose the content of the report by selecting the appropriate module and files in the Report Base list, and then by double-clicking to select fields in the Fields Available list.
5. Use other tabs on the Report Manager form to refine the report design.
6. To store the report design, on the Record menu, click Save.

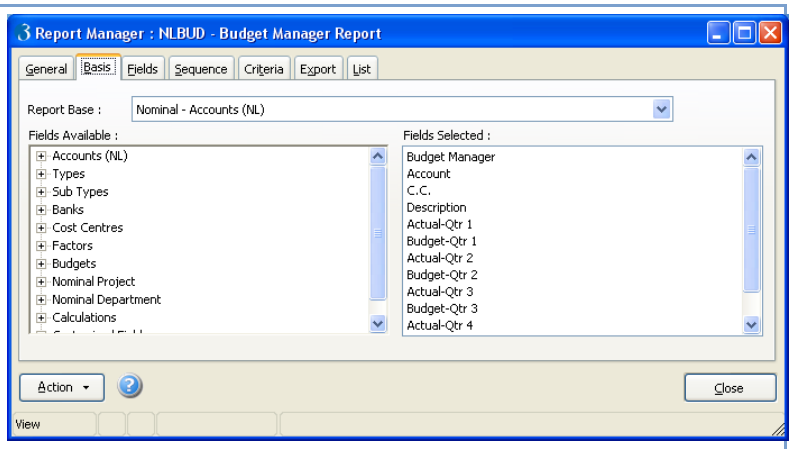
Note: Once you have saved the report design, you can preview the report's content using the **Preview** command button.

Stage	Description	Screen Shot & Notes
Stage 1	The General tab is where you begin. Using the Create a new record button on the toolbar, you can start the process of creating your design, giving each report a unique reference and a title by which it will be recognised subsequently.	 **Report Type:** An option that determines whether the report is printed in detailed or summary form. You can select from a list. If you select Detailed List, individual items, such as transactions that apply to the same master record, will be listed separately in the body of the report. If you select Summary List, such items will be grouped, for example to print a total value of all transactions for the same master record.

Creating & Editing Reports

Notes

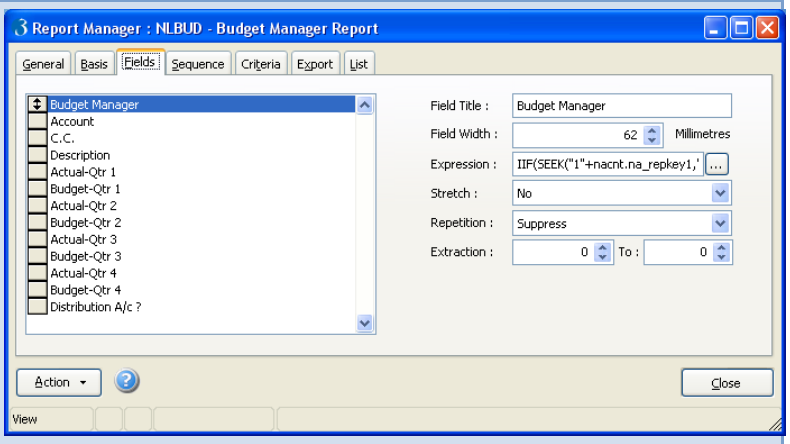
Stage 2 The Basis tab is where you choose the source of data that makes up the columns in your report. You can pick data fields from a wide selection of modules and their tables. You can also include customised fields for calculated values. These first two stages constitute the minimum requirement for a report design. At this stage you can save the design, and even use the Preview command button to see what the report looks like. In the following stages, all of which are optional, you can refine the content of your report.



The Report Base is an option that determines the module and files used as the basis for the selection of fields for inclusion in the report design. You can select from a list. The report base options available depend on the activation code and the user menu access controls assigned to your user profile. This is an important part of the report creation process. Ensuring the appropriate starting point is selected is essential.

The *Customised Fields* option allows the user to create calculated results to include on the report. Selecting 'Formula' will allow you to do numeric calculations, and selecting 'Construct' will allow you to create text expressions.

Stage 3 The Fields tab is where you control the attributes of individual fields selected for the report. Each field represents a column of the report, though you can exclude a field from the output if you only want to use if for calculations, sorting or selection criteria. The Fields tab is also where you define calculated fields.



The Expression field shows the coding that determines the field content. This may be a direct reference to the content of data in the database or a more complex expression of a field based on a formula. You can open the Expression Builder form by clicking the ellipses button to the right of this box.

Note: The message "Expression is too long" is displayed if the expression is longer that 255 characters. This is a limitation in Visual Foxpro, the software development language used to write the application.

Stage 3 – Further field information

Notes

Stretch:For text fields, an option that determines whether the field space within the column should grow large enough vertically to accommodate all text from the field. Otherwise, only a portion of the field will print. This option is especially useful when you're printing data from memo fields. You can select 'Yes' or 'No' from a list.

Format: For numeric fields only, an option that determines the number of decimal places to be used. You can select one of the following from a list: Whole Number, 2 Decimal Places, 4 Decimal Places, 6 Decimal Places or Factored Quantity.

Note: Formatting to decimal places will only truncate the value and will not round the value up or down. For example, 2.346 to 2 decimal places will be given as 2.34.

Repetition: An option that determines whether the contents of a field is duplicated under certain circumstances. For example, if you select the fields for a master record and transactions, the report may print the master record's key reference on every transaction line. This is due to the way in which a transaction is linked to its master record. You can suppress this duplication if required so that, for example, the master record's key appears only once for each set of transactions belonging to that master record. Select Print or Suppress from a list.

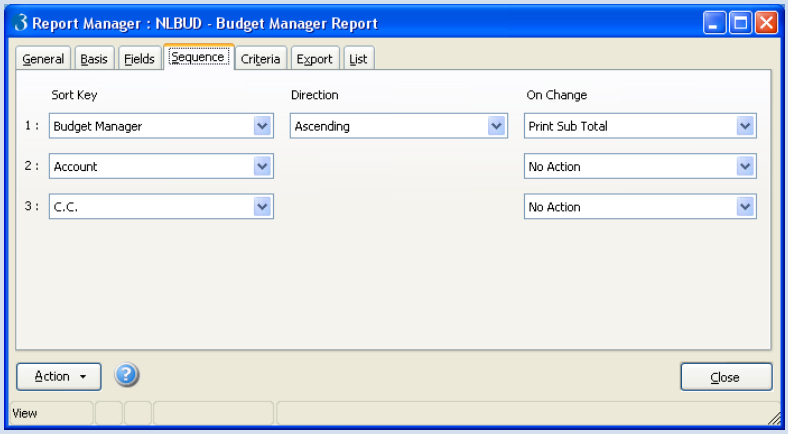
Extraction: For text fields only, you can control which characters within the field are included in the report. Using the two boxes provided, you identify the first and last character positions that you want to be printed. For example, you have selected an account code which may be eight characters long, but you want to report on the basis of only three of the characters, starting with the fourth one. You would enter 4 in the first box and 6 in the second box which is interpreted as meaning you want to use characters 4, 5 and 6.

Total Column: For numeric fields only, an option that determines whether a total is provided for the values in the column on the report. You can select on of the following from a list: No total (meaning that no total is printed for the report column); Count (meaning that a count of the number of records in the column is printed); Sum (meaning that the values in the column are added together for a sub total); Average (meaning that the average of the values in the column is printed); Minimum (meaning that the lowest value in the column is printed) or Maximum (meaning that the highest value in the column is printed).

Date Format: For date fields only, an option that determines the format of the field content. You can select one of the following from a list: DD/MM/YYYY (meaning that the date is printed in numbers, such as 25/10/2003 for 25th October, 2003); Day (meaning that the name of the day is printed, such as Tuesday); Month (meaning that the name of the month is printed, such as January); Year (meaning that the year number is printed, such as 2003) or Age (meaning that the date in the report column is compared with the system date and the difference, in days, is printed).

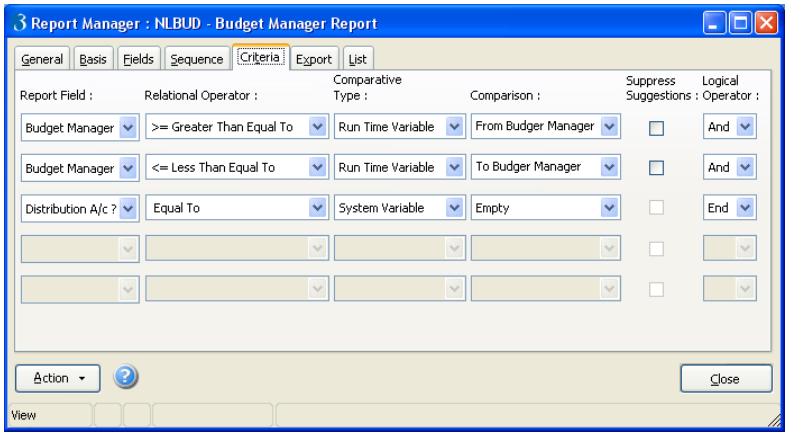
Notes

Stage 4 If you want data sorted in a particular order, you use the Sequence tab to determine which fields are used for sorting, whether the field is sorted in ascending, descending or no order and what action is taken when there is a change in the sort key value. You can specify up to three sort orders.



The On Change field is an option that determines what action, if any, is taken when the value associated with the sort key field changes. You can choose one of the following from a list: **No Action** (meaning that no action is taken when the value of the sort key field changes); **Print Sub Total** (meaning that a subtotal is printed when the value of the sort key field changes. This only applies where totalling is required in the report for columns containing numeric data) or **Print Page Totals** (meaning that a total is required when the value of the sort key field changes and a new page is then started. This only applies where totalling is required in the report for columns containing numeric data).

Stage 5 You may not want all records to be selected when you run a report, and you can control which are included by defining selection criteria. You use the Criteria tab to determine which fields are used for selection purposes and how that selection takes place. Records can be selected on the basis of matching or not matching various criteria. You can define up to five different criteria.



Stage 5 – Further field information

Notes

Relational Operator:An option that determines what kind of selection criterion is applied to the selected field. You can select one of the following from a list:

- Equal To,
- <>Not Equal To,
- Greater Than,
- >= Greater Than Equal To,
- <=Less Than Equal To,
- < Less Than,
- Contains,
- Not Contains,
- Between or
- Not Between.

These work on both alphabetic and numeric characters. For example, to select all accounts whose account codes begin with ‘A’ or ‘B’ you would select the < Less Than option in the Relational Operator box, select Constant in the Comparative Type box and then and enter 'C' in the Comparison box. You could achieve the same result by selecting < Less Than Equal To option in the Relational Operator box, selecting Constant in the Comparative Type box and then and entering 'B' in the Comparison box.

Comparative Type:An option that determines the type of data used to compare against the selection criterion. You can select one of the following from a list:

- Constant (meaning that the comparison is done using a single, fixed value);
- List (meaning that the comparison is done against a list of constants);
- Range (meaning that the comparison is done against a range of constants);
- Field (meaning that the comparison is done against another field included in the report design);
- System Variable (meaning that the comparison is done against a system variable, such as the system date, which can change each time the report is run) or
- Runtime Variable (meaning that the comparison is done against a value entered at the time the report is run).

Note that some comparative types cannot be used with certain relational operators. For example, the Relational Operator 'Between' can only be used with the Comparative type 'Range'.

Comparison: The value used for comparison purposes which depends on the selection you have made in the Comparative Type list.

- Constant: A reference, value or date against which the report field is compared for inclusion in, or exclusion from, the report.

  A value must be ‘hard coded’ into the report design and this will remain unchanged unless the report design itself is edited. For example :

A/c	= Equal To	Constant	A001

Notes

means that everytime the report is run it will only include records where the account number field contains A001.

A '*' wildcard is always assumed on the end of the comparison, therefore :

A/c	= Equal To	Constant	A

will actually include all records where the account number begins with an A.

- List: A list of reference, values or dates against which the report field is compared for inclusion in, or exclusion from, the report. You must separate each entry with a comma.

  This is similar to CONSTANT except more than one value can be compared on a single criteria line. Separate the values with commas. *For example :*

A/c	*= Equal To*	*List*	A001,B001,C001,D001

  means that the report will only include records where the account number field contains A001 *or* B001 *or* C001 *or* D001.

- Range: A range of references, values or dates against which the report field is compared for inclusion in, or exclusion from, the report. You must separate the 'From' and 'To' values with a colon.

  This can only be used with a comparative type of Between or Not Between. The range is specified by using the ':' (colon) symbol. *For example :*

Curr Bal	Between	Range	1000:10000

  means that only records where the value in the current balance field is between 1000 and 10000, inclusive.

- Field: A field selected from a list, against which the report field is compared for inclusion in, or exclusion from, the report. You can only select fields already selected for the report design on the Basis tab.

  This can only be used to compare one ‘Selected Field’ with another ‘Selected Field’. *For example :*

Curr Bal	> Greater Than	Field	Cr Limit

  If you only want to print records where a customers current balance exceeds their credit limit.

- System Variable: A variable selected from a list, against which the report field is compared for inclusion in, or exclusion from, the report. This will depend on the type of field. For example, if the report field is a date, you can select the system date, day, month or year.

- Runtime Variable: A prompt or heading for the type of information to be entered when the report is run against which the report field is compared for inclusion in, or exclusion from, the report. For example, if the report field is a date field, you might type 'Enter Date' in the Comparison box.

Notes

This is used to allow the operator to enter or select a comparative string at the time of running the report. At the design stage an on screen prompt is entered, that will appear when the report is run. *For example :*

A/c | >= Greater Than Equal To | Run Time Variable | First A/c
And

A/c | <= Less Than Equal To | Run Time Variable | Last A/c

will get the operator to enter a start and end range of accounts for inclusion on the report.

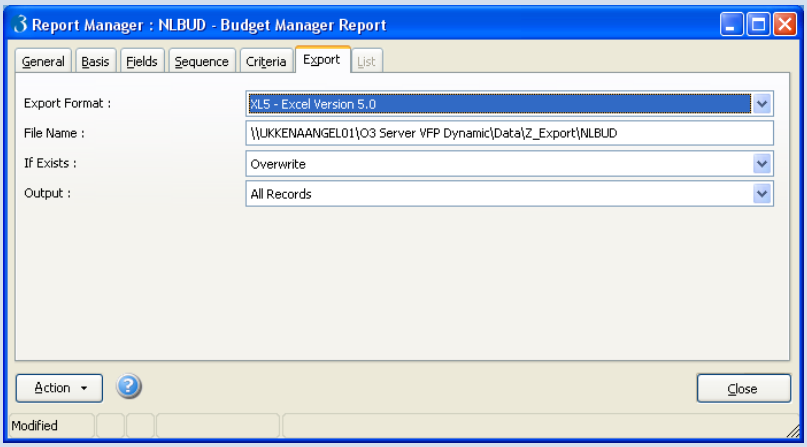
Suppress Suggestions: An option that suppresses the display of a suggested range of values whenever the 'Run Time Variable' option is selected for the Comparative Type box. For example, if you want to enter a range of dates at runtime, but already know what date range you want to use, you might prefer to suppress the display of suggested 'from' and 'to' dates. This means report processing will be quicker because the application does not have to build and extract the possible range options from the database.

Logical Operator: An option that determines how the selection criteria are linked. This does not apply if only one criterion is defined. You can select one of the following from a list:

- End (used to signify the last criterion);
- And (used to signify that *both* the current *and* next criterion must be satisfied before a record is selected for inclusion in or exclusion from the report); or
- Or (used to signify that *either* the current *or* next criterion must be satisfied before a record is selected for inclusion in or exclusion from the report).

Stage 6

If you want to create a file containing the data you have extracted for a report, you use the Export tab to define the format of that file and where it will be stored. If you only want to print or preview reports, you can ignore the Export tab altogether.



Valid formats include; **CSV** (Comma Separated Variable); **DBF** (FoxBase+); **DBF** (FoxPro 2.x+); **DIF** (VisiCalc Data Interchange Format); **MOD** (Multiplan Ver 4.01), **SDF** (System Data Format), **SLK** (Multiplan Symbolic Link), **WK1** (Lotus 123 Version 2.x), **WKS** (Lotus 123 Version 1a), **WR1** (Lotus Symphony Version 1.2), **WRK** (Lotus Symphony Version 1.0), **XLS** (Excel Version 2.0), **XL5** (Excel Version 5.0) or **TXT** (Text Memo).

Notes

Be aware for the If Exist option as neither merge or append are valid choices for DBF-FoxBase + or DBF-FoxPro Version 2.x+. Merge is also not valid for TXT- Text Memo, but append is acceptable.

The Filename : ***Important:*** Prior to Opera 3 (5.14), the suggested location for exported files was `c:\Program Files\Pegasus\Client VFP`. If you have upgraded your computer from Windows XP or another operating system to ***Windows Vista*** and have older reports saved in the `c:\Program Files\Pegasus\Client VFP` folder, you need to change the path to another location in the ***File Name*** field before you export the report again.

Stage 7	The last stage of the report design process involves saving your report design. You can do this after you have completed stages 1 and 2 as the rest are optional. Once stored, you can use the Preview command button to see what the report will look like. You can also run the report using the Reports command in the Reporter folder, or you can use the design as the basis for another report.

A Note About Report Headings

The title you give your report always appears as a report heading on a printed report. Printed reports automatically include the following information in addition:

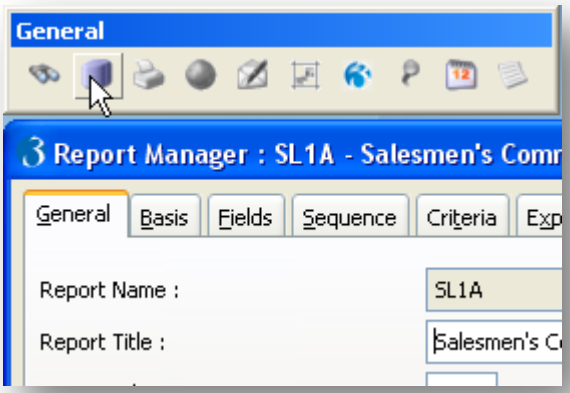
- Company name
- Date and time the report is printed
- Selection criteria applied when the report is printed
- Page numbers

2.1 Copying a Report Template

Notes

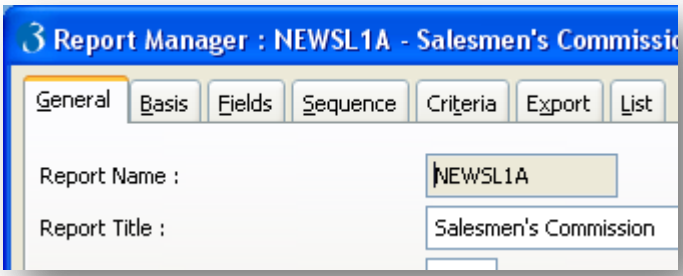
To Copy a Report Template

1. Open the Reporter folder, and then click Report Manager.
2. Retrieve the report you want to use as the basis of a new report.



3. Click the Copy data button on the toolbar.
4. In New Report Name, type the name you want to give to the new report.

5. Click OK.
6. Make the changes you want to the template.
7. To store the report design, on the Record menu, click Save.



2.2 Report Expressions

Notes

Report expressions are provided for formulae and enables you to specify the nature of the calculation to be carried out. During the report print, the system will calculate values based on the formula you define and print the result. The expression builder is part of the Visual FoxPro database, and, in its full functionality is outside the scope of this documentation. If you want to use the expression builder extensively to create your reports, please refer to your Reseller or to the Visual FoxPro documentation.

Warning: Take great care when using the expression builder's facilities as they can be used on any field in a report; not just for formulae and constructs.

You can build simple expressions for formula fields using the information provided below:

Functions: Using the Math list, you can express the relationship between successive elements of the formula. For example, you can use of the following simple arithmetic functions: + (to add two or more formula elements); - (to subtract one element from another); * (to multiply one element by another) or / (to divide one element by another).

Fields and Constants: Formulae can include fields and/or constants. A list of fields is provided from which you can select the one you want, and those available will depend on the selection you have made for the report base on the Basis tab. If you are not sure what fields contain what information, ask your Reseller for assistance. You can include numeric constants ('fixed' values) as part of the formula. These you enter directly in the appropriate part of your formula. If you want to prioritise calculations in a complex formula, you can use parenthesis ().

Picking Fields: You can enter a formula/construct expression of "?-?*?", where each "?" character results in a list of fields being displayed. Each "?" is replaced by the expression for the field that was chosen. Once all fields have been chosen from the list, a list of suggested expressions is displayed. The suggested expression list has rules of precedence that depend on the structure of the expression and the fields chosen.

Number	User Expression Type	Type and order of precedence for list
1	No fields chosen are formatted values.	1
2	All fields chosen are formatted values where the baseless expressions are all identical or are not formatted values AND the first field forms the first part of the expression (ie, no alphabetic characters appear before the first "?"). For example, in the expressions "?+?" and "((?+?)-?)", the first field forms the first part of the expression. In the expression "RGEXPQUAN(?, cfact.cf_factor, cfact.cf_dps)", the first field does not form the first part of the expression.	2, 3, 1, 4
3	All fields chosen are formatted values where the baseless expressions are all identical or are not formatted values AND the first field does not form the first part of the expression (see above definition).	1, 3, 2, 4

4	Fields chosen contain formatted values of mixed baseless expressions, where the first field forms the first part of the expression.	4, 3, 1, 2
5	Fields chosen contain formatted values of mixed baseless expressions, where the first field does not form the first part of the expression.	3, 4, 1, 2

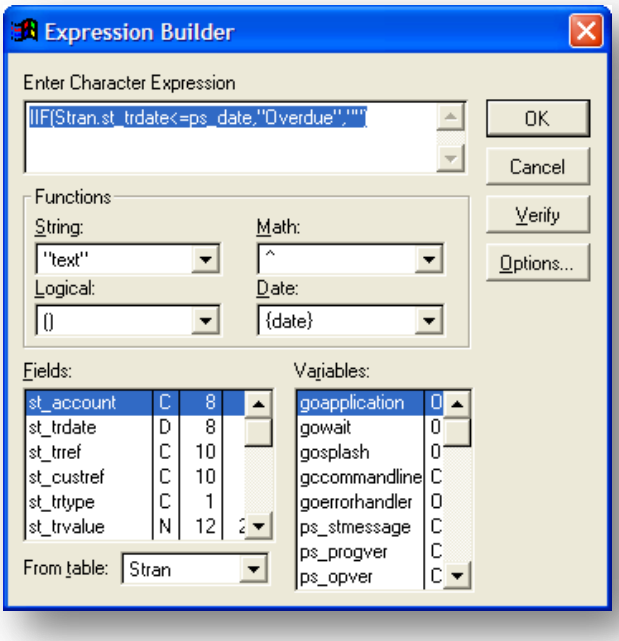
Notes

The suggested expression types can be summarised as follows:

Suggested expression type	Details
1 (Data Reference Only)	"?" characters are replaced by the base values, in the case of formatted values, or field expressions, in the case of non formatted values.
2 (Data Reference Formatted)	The base field within the first formatted value field expression is replaced by the user expression, where "?" characters are replaced by the base fields, in the case of formatted values, or the field expressions, in the case of non formatted values.
3 (Full Expression Only)	"?" characters are replaced by the NUMBERFUNC conversion of the full field expressions, in the case of formatted values, or the field expressions, in the case of non formatted values.
4 (Full Expression Formatted)	The conversion of the first formatted value field is used. The first formatted values is replaced by the base field. All other formatted values are replaced by the conversion of the expression using NUMBERFUNC. Non-formatted value fields are replaced by the field expression.

2.2.1 Expression Builder

The FoxPro expression builder is a means of generating an expression by making selections from list boxes, rather than having to remember or look up variables, functions, tables and field names.



Notes

The expression is built in the large scrollable area at the top of the form, by a combination of typing, and selecting from the 'Functions' lists, table and field lists and variable list.

For example :

To generate the expression displayed in the diagram:

1. Select the IIF(,,) function from the 'Logical' functions list.
2. Select the table 'Stran' from the 'From Table' list.
3. Double-click on the field 'st_dueday' form the 'Fields' list.
4. Type <= at the keyboard.
5. Double-click on the variable ps_date from the 'Variables' list.
6. Press the Cursor right (→) key to position the cursor after the first comma.
7. Type "Overdue" (in quotes).
8. Press the Cursor right (→) key to position the cursor after the second comma.
9. Type "" (empty quotes).
10. Click on the 'Verify' button. If no error message appears the syntax of the expression is correct and a result can be calculated. (If there is a problem with the expression an error message will be displayed.)

Functions

The four dropdown lists labelled 'Functions' are built in Visual FoxPro functions which can not be altered in any way. A detailed explanation of these functions can be found in the Visual FoxPro Language Reference Guide. The most commonly functions used in Pegasus Opera 3, however, have been listed here with a brief explanation of their use.

String

ALLTRIM(expC)	*expC*, a character string (not numeric or date type data) will have any leading or trailing spaces removed. ALLTRIM(" Bill Bloggs ")="Bill Bloggs"
LEFT(expC,x)	Returns the leftmost *'x'* number of characters from the string *expC*. LEFT("Qwerty",3)="Qwe"
LEN(expC)	Returns the number of characters in string *expC*. LEN("Qwerty")=6
RIGHT(expC,x)	Returns the rightmost *'x'* number of characters from the string *expC*. RIGHT("Qwerty",3)="rty"
STR(expN,x,y)	The number *expN* will be changed into a string of characters. I.e. the number 15 will become the characters "15". *'x'* is the length of the resulting string, and *'y'* is the number of decimal places the number has. The purpose of this is to allow manipulation of the number which would otherwise not be possible. STR(39.50,8,2)=" 39.50"
SUBSTR(expC,x,y)	Returns *'y'* characters from string *expC*, starting at character *'x'*. SUBSTR("Qwerty",2,3)="wer"

Notes

Math

ABS(expN)	Returns the absolute value of *expN*, disregarding sign. The result is always a positive number. ABS(521.63)=521.63 ABS(-521.63)=521.63
INT(expN)	Returns the integer part of *expN*. Decimal places are removed. No rounding up occurs. INT(521.63)=521
ROUND(expN,x)	Returns the value of *expN*, rounded to *'x'* decimal places. (0.5 and above rounds up, below 0.5 rounds down.) ROUND(12.3477,2)=12.35
VAL(expC)	Returns a numeric result, from a string of characters. VAL("1213.55")=1213.55

Logical

EMPTY(exp)	Returns .T. (True) or .F. (False) depending upon whether *exp* is empty (contains no characters).
IIF(expL,exp1,exp2)	Returns *exp1* if *expL* is true (.T.), and *exp2* if *expL* is false (.F.). IIF(EMPTY(fld1),"Empty","Contains " + fld1) If *fld1* contains nothing, result="*Empty*" If *fld1* contains "ABC", result="*Contains ABC*"
SEEK(exp1,exp2,exp3)	Looks for, and moves the record pointer to, the first matching record in an open data table, based on criteria matching an indexed field of that table. *exp1* - criteria which must match the index. *exp2* - name of the table to search. It must be open. *exp3* - index number of the field to be used in the search. If omitted, the primary index is assumed. SEEK("ADA0001","Sname")=.T. (and moves the record pointer to that record.)

Date

CDOW(expD)	Returns the name of the day, on the date *expD*. CDOW({30/06/00})="Friday"
CMONTH(expD)	Returns the name of the month, on the date *expD*. CMONTH({30/06/00})="June"
DATE()	Returns the system date (set by the hardware clock).
DAY(expD)	Returns the day number of the date *expD.* DAY({30/06/00})=30
YEAR(expD)	Returns the year number of the date *expD.* YEAR({30/06/00})=2000

Notes

Variables

Variables can be included in expressions by selection from the list displayed in the Visual FoxPro Expression Builder. Variables tend to be pieces of information generated at run-time and stored in memory for a limited time, during processing. For example, when printing a report, the current page number is stored in a variable. Here is a list of those that are likely to be most commonly required when designing a report.

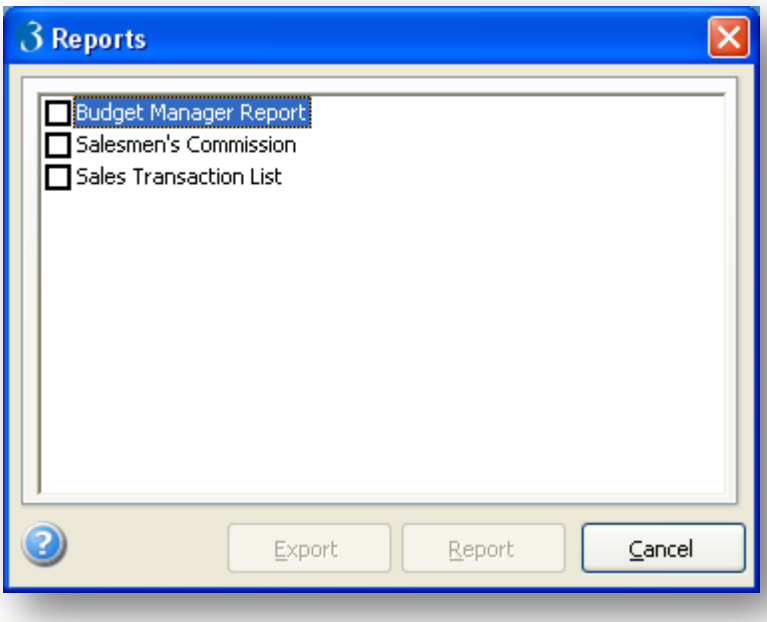
ps_date	Current Opera 3 processing date.
ps_coid	Company ID of company logged into.
ps_coname	Company name of company logged into.
ps_currncy	Home currency of company logged into.
ps_tax	Tax description of company logged into.
ps_user	Login user name.
construct1, construct2 etc	These are the results of the customised 'construct'fields defined in the report.
formula1, formula2 etc	These are the results of the customised 'formula' fields defined in the report.
_pageno	Current page number of the report.

Notes

Accreditation Training Courseware

3 Running a Report

When you save a report design, the report is available to run using the Reports command. The Report Selection form displays a list of the report titles with a check box alongside each one. You click the check box to select a report you want to run, and you can choose any number of reports to run in one process. When you have selected the report(s) you want to run, you use the Export or Report command buttons according to whether you want to print the report content or export it to a file. The format of an exported report depends on the settings on the Export tab of the Report Manager form. If a selection criterion exists for a report that uses a runtime variable, a form will appear when you run that report for input of the variables. For example, you may have defined runtime criteria that prompt for the input of a range of dates. Note that the default value for any runtime boxes is the 'minimum' or 'maximum' value on file, depending on the relational operator used. Selection criteria are determined by the settings on the Criteria tab of the Report Manager form. The output order for records included in the report is determined by the settings on the Sequence tab of the Report Manager form.



Tick any or all of the displayed reports to print them consecutively, to the chosen output device.

Although through the report design you have been able to enter the title, select the columns of information which will appear on the report, enter the selection criteria and the output sequence of the report, the basic layout is predetermined.

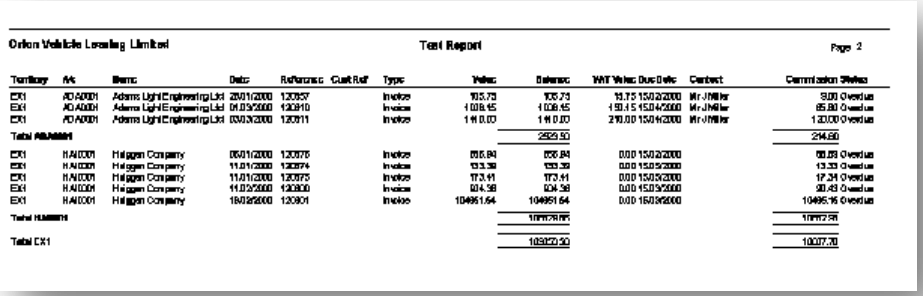
Running a Report

To Run a Report

Notes

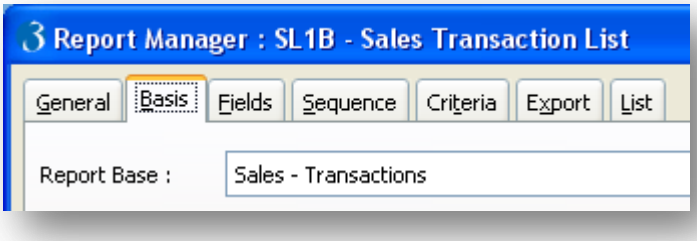
1. Open the Reporter folder, and then click Reports.
2. Click to select the check boxes alongside the report(s) you want to run.
3. Do one of the following:
   a. To export the report(s), click Export. The settings used to determine the format of the exported report depend on the Export tab entries on the Report Manager form.
   b. To print the report or display it on screen, click Report.
4. If any runtime criteria have been defined for the report, complete any boxes presented.

Sample Report:

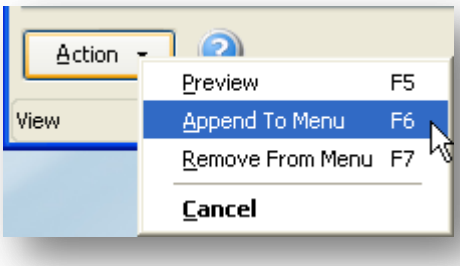


3.1 Appending a Report to the Menu

An Example of an existing report is illustrated below. The Salesmen's Commission Report (based on the Sales Ledger tables).

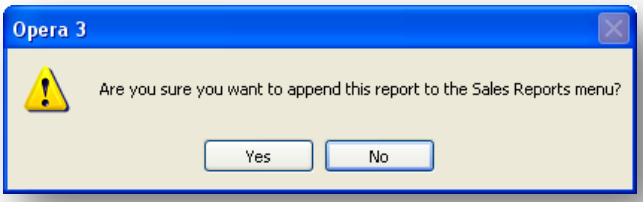


From the Action menu you have the ability to "Append to Menu".

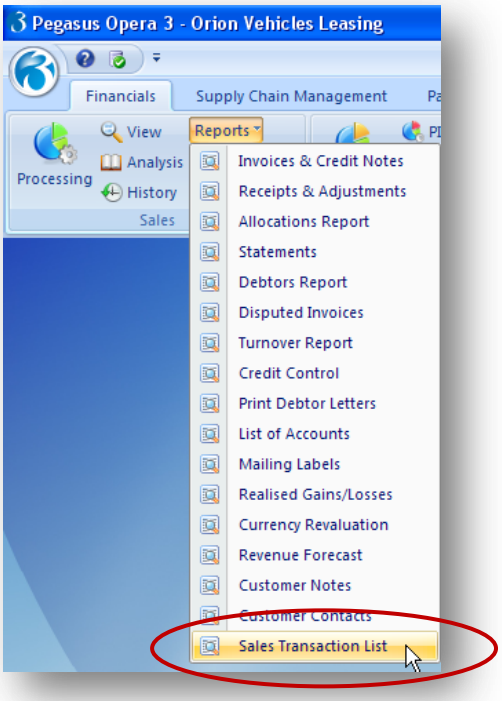


Notes

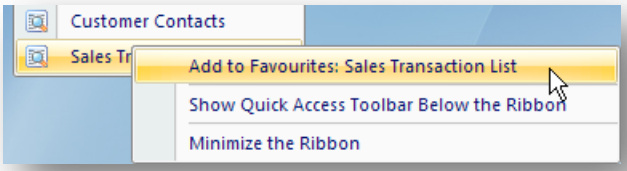
The system is clever enough to know what module the report is based on. It will then prompt the user to confirm appending the report to the module's report menu structure.



Once complete, this can now be run from the Reports menu from the respective module after you re-logon.



This report could now also be added to your favourites bar if required.



Accreditation Training Courseware

Appendix

Appendix A – Pegasus Pre-Defined Expressions

When printing reports it is often desirable to display full descriptions rather than the short descriptions for types of data that appear in Opera 3

For Example

Status recorded in CTRAN.ct_type = A, I, J, R,T,V,X

Rather than display the letter users may wish to display the type as a full description i.e.

Journal

Return to Vendor

Transfer

Opera 3 has some functions that convert the data to “real names”. These functions are available for use only within the Reporter module and the Visual FoxPro designer when used within Opera 3.

EC. VAT	
VCNTRY	
VT_HOME	iif(vcntry.vt_home,"y","")
VT_MEMBER	iif(vcntry.vt_member,"y","")
VT_VATREGD	iif(vcntry.vt_vatregd,"y","")
VT_AFTER	iif(vcntry.vt_after,"y","")
VT_BASECUR	iif(rgseek(vcntry.vt_basecur,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,vcntry.vt_basecur)
VT_SLCURR	iif(rgseek(vcntry.vt_slcurr,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,vcntry.vt_slcurr)
VT_PLCURR	iif(rgseek(vcntry.vt_plcurr,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,vcntry.vt_plcurr)
CASHBOOK	
ABBAL	
AB_STATBAL	sqfcval(abbal.ab_statbal,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_CLRBAL	sqfcval(abbal.ab_clrbal,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_1DCBAL	sqfcval(abbal.ab_1dcbal,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_2DCBAL	sqfcval(abbal.ab_2dcbal,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_CREVAL	sqfcval(abbal.ab_creval,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_DEBVAL	sqfcval(abbal.ab_debval,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_CCVAL	sqfcval(abbal.ab_ccval,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_UCVAL	sqfcval(abbal.ab_ucval,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_CURBAL	sqfcval(abbal.ab_curbal,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AB_RECBAL	sqfcval(abbal.ab_recbal,iif(rgseek(abbal.ab_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
ABPRMS	
AM_PTR	iif(abprms.am_ptr,"y","")
AM_RETSTRU	iif(abprms.am_retstru,"y","")
ABSOFT	
AW_DOS	iif(absoft.aw_dos,"y","")
AW_WIN	iif(absoft.aw_win,"y","")

ADPOUT	
AU_HOME	sqfcval(adpout.au_home,2,.t.)
AU_FCVAL	sqfcval(adpout.au_fcval , iif(empty(adpout.au_fcurr) ,2,adpout.au_fcdec) ,.t.)
AU_FCMULT	iif(adpout.au_fcmult,"y","")
AENTRY	
AE_CBTYPE	iif(rgseek(aentry.ae_cbtype,"atype"),atype.ay_cbtype+" "+atype.ay_desc, aentry.ae_cbtype)
AE_VALUE	sqfcval(aentry.ae_value,iif(empty(nbank.nk_fcurr),2,nbank.nk_fcdec),.t.)
AE_RECBAL	sqfcval(aentry.ae_recbal,iif(empty(nbank.nk_fcurr),2,nbank.nk_fcdec),.t.)
AE_REMOVE	iif(aentry.ae_remove,"y","")
AE_COMPLET	iif(aentry.ae_complet,"y","")
AHPOUT	
AO_FCVAL	sqfcval(ahpout.ao_fcval,iif(empty(ahpout.ao_fcurr),2,ahpout.ao_fcdec),.t.)
AO_HOME	sqfcval(ahpout.ao_home,2,.t.)
AO_TOTDEB	sqfcval(ahpout.ao_totdeb,2,.t.)
AO_LOCK	iif(ahpout.ao_lock,"y","")
ANOML	
AX_SOURCE	cbg_expsrc_c(anoml.ax_source)
AX_FVALUE	sqfcval(anoml.ax_fvalue,anoml.ax_fcdec,.t.)
AX_FCMULT	iif(anoml.ax_fcmult,"y","")
APARM	
AP_PRNPOST	iif(aparm.ap_prnpost,"y","")
ARHEAD	
AE_TYPE	cbg_exptype_c(arhead.ae_type)
AE_FREQ	cbg_expfreq_c(arhead.ae_freq)
AE_VATANAL	iif(arhead.ae_vatanal,"y","")
ARLINE	
AT_CBTYPE	iif(rgseek(arline.at_cbtype,"atype"),atype.ay_cbtype+" "+atype.ay_desc,arline.at_cbtype)
AT_VALUE	sqfcval(arline.at_value,arline.at_fcdec,.t.)
AT_DISC	sqfcval(arline.at_disc,arline.at_fcdec,.t.)
AT_VATVAL	sqfcval(arline.at_vatval,arline.at_fcdec,.t.)
AT_REMIT	iif(arline.at_remit,"y","")
AT_CHQPRN	iif(arline.at_chqprn,"y","")
AT_BACPRN	iif(arline.at_bacprn,"y","")
AT_ECB	iif(arline.at_ecb,"y","")
ASTAT	
AS_VALUE	sqfcval(astat.as_value,iif(rgseek(astat.as_fcurr,"zxchg"),zxchg.xc_dec,2),.t.)
AS_COMPLET	iif(astat.as_complet,"y","")
ATRAN	
AT_CBTYPE	iif(rgseek(atran.at_cbtype,"atype"),atype.ay_cbtype+" "+atype.ay_desc,atran.at_cbtype)
AT_TYPE	cbg_exptype_c(atran.at_type)
AT_TPERIOD	iif(atran.at_tperiod,"y","")
AT_VALUE	sqfcval(atran.at_value,atran.at_fcdec,.t.)
AT_DISC	sqfcval(atran.at_disc,atran.at_fcdec,.t.)
AT_FCMULT	iif(atran.at_fcmult,"y","")
AT_REMOVE	iif(atran.at_remove,"y","")
AT_CHQPRN	iif(atran.at_chqprn,"y","")
AT_CHQLST	iif(atran.at_chqlst,"y","")
AT_BACPRN	iif(atran.at_bacprn,"y","")
AT_CCDPRN	iif(atran.at_ccdprn,"y","")
AT_PYSPRN	iif(atran.at_pysprn,"y","")
AT_CASH	iif(atran.at_cash,"y","")
AT_REMIT	iif(atran.at_remit,"y","")
AT_ECB	iif(atran.at_ecb,"y","")

ATYPE	
AY_TYPE	iif(atype.ay_type="p","payment",iif(atype.ay_type="r","receipt","transfer"))
AY_BANKLST	iif(atype.ay_banklst,"y","")
AY_CRDTCRD	iif(atype.ay_crdtcrd,"y","")
AY_BATCHED	iif(atype.ay_batched,"y","")
CBAUD	
CE_REPONLY	iif(cbaud.ce_reponly,"y","")
CE_FCVAL1	sqfcval(cbaud.ce_fcval1,cbaud.ce_fcdec,.t.)
CE_FCVAL2	sqfcval(cbaud.ce_fcval2,cbaud.ce_fcdec,.t.)
CE_EURVAL1	val(sqfcval(cbaud.ce_eurval1,2,.t.))
CE_EURVAL2	val(sqfcval(cbaud.ce_eurval2,2,.t.))
SALES	
HSANAL	
SA_TRTYPE	expandtyp(hsanal.sa_trtype)
SA_REGION	iif(rgseek("g"+hsanal.sa_region,"slook"),slook.lk_code+" "+slook.lk_desc,hsanal.sa_region)
SA_TERR	iif(rgseek("t"+hsanal.sa_terr,"slook"),slook.lk_code+" "+slook.lk_desc,hsanal.sa_terr)
SA_CUSTYPE	iif(rgseek("y"+hsanal.sa_custype,"slook"),slook.lk_code+" "+slook.lk_desc,hsanal.sa_custype)
SA_ADVANCE	iif(hsanal.sa_advance="y","advanced","")
SALLOC	
AL_TYPE	expandtyp(salloc.al_type)
AL_PAYIND	expandind(salloc.al_payind)
AL_ADVIND	iif(salloc.al_advind,"y","")
AL_PREPRD	iif(salloc.al_preprd,"y","")
SANAL	
SA_TRTYPE	expandtyp(sanal.sa_trtype)
SA_REGION	iif(rgseek("g"+sanal.sa_region,"slook"),slook.lk_code+" "+slook.lk_desc,sanal.sa_region)
SA_TERR	iif(rgseek("t"+sanal.sa_terr,"slook"),slook.lk_code+" "+slook.lk_desc,sanal.sa_terr)
SA_CUSTYPE	iif(rgseek("y"+sanal.sa_custype,"slook"),slook.lk_code+" "+slook.lk_desc,sanal.sa_custype)
SA_ADVANCE	iif(sanal.sa_advance="y","advanced","")
SA_INTERBR	iif(sanal.sa_interbr,"y","")
SA_NRTHIRE	iif(sanal.sa_nrthire,"y","")
SA_BOX6	iif(sanal.sa_box6,"y","")
SA_BOX8	iif(sanal.sa_box8,"y","")
SA_BOX1	iif(sanal.sa_box1,"y","")
SA_ESLSUPP	iif(sanal.sa_eslsupp,"y","")
SA_ESLPROC	iif(sanal.sa_eslproc,"y","")
SA_SSDPRE	iif(sanal.sa_ssdpre,"y","")
SA_SSDPOST	iif(sanal.sa_ssdpost,"y","")
SA_SSDSUPP	iif(sanal.sa_ssdsupp,"y","")
SA_DELTERM	iif(rgseek("l"+sanal.sa_delterm,"vlook"),vlook.lk_code+" "+vlook.lk_desc,sanal.sa_delterm)
SA_TRANSAC	iif(rgseek("n"+sanal.sa_transac,"vlook"),vlook.lk_code+" "+vlook.lk_desc,sanal.sa_transac)
SA_TRANSPT	iif(rgseek("m"+sanal.sa_transpt,"vlook"),vlook.lk_code+" "+vlook.lk_desc,sanal.sa_transpt)
SA_SENTVAT	iif(sanal.sa_sentvat,"y","")
SA_ESLTRIG	iif(sanal.sa_esltrig,"y","")
SECAUD	
SE_REPONLY	iif(secaud.se_reponly,"y","")
SNAME	
SN_REGION	iif(rgseek("g"+sname.sn_region,"slook"),slook.lk_code+" "+slook.lk_desc,sname.sn_region)
SN_TERRTRY	iif(rgseek("t"+sname.sn_terrtry,"slook"),slook.lk_code+"

	"+slook.lk_desc,sname.sn_terrtry)
SN_CUSTYPE	iif(rgseek("y"+sname.sn_custype,"slook"),slook.lk_code+" "+slook.lk_desc,sname.sn_custype)
SN_STOP	iif(sname.sn_stop,"y","")
SN_ACKNOW	iif(sname.sn_acknow,"y","")
SN_DESP	iif(rgseek("d"+sname.sn_desp,"slook"),slook.lk_code+" "+slook.lk_desc,sname.sn_desp)
SN_ROUTE	iif(rgseek("r"+sname.sn_route,"slook"),slook.lk_code+" "+slook.lk_desc,sname.sn_route)
SN_NRTHIRE	iif(sname.sn_nrthire,"y","")
SN_DELT	iif(rgseek("l"+sname.sn_delt,"vlook"),vlook.lk_code+" "+vlook.lk_desc,sname.sn_delt)
SN_NTRN	iif(rgseek("n"+sname.sn_ntrn,"vlook"),vlook.lk_code+" "+vlook.lk_desc,sname.sn_ntrn)
SN_MTRN	iif(rgseek("m"+sname.sn_mtrn,"vlook"),vlook.lk_code+" "+vlook.lk_desc,sname.sn_mtrn)
SN_BRANCH	iif(sname.sn_branch,"y","")
SN_MODEL	iif(sname.sn_model,"y","")
SN_EMAILST	iif(sname.sn_emailst,"y","")
SN_EMAILOA	iif(sname.sn_emailoa,"y","")
SNOML	
SX_TYPE	expandtyp(snoml.sx_type)
SX_FVALUE	sqfcval(snoml.sx_fvalue,snoml.sx_fcdec,.t.)
SX_FCMULT	iif(snoml.sx_fcmult,"y","")
SPARM	
SP_FCSTAT	iif(sparm.sp_fcstat,"y","")
SP_CASHINV	iif(sparm.sp_cashinv,"y","")
SP_EXTEN	iif(sparm.sp_exten,"y","")
SP_STATZRO	iif(sparm.sp_statzro,"y","")
SP_OVRBANK	iif(sparm.sp_ovrbank,"y","")
SP_CONNECT	iif(sparm.sp_connect,"y","")
SP_RCASH01	iif(sparm.sp_rcash01,"y","")
SP_RCASH02	iif(sparm.sp_rcash02,"y","")
SP_RCASH03	iif(sparm.sp_rcash03,"y","")
SP_RCASH04	iif(sparm.sp_rcash04,"y","")
SP_RCASH05	iif(sparm.sp_rcash05,"y","")
SP_RCASH06	iif(sparm.sp_rcash06,"y","")
SP_RCASH07	iif(sparm.sp_rcash07,"y","")
SP_RCASH08	iif(sparm.sp_rcash08,"y","")
SP_RCASH09	iif(sparm.sp_rcash09,"y","")
SP_RCASH10	iif(sparm.sp_rcash10,"y","")
SP_RCASH11	iif(sparm.sp_rcash11,"y","")
SP_RCASH12	iif(sparm.sp_rcash12,"y","")
SP_FCASH01	iif(sparm.sp_fcash01,"y","")
SP_FCASH02	iif(sparm.sp_fcash02,"y","")
SP_FCASH03	iif(sparm.sp_fcash03,"y","")
SP_FCASH04	iif(sparm.sp_fcash04,"y","")
SP_FCASH05	iif(sparm.sp_fcash05,"y","")
SP_FCASH06	iif(sparm.sp_fcash06,"y","")
SP_FCASH07	iif(sparm.sp_fcash07,"y","")
SP_FCASH08	iif(sparm.sp_fcash08,"y","")
SP_FCASH09	iif(sparm.sp_fcash09,"y","")
SP_FCASH10	iif(sparm.sp_fcash10,"y","")
SP_FCASH11	iif(sparm.sp_fcash11,"y","")
SP_FCASH12	iif(sparm.sp_fcash12,"y","")
SP_RFCRD01	iif(sparm.sp_rfcrd01,"y","")
SP_RFCRD02	iif(sparm.sp_rfcrd02,"y","")

SP_RFCRD03	iif(sparm.sp_rfcrd03,"y","")
SP_RFCRD04	iif(sparm.sp_rfcrd04,"y","")
SP_RFCRD05	iif(sparm.sp_rfcrd05,"y","")
SP_RFCRD06	iif(sparm.sp_rfcrd06,"y","")
SP_RFCRD07	iif(sparm.sp_rfcrd07,"y","")
SP_RFCRD08	iif(sparm.sp_rfcrd08,"y","")
SP_RFCRD09	iif(sparm.sp_rfcrd09,"y","")
SP_RFCRD10	iif(sparm.sp_rfcrd10,"y","")
SP_RFCRD11	iif(sparm.sp_rfcrd11,"y","")
SP_RFCRD12	iif(sparm.sp_rfcrd12,"y","")
SP_ASKLETR	iif(sparm.sp_askletr,"y","")
SPRFLS	
SC_OPEN	iif(sprfls.sc_open,"y","")
SC_STATMNT	iif(sprfls.sc_statmnt,"y","")
SC_CURRNCY	iif(rgseek(sprfls.sc_currncy,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,sprfls.sc_currncy)
SC_OVRBANK	iif(sprfls.sc_ovrbank,"y","")
SC_DEFREC	iif(empty(sprfls.sc_defrec),"",eval("sparm.sp_rec"+right(str(100+sprfls.sc_defrec),2)))
SC_DEFRFD	iif(empty(sprfls.sc_defrfd),"",eval("sparm.sp_rfd"+right(str(100+sprfls.sc_defrfd),2)))
SC_PAYNAME	iif(sprfls.sc_payname,"y","")
STERMS	
SR_DEBTLTR	iif(sterms.sr_debtltr,"y","")
SR_SDSCDEL	iif(sterms.sr_sdscdel,"y","")
SR_VATINCL	iif(sterms.sr_vatincl,"y","")
STRAN	
ST_TRTYPE	expandtyp(stran.st_trtype)
ST_PAID	expandind(stran.st_paid)
ST_ADVANCE	iif(stran.st_advance="y","advanced","")
ST_FCVAL	sqfcval(stran.st_fcval,stran.st_fcdec,.f.)
ST_FCBAL	sqfcval(stran.st_fcbal,stran.st_fcdec,.f.)
ST_FCMULT	iif(stran.st_fcmult,"y","")
ST_DISPUTE	iif(stran.st_dispute,"y","")
ST_EDI	iif(stran.st_edi,"y","")
ST_BINREP	iif(stran.st_binrep,"y","")
INVOICING/SOP	
IALLOC	
IT_QUAN	rgexpquan(ialloc.it_quan , cfact.cf_factor,cfact.cf_dps)
IT_QTYALLC	rgexpquan(ialloc.it_qtyallc , cfact.cf_factor,cfact.cf_dps)
IT_QTYPICK	rgexpquan(ialloc.it_qtypick , cfact.cf_factor,cfact.cf_dps)
IT_QTYORIG	rgexpquan(ialloc.it_qtyorig ,cfact.cf_factor,cfact.cf_dps)
IT_QTYEDIT	rgexpquan(ialloc.it_qtyedit , cfact.cf_factor,cfact.cf_dps)
IT_COST	sqfcval(ialloc.it_cost , cfact.cf_selldps, .t.)
IT_ROUTE	iif(rgseek("r"+ialloc.it_route,"slook"),slook.lk_code+" "+slook.lk_desc,ialloc.it_route)
IT_PORDER	iif(ialloc.it_porder,"y","")
IT_WORDER	iif(ialloc.it_worder,"y","")
IHEAD	
IH_DESP	iif(rgseek("d"+ihead.ih_desp,"slook"),slook.lk_code+" "+slook.lk_desc,ihead.ih_desp)
IH_LOC	iif(rgseek(ihead.ih_loc,"cware"),cware.cw_code+" "+cware.cw_desc,ihead.ih_loc)
IH_REGION	iif(rgseek("g"+ihead.ih_region,"slook"),slook.lk_code+" "+slook.lk_desc,ihead.ih_region)
IH_TERRTRY	iif(rgseek("t"+ihead.ih_terrtry,"slook"),slook.lk_code+" "+slook.lk_desc,ihead.ih_terrtry)
IH_ROUTE	iif(rgseek("r"+ihead.ih_route,"slook"),slook.lk_code+" "+slook.lk_desc,ihead.ih_route)

IH_FCURR	iif(rgseek(ihead.ih_fcurr,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,ihead.ih_fcurr)
IH_FCVAL	sqfcval(val(ihead.ih_fcval),iif(!empty(ihead.ih_fcurr),ihead.ih_fcdec,2),.t.)
IH_FCVAT	sqfcval(val(ihead.ih_fcvat),iif(!empty(ihead.ih_fcurr),ihead.ih_fcdec,2),.t.)
IH_DELT	iif(rgseek("l"+ihead.ih_delt,"vlook"),vlook.lk_code+" "+vlook.lk_desc,ihead.ih_delt)
IH_NTRN	iif(rgseek("n"+ihead.ih_ntrn,"vlook"),vlook.lk_code+" "+vlook.lk_desc,ihead.ih_ntrn)
IH_MTRN	iif(rgseek("m"+ihead.ih_mtrn,"vlook"),vlook.lk_code+" "+vlook.lk_desc,ihead.ih_mtrn)
IH_FCMULT	iif(ihead.ih_fcmult,"y","")
IH_EDI	iif(ihead.ih_edi,"y","")
IH_EBRIDGE	iif(ihead.ih_ebridge,"y","")
ITRAN	
IT_QUAN	rgexpquan(itran.it_quan , cfact.cf_factor,cfact.cf_dps)
IT_PRICE	iif(ps_fundec_l and !empty(itran.it_fcurr), str(itran.it_price / 10^itran.it_fundec,10,itran.it_fundec),sqfcval(itran.it_price, cfact.cf_selldps, .t.))
IT_LINEVAL	sqfcval(itran.it_lineval,iif(!empty(itran.it_fcurr),itran.it_fcdec,2),.t.)
IT_SDISC	iif(itran.it_sdisc,"y","")
IT_ODISC	iif(itran.it_odisc,"y","")
IT_EXVAT	sqfcval(itran.it_exvat,iif(!empty(itran.it_fcurr),itran.it_fcdec,2),.t.)
IT_DISCVAL	sqfcval(itran.it_discval,iif(!empty(itran.it_fcurr),itran.it_fcdec,2), .t.)
IT_SETT1	sqfcval(itran.it_sett1,iif(!empty(itran.it_fcurr),itran.it_fcdec,2),.t.)
IT_SETT2	sqfcval(itran.it_sett2,iif(!empty(itran.it_fcurr),itran.it_fcdec,2),.t.)
IT_OVERALL	sqfcval(itran.it_overall,iif(!empty(itran.it_fcurr),itran.it_fcdec,2),.t.)
IT_VATVAL	sqfcval(itran.it_vatval,iif(!empty(itran.it_fcurr),itran.it_fcdec,2),.t.)
IT_COST	sqfcval(itran.it_cost, cfact.cf_selldps, .t.)
IT_QTYALLC	rgexpquan(itran.it_qtyallc , cfact.cf_factor,cfact.cf_dps)
IT_QTYPICK	rgexpquan(itran.it_qtypick , cfact.cf_factor,cfact.cf_dps)
IT_QTYDELV	rgexpquan(itran.it_qtydelv , cfact.cf_factor,cfact.cf_dps)
IT_QTYINV	rgexpquan(itran.it_qtyinv , cfact.cf_factor,cfact.cf_dps)
IT_DELT	iif(rgseek("l"+itran.it_delt,"vlook"),vlook.lk_code+" "+vlook.lk_desc,itran.it_delt)
IT_NTRN	iif(rgseek("n"+itran.it_ntrn,"vlook"),vlook.lk_code+" "+vlook.lk_desc,itran.it_ntrn)
IT_MTRN	iif(rgseek("m"+itran.it_mtrn,"vlook"),vlook.lk_code+" "+vlook.lk_desc,itran.it_mtrn)
IT_FCMULT	iif(itran.it_fcmult,"y","")
IT_FROMJC	iif(itran.it_fromjc,"y","")
IT_MASSUT	iif(itran.it_massut,"y","")
IT_WORDER	iif(itran.it_worder,"y","")
IT_PORDER	iif(itran.it_porder,"y","")
IT_PONOW	iif(itran.it_ponow,"y","")
IT_POCHG	iif(itran.it_pochg,"y","")
IT_WOCHG	iif(itran.it_wochg,"y","")
IT_WONOW	iif(itran.it_wonow,"y","")
PURCHASE	
HPANAL	
PA_TRTYPE	expandtyp(hpanal.pa_trtype)
PA_SUPTYPE	iif(rgseek("s"+hpanal.pa_suptype,"plook"),plook.lk_code+" "+plook.lk_desc,hpanal.pa_suptype)
PA_ADVANCE	iif(hpanal.pa_advance="y","advanced","")
PALLOC	
AL_TYPE	expandtyp(palloc.al_type)
AL_PAYIND	expandind(palloc.al_payind)
AL_ADVIND	iif(palloc.al_advind,"y","")
AL_ADVTRAN	iif(palloc.al_advtran,"y","")
AL_PREPRD	iif(palloc.al_preprd,"y","")
PANAL	
PA_TRTYPE	expandtyp(panal.pa_trtype)
PA_ADVANCE	iif(panal.pa_advance="y","advanced","")

PA_INTERBR	iif(panal.pa_interbr,"y","")
PA_NRTHIRE	iif(panal.pa_nrthire,"y","")
PA_BOX7	iif(panal.pa_box7,"y","")
PA_BOX9	iif(panal.pa_box9,"y","")
PA_BOX2	iif(panal.pa_box2,"y","")
PA_BOX4	iif(panal.pa_box4,"y","")
PA_SSDPRE	iif(panal.pa_ssdpre,"y","")
PA_SSDPOST	iif(panal.pa_ssdpost,"y","")
PA_SSDSUPP	iif(panal.pa_ssdsupp,"y","")
PA_DELTERM	iif(rgseek("l"+panal.pa_delterm,"vlook"),vlook.lk_code+" "+vlook.lk_desc,panal.pa_delterm)
PA_TRANSAC	iif(rgseek("n"+panal.pa_transac,"vlook"),vlook.lk_code+" "+vlook.lk_desc,panal.pa_transac)
PA_TRANSPT	iif(rgseek("m"+panal.pa_transpt,"vlook"),vlook.lk_code+" "+vlook.lk_desc,panal.pa_transpt)
PA_SENTVAT	iif(panal.pa_sentvat,"y","")
PECAUD	
PE_REPONLY	iif(pecaud.pe_reponly,"y","")
PHEAD	
PH_TRTYPE	expandtyp(phead.ph_trtype)
PH_FCM	iif(phead.ph_fcm,"y","")
PH_FCVAL	sqfcval(phead.ph_fcval*iif(phead.ph_trtype="i",1,1),iif(empty(phead.ph_fcurr),2,phead.ph_fcdec),.t.
PH_FCMVAL	sqfcval(phead.ph_fcmval*iif(phead.ph_trtype="i",1,1),iif(empty(phead.ph_fcurr),2,phead.ph_fcdec),.t.)
PH_INTERBR	iif(phead.ph_interbr,"y","")
PH_NRTHIRE	iif(phead.ph_nrthire,"y","")
PH_TRVALUE	phead.ph_trvalue *iif(phead.ph_trtype="i",1,-1)
PH_MVAL	phead.ph_mval *iif(phead.ph_trtype="i",1,-1)
PH_VATVAL	phead.ph_vatval *iif(phead.ph_trtype="i",1,-1)
PH_MVAT	phead.ph_mvat *iif(phead.ph_trtype="i",1,-1)
PH_POSTED	iif(phead.ph_posted,"y","")
PH_AUTHOR	iif(phead.ph_author,"y","")
PH_SETMARK	iif(phead.ph_setmark,"y","")
PLINE	
PL_INVQUA	rgexpquan(pline.pl_invqua , cfact.cf_factor,cfact.cf_dps)
PL_INVPRI	stg_strprice_c(pline.pl_invpri ,pline.pl_cnref,.t.,12)
PL_FCM	iif(pline.pl_fcm,"y","")
PL_FCVAL	sqfcval(pline.pl_fcval*iif(phead.ph_trtype="i",1,-1),iif(empty(pline.pl_fcurr),2,pline.pl_fcdec),.t.)
PL_FCMVAL	sqfcval(pline.pl_fcmval*iif(phead.ph_trtype="i",1,-1),iif(empty(pline.pl_fcurr),2,pline.pl_fcdec),.t.)
PL_BOX7	iif(pline.pl_box7,"y","")
PL_BOX9	iif(pline.pl_box9,"y","")
PL_BOX2	iif(pline.pl_box2,"y","")
PL_BOX4	iif(pline.pl_box4,"y","")
PL_SSDPRE	iif(pline.pl_ssdpre,"y","")
PL_SSDPOST	iif(pline.pl_ssdpost,"y","")
PL_SSDSUPP	iif(pline.pl_ssdsupp,"y","")
PL_DELTERM	iif(rgseek("m"+ pline.pl_delterm ,"vlook"),vlook.lk_code+" "+vlook.lk_desc, pline.pl_delterm)
PL_TRANSAC	iif(rgseek("m"+ pline.pl_transac ,"vlook"),vlook.lk_code+" "+vlook.lk_desc, pline.pl_transac)
PL_TRANSPT	iif(rgseek("m"+ pline.pl_transpt ,"vlook"),vlook.lk_code+" "+vlook.lk_desc, pline.pl_transpt)
PL_SSDVAL	pline.pl_ssdval*iif(phead.ph_trtype="i",1,-1)
PL_SSDFVAL	sqfcval(pline.pl_ssdfval*iif(phead.ph_trtype="i",1,-1),iif(empty(pline.pl_fcurr),2,pline.pl_fcdec),.t.)
PL_AVALUE	pline.pl_avalue*iif(phead.ph_trtype="i",1,-1)

PL_AVAT	pline.pl_avat*iif(phead.ph_trtype="i",1,-1)
PL_MQTY	rgexpquan(pline.pl_mqty , cfact.cf_factor,cfact.cf_dps)
PL_VATVAL	pline.pl_vatval*iif(phead.ph_trtype="i",1,-1)
PL_POSTED	iif(pline.pl_posted,"y","")
PL_FCVAT	sqfcval(pline.pl_fcvat*iif(phead.ph_trtype="i",1,-1),iif(empty(pline.pl_fcurr),2,pline.pl_fcdec),.t.)
PL_MULTNM	iif(pline.pl_multnm,"y","")
PMATCH	
PM_QTY	rgexpquan(pmatch.pm_qty , cfact.cf_factor,cfact.cf_dps)
PM_POSTED	iif(pmatch.pm_posted,"y","")
PNAME	
PN_SUPTYPE	iif(rgseek("s"+pname.pn_suptype,"plook"),plook.lk_code+" "+plook.lk_desc,pname.pn_suptype)
PN_STOP	iif(pname.pn_stop,"y","")
PN_ORDER	iif(pname.pn_order,"y","")
PN_PAYMETH	iif(pname.pn_paymeth = 'c',"cheque","bacs")
PN_NRTHIRE	iif(pname.pn_nrthire,"y","")
PN_DELT	iif(rgseek("l" + pname.pn_delt, "vlook"), vlook.lk_code + " " + vlook.lk_desc, pname.pn_delt)
PN_NTRN	iif(rgseek("n" + pname.pn_ntrn, "vlook"), vlook.lk_code + " " + vlook.lk_desc, pname.pn_ntrn)
PN_MTRN	iif(rgseek("m" + pname.pn_mtrn, "vlook"), vlook.lk_code + " " + vlook.lk_desc, pname.pn_mtrn)
PN_BRANCH	iif(pname.pn_branch,"y","")
PN_MODEL	iif(pname.pn_model,"y","")
PN_NOPRTPO	iif(pname.pn_noprtpo,"y","")
PN_EMAILRA	iif(pname.pn_emailra,"y","")
PNOML	
PX_TYPE	expandtyp(pnoml.px_type)
PX_FVALUE	sqfcval(pnoml.px_fvalue,pnoml.px_fcdec,.t.)
PX_FCMULT	iif(pnoml.px_fcmult,"y","")
PPARM	
PP_SETTOVR	iif(pparm.pp_settovr,"y","")
PP_OVRBANK	iif(pparm.pp_ovrbank,"y","")
PP_PCASH01	iif(pparm.pp_pcash01,"y","")
PP_PCASH02	iif(pparm.pp_pcash02,"y","")
PP_PCASH03	iif(pparm.pp_pcash03,"y","")
PP_PCASH04	iif(pparm.pp_pcash04,"y","")
PP_PCASH05	iif(pparm.pp_pcash05,"y","")
PP_PCASH06	iif(pparm.pp_pcash06,"y","")
PP_PCASH07	iif(pparm.pp_pcash07,"y","")
PP_PCASH08	iif(pparm.pp_pcash08,"y","")
PP_PCASH09	iif(pparm.pp_pcash09,"y","")
PP_PCASH10	iif(pparm.pp_pcash10,"y","")
PP_PCASH11	iif(pparm.pp_pcash11,"y","")
PP_PCASH12	iif(pparm.pp_pcash12,"y","")
PP_FCASH01	iif(pparm.pp_fcash01,"y","")
PP_FCASH02	iif(pparm.pp_fcash02,"y","")
PP_FCASH03	iif(pparm.pp_fcash03,"y","")
PP_FCASH04	iif(pparm.pp_fcash04,"y","")
PP_FCASH05	iif(pparm.pp_fcash05,"y","")
PP_FCASH06	iif(pparm.pp_fcash06,"y","")
PP_FCASH07	iif(pparm.pp_fcash07,"y","")
PP_FCASH08	iif(pparm.pp_fcash08,"y","")
PP_FCASH09	iif(pparm.pp_fcash09,"y","")
PP_FCASH10	iif(pparm.pp_fcash10,"y","")
PP_FCASH11	iif(pparm.pp_fcash11,"y","")

PP_FCASH12	iif(pparm.pp_fcash12,"y","")
PPRFLS	
PC_OPEN	iif(pprfls.pc_open,"y","")
PC_CURRNCY	iif(rgseek(pprfls.pc_currncy,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,pprfls.pc_currncy)
PC_REMIT	iif(pprfls.pc_remit,"y","")
PC_OVRBANK	iif(pprfls.pc_ovrbank,"y","")
PC_DEFPAY	iif(empty(pprfls.pc_defpay),"",eval("pparm.pp_pay"+right(str(100+pprfls.pc_defpay),2)))
PC_DEFRFD	iif(empty(pprfls.pc_defrfd),"",eval("pparm.pp_rfd"+right(str(100+pprfls.pc_defrfd),2)))
PC_PAYNAME	iif(pprfls.pc_payname,"y","")
PTERMS	
PR_SDSCDEL	iif(pterms.pr_sdscdel,"y","")
PR_VATINCL	iif(pterms.pr_vatincl,"y","")
PTRAN	
PT_TRTYPE	expandtyp(ptran.pt_trtype)
PT_PAID	expandind(ptran.pt_paid)
PT_ADVANCE	iif(ptran.pt_advance="y","advanced","")
PT_FCVAL	sqfcval(ptran.pt_fcval,ptran.pt_fcdec,.f.)
PT_FCBAL	sqfcval(ptran.pt_fcbal,ptran.pt_fcdec,.f.)
PT_FCMULT	iif(ptran.pt_fcmult,"y","")
POP	
DMHEAD	
DC_EXMULT	iif(dmhead.dc_exmult,"y","")
DC_TOTVAL	sqfcval(dmhead.dc_totval,dmhead.dc_dplace,.t.)
DC_BINREP	iif(dmhead.dc_binrep,"y","")
DMLINE	
DO_REQQTY	rgexpquan(dmline.do_reqqty,cfact.cf_factor,cfact.cf_dps)
DO_QUOTED	iif(dmline.do_quoted,"y","")
DO_PRICE	iif(ps_fundec_l and !empty(dmline.do_currcy), str(dmline.do_price/10^dmline.do_fundec,12,dmline.do_fundec), stg_strprice_c(dmline.do_price,dmline.do_cnref,.t.,12))
DO_VALUE	sqfcval(dmline.do_value,dmline.do_dplace,.t.)
DO_DISCV	sqfcval(dmline.do_discv,dmline.do_dplace,.t.)
DO_ORDDSO	iif(dmline.do_orddso,"y","")
DNAME	
DN_INACT	iif(dname.dn_inact,"y","")
DODOC	
DC_ONORDER	pog_rqonorder_c(dodoc.dc_onorder)
DC_PRINTED	iif(dodoc.dc_printed,"y","")
DC_EXMULT	iif(dodoc.dc_exmult,"y","")
DC_TOTVAL	sqfcval(dodoc.dc_totval,dodoc.dc_dplace,.t.)
DOHEAD	
DC_CANCEL	iif(dohead.dc_cancel,"y","")
DC_EXMULT	iif(dohead.dc_exmult,"y","")
DC_TOTVAL	sqfcval(dohead.dc_totval,dohead.dc_dplace,.t.)
DC_BINREP	iif(dohead.dc_binrep,"y","")
DC_PRINTED	iif(dohead.dc_printed,"y","")
DOLINE	
DO_REQQTY	rgexpquan(doline.do_reqqty,cfact.cf_factor,cfact.cf_dps)
DO_RECQTY	rgexpquan(doline.do_recqty,cfact.cf_factor,cfact.cf_dps)
DO_INVQTY	rgexpquan(doline.do_invqty,cfact.cf_factor,cfact.cf_dps)
DO_QUOTED	iif(doline.do_quoted,"y","")
DO_PRICE	iif(isnull(dsprod.ds_curr),str(doline.do_price/10^doline.do_dplace,12,doline.do_dplace), rgstrdolineprice(dsprod.ds_curr, doline.do_price, dsprod.ds_fundec, dsprod.ds_cnref, dsprod.ds_fcdec))

DO_VALUE	sqfcval(doline.do_value,doline.do_dplace,.t.)
DO_DISCV	sqfcval(doline.do_discv,doline.do_dplace,.t.)
DO_COMMIT	iif(doline.do_commit,"y","")
DO_ORDDSO	iif(doline.do_orddso,"y","")
DO_CMPLTD	iif(doline.do_cmpltd,"y","")
DO_CMPLTI	iif(doline.do_cmplti,"y","")
DPARM	
DP_CRSPROD	iif(dparm.dp_crsprod,"y","")
DP_OVERRPO	iif(dparm.dp_overrpo,"y","")
DP_NOIMAGE	iif(dparm.dp_noimage,"y","")
DP_PRNSEQ	iif(dparm.dp_prnseq,"y","")
DP_DELDEL	iif(dparm.dp_deldel,"y","")
DRHEAD	
DH_AUTHED	iif(drhead.dh_authed,"y","")
DH_SFIED	iif(drhead.dh_sfied,"y","")
DH_ONORDER	pog_rqonorder_c(drhead.dh_onorder)
DH_PRINTED	iif(drhead.dh_printed,"y","")
DRLINE	
DL_ONORDER	pog_rqonorder_c(drline.dl_onorder)
DL_REQQTY	rgexpquan(drline.dl_reqqty,cfact.cf_factor,cfact.cf_dps)
DL_PRICE	stg_strprice_c(drline.dl_price,drline.dl_cnref,.t.,12)
DL_MINQTY	rgexpquan(drline.dl_minqty,cfact.cf_factor,cfact.cf_dps)
DL_QUOTED	iif(drline.dl_quoted,"y","")
DSPROD	
DS_COST	iif(ps_fundec_l and !empty(dsprod.ds_curr), str(dsprod.ds_cost/10^dsprod.ds_fundec,12,dsprod.ds_fundec), stg_strprice_c(dsprod.ds_cost,dsprod.ds_cnref,.t.,12,.f.,iif(!empty(dsprod.ds_fcdec), dsprod.ds_fcdec,"")))
DS_LINE	iif(dsprod.ds_line,"y","")
DS_OVER	iif(dsprod.ds_over,"y","")
DS_LDQTY	rgexpquan(dsprod.ds_ldqty,cfact.cf_factor,cfact.cf_dps)
DS_LDCOST	stg_strprice_c(dsprod.ds_ldcost,dsprod.ds_cnref,.t.,12)
DS_LIQTY	rgexpquan(dsprod.ds_liqty,cfact.cf_factor,cfact.cf_dps)
DS_LICOST	stg_strprice_c(dsprod.ds_licost,dsprod.ds_cnref,.t.,12)
DS_ECQTY	rgexpquan(dsprod.ds_ecqty,cfact.cf_factor,cfact.cf_dps)
NOMINAL	
NACNT	
NA_REDIST	iif(nacnt.na_redist,"y","")
NA_OPEN	iif(nacnt.na_open,"y","")
NA_POST	iif(nacnt.na_post,"y","")
NA_COMM	iif(nacnt.na_comm,"y","")
SQ_PRIVATE	iif(nacnt.sq_private,"y","")
NA_REPKEY1	iif(rgseek("1"+nacnt.na_repkey1,"nrpkey"),nrpkey.nr_code+" "+nrpkey.nr_desc,padr(nacnt.na_repkey1,25))
NA_REPKEY2	iif(rgseek("2"+nacnt.na_repkey2,"nrpkey"),nrpkey.nr_code+" "+nrpkey.nr_desc,padr(nacnt.na_repkey2,25))
NA_REPKEY3	iif(rgseek("3"+nacnt.na_repkey3,"nrpkey"),nrpkey.nr_code+" "+nrpkey.nr_desc,padr(nacnt.na_repkey3,25))
NA_REPKEY4	iif(rgseek("4"+nacnt.na_repkey4,"nrpkey"),nrpkey.nr_code+" "+nrpkey.nr_desc,padr(nacnt.na_repkey4,25))
NA_REPKEY5	iif(rgseek("5"+nacnt.na_repkey5,"nrpkey"),nrpkey.nr_code+" "+nrpkey.nr_desc,padr(nacnt.na_repkey5,25))
NA_FCMULT	iif(nacnt.na_fcmult,"y","")
NA_FBALP01	sqfcval(nacnt.na_fbalp01,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP02	sqfcval(nacnt.na_fbalp02,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP03	sqfcval(nacnt.na_fbalp03,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP04	sqfcval(nacnt.na_fbalp04,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)

NA_FBALP05	sqfcval(nacnt.na_fbalp05,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP06	sqfcval(nacnt.na_fbalp06,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP07	sqfcval(nacnt.na_fbalp07,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP08	sqfcval(nacnt.na_fbalp08,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP09	sqfcval(nacnt.na_fbalp09,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP10	sqfcval(nacnt.na_fbalp10,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP11	sqfcval(nacnt.na_fbalp11,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP12	sqfcval(nacnt.na_fbalp12,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALP13	sqfcval(nacnt.na_fbalp13,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC01	sqfcval(nacnt.na_fbalc01,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC02	sqfcval(nacnt.na_fbalc02,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC03	sqfcval(nacnt.na_fbalc03,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC04	sqfcval(nacnt.na_fbalc04,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC05	sqfcval(nacnt.na_fbalc05,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC06	sqfcval(nacnt.na_fbalc06,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC07	sqfcval(nacnt.na_fbalc07,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC08	sqfcval(nacnt.na_fbalc08,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC09	sqfcval(nacnt.na_fbalc09,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC10	sqfcval(nacnt.na_fbalc10,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC11	sqfcval(nacnt.na_fbalc11,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC12	sqfcval(nacnt.na_fbalc12,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FBALC13	sqfcval(nacnt.na_fbalc13,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FPTDDR	sqfcval(nacnt.na_fptddr,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FPTDCR	sqfcval(nacnt.na_fptdcr,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FYTDDR	sqfcval(nacnt.na_fytddr,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FYTDCR	sqfcval(nacnt.na_fytdcr,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FPRYDR	sqfcval(nacnt.na_fprydr,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NA_FPRYCR	sqfcval(nacnt.na_fprycr,iif(empty(nacnt.na_fcurr),2,nacnt.na_fcdec),.t.)
NBANK	
NK_PETTY	iif(nbank.nk_petty,"y","")
NK_RECBAL	sqfcval(nbank.nk_recbal,iif(empty(nbank.nk_fcurr),2,nbank.nk_fcdec),.t.)
NK_CURBAL	sqfcval(nbank.nk_curbal,iif(empty(nbank.nk_fcurr),2,nbank.nk_fcdec),.t.)
NK_RECCFWD	sqfcval(nbank.nk_reccfwd,iif(empty(nbank.nk_fcurr),2,nbank.nk_fcdec),.t.)
NK_ECBREC	iif(nbank.nk_ecbrec,"y","")
NK_ECBPAY	iif(nbank.nk_ecbpay,"y","")
NK_ECBDWN	iif(nbank.nk_ecbdwn,"y","")
NBUDG	
NB_METH	iif(nbudg.nb_meth="m","manual",iif(nbudg.nb_meth="f","factored",iif(nbudg.nb_meth="b","budget prior",iif(nbudg.nb_meth="a","actual",iif(nbudg.nb_meth="r","revised prior","unknown")))))
NB_FACT	iif(rgseek(nbudg.nb_fact,"nfact"),nfact.nf_fact+" "+nfact.nf_desc,nbudg.nb_fact)
NDETL	
ND_FVALUE	sqfcval(ndetl.nd_fvalue,ndetl.nd_fcdec,.t.)
ND_FCMULT	iif(ndetl.nd_fcmult,"y","")
ND_TFVALUE	sqfcval(ndetl.nd_tfvalue,ndetl.nd_tfcdec,.t.)
ND_TFCMULT	iif(ndetl.nd_tfcmult,"y","")
NHEAD	
NH_RECUR	iif(nhead.nh_recur,"y","")
NH_RETAIN	iif(nhead.nh_retain,"y","")
NH_FREQ	iif(nhead.nh_nh_freq="w","week",iif(nhead.nh_freq="m","month",iif(nhead.nh_freq="p","period",iif(nhead.nh_freq="d","days","unknown"))))
NH_REV	iif(nhead.nh_rev,"y","")
NH_VATANAL	iif(nhead.nh_vatanal,"y","")
NH_FCMULT	iif(nhead.nh_fcmult,"y","")
NH_FDRTOT	sqfcval(nhead.nh_fdrtot,nhead.nh_fcdec,.t.)

NH_FCRTOT	sqfcval(nhead.nh_fcrtot,nhead.nh_fcdec,.t.)
NJMEMO	
NJ_BINREP	iif(njmemo.nj_binrep,"y","")
NMRHED	
NMH_SUMMRY	iif(nmrhed.nmh_summry,"y","")
NMH_ZERO	iif(nmrhed.nmh_zero,"y","")
NMH_XTRACT	iif(nmrhed.nmh_xtract,"y","")
NMH_MISSNG	iif(nmrhed.nmh_missng,"y","")
NPARM	
NP_IMGSUMM	iif(nparm.np_imgsumm,"y","")
NSUBT	
NS_TYPE	iif(rgseek(nsubt.ns_type,"ntype"),ntype.nt_type+" "+ntype.nt_desc,nsubt.ns_type)
NTRAN	
NT_TRTYPE	expntype(ntran.nt_trtype)
NT_RVRSE	iif(ntran.nt_rvrse,"y","")
NT_PREVYR	iif(ntran.nt_prevyr,"y","")
NT_CONSOL	iif(ntran.nt_consol,"y","")
NT_FVALUE	sqfcval(ntran.nt_fvalue,ntran.nt_fcdec,.t.)
NT_FCMULT	iif(ntran.nt_fcmult,"y","")
NTYPE	
NT_CLASS	iif(ntype.nt_class="e","expense",iif(ntype.nt_class="i","income",iif(ntype.nt_class="l","liablty",iif(ntype.nt_class="a","asset",iif(ntype.nt_class="c","cap&res","unknown"))))
NVAT	
NV_TYPE	iif(nvat.nv_type="i","invoice","credit")
NV_ADVANCE	iif(nvat.nv_advance,"y","")
PAYROLL	
WBANK	
WK_BKSORT	iif(empty(wbank.wk_bksort),"",left(wbank.wk_bksort,2)+"-"+substr(wbank.wk_bksort,3,2)+"-"+right(wbank.wk_bksort,2))
WCTRL	
WU_CALC	iif(wctrl.wu_calc,"y","")
WU_CBTRANS	iif(wctrl.wu_cbtrans,"y","")
WDEPT	
WA_CNTR	iif(isnull(ncntr.nc_cntr), wdept.wa_cntr, ncntr.nc_cntr+" "+ncntr.nc_desc)
WEAOS	
WAO_TYPE	iif(weaos.wao_type=1,'priority aeo', iif(weaos.wao_type=2,'non-priority aeo',iif(weaos.wao_type=3,'deo',iif(weaos.wao_type=4,'ccaeo',iif(weaos.wao_type=5,'ctaeo',iif(weaos.wao_type=6,'ea','cma'))))))
WAO_PROEPR	iif(weaos.wao_proepr=.t.,'yes','')
WAO_DEDADM	iif(weaos.wao_dedadm=.t.,'yes','')
WAO_TCMPL	iif(weaos.wao_tcmpl=.t.,'yes','')
WAO_LCMPL	iif(weaos.wao_lcmpl=.t.,'yes','')
WAO_PSM03	iif(weaos.wao_psm03, 'y', ' ')
WENIC	
WC_DIRFLG	iif(wenic.wc_dirflg, 'y', ' ')
WEPEN	
WEP_ERPER	iif(wepen.wep_erper=2,"per","val")
WEP_EEPER	iif(wepen.wep_eeper=2,"per","val")
WEP_AVCPER	iif(wepen.wep_avcper=2,"per","val")
WEP_EEBTAX	iif(wepen.wep_eebtax,"yes","no ")
WEP_INACTV	iif(wepen.wep_inactv=.t.,"yes","no")
WEP_ERBTAX	iif(wepen.wep_erbtax=.t.,"yes", "no ")
WESMP	
WW_EELVR	iif(wesmp.ww_eelvr,"y","")
WW_MEDEV	iif(wesmp.ww_medev,"y","")

WW_WORKPC	iif(wesmp.ww_workpc,"y","")
WW_ENTSMP	iif(wesmp.ww_entsmp,"y","")
WW_ACTIVE	iif(wesmp.ww_active,"y","")
WW_MINVAL	iif(wesmp.ww_minval,"y","n")
WESPP	
WPP_EELVR	iif(wespp.wpp_eelvr, 'y', ' ')
WPP_DECLR	iif(wespp.wpp_declr, 'y', ' ')
WPP_ENTSPP	iif(wespp.wpp_entspp, 'y', ' ')
WPP_ACTIVE	iif(wespp.wpp_active, 'y', ' ')
WESSPT	
WI_SSPWTHL	iif(wesspt.wi_sspwthl,"y","")
WI_SSPCOMM	iif(wesspt.wi_sspcomm,"y","")
WETXC	
WB_TTCCMPL	iif(wetxc.wb_ttccmpl,"y","")
WB_LTCCMPL	iif(wetxc.wb_ltccmpl,"y","")
WB_TTCACTV	iif(wetxc.wb_ttcactv,"y","")
WB_LTCACTV	iif(wetxc.wb_ltcactv,"y","")
WHIST	
WH_BINREP	iif(whist.wh_binrep,"y","")
WH_PAYFREQ	iif(whist.wh_payfreq="w","weekly",iif(whist.wh_payfreq="m","monthly",iif(inlist(whist.wh_payfreq,"2","4"),whist.wh_payfreq+"-weekly"," ")))
WH_DIRFLG	iif(whist.wh_dirflg, 'y', ' ')
WH_TAXCHG	iif(whist.wh_taxchg, 'y', ' ')
WH_NIADJ	iif(whist.wh_niadj, 'y', ' ')
WNAME	
WN_TAXCDE	padr(alltrim(wname.wn_stcde + wname.wn_taxcde),8)
WN_BKSORT	iif(empty(wname.wn_bksort),"",left(wname.wn_bksort,2)+"-"+substr(wname.wn_bksort,3,2)+"-"+right(wname.wn_bksort,2))
WL_HOLIP	iif(wname.wl_holip,"y","")
HR_ONLYPER	iif(wname.hr_onlyper,"y","")
WN_P11D	iif(wname.wn_p11d,"y","")
WN_CCCOF	iif(wname.wn_cccof,"y","")
WN_REQP11D	iif(wname.wn_reqp11d,"y","")
WN_SLFLAG	iif(wname.wn_slflag,"y","")
WN_SSPPAYB	iif(wname.wn_ssppayb,"y","")
WL_SSPPAYB	iif(wname.wl_ssppayb,"y","")
WN_TRDISP	iif(wname.wn_trdisp, 'y', ' ')
WN_DIRAPP	iif(wname.wn_dirapp, 'y', ' ')
WN_NIADJ	iif(wname.wn_niadj, 'y', ' ')
WNOML	
WX_TYPE	expntype(wnoml.wx_type)
WX_FVALUE	sqfcval(wnoml.wx_fvalue,wnoml.wx_fcdec,.t.)
WX_FCMULT	iif(wnoml.wx_fcmult,"y","")
WPARM	
WP_COMP	iif(wparm.wp_comp,"y","")
WP_COMPER	iif(wparm.wp_comper,"y","")
WP_COMPNIL	iif(wparm.wp_compnil,"y","")
WP_ANALDEP	iif(wparm.wp_analdep,"y","")
WP_HRSDORM	iif(wparm.wp_hrsdorm,"y","")
WP_EMPHOL	iif(wparm.wp_emphol,"y","")
WP_NLMASK	iif(wparm.wp_nlmask,"y","")
WP_NMNICST	iif(wparm.wp_nmnicst,"y","")
WP_NMNIEER	iif(wparm.wp_nmnieer,"y","")
WP_NMPAYE	iif(wparm.wp_nmpaye,"y","")
WP_NMCOMPR	iif(wparm.wp_nmcompr,"y","")
WP_NMCOMPE	iif(wparm.wp_nmcompe,"y","")

WP_NMWGCON	iif(wparm.wp_nmwgcon,"y","")
WP_NMWGSUS	iif(wparm.wp_nmwgsus,"y","")
WP_NMROUND	iif(wparm.wp_nmround,"y","")
WP_NMERPC	iif(wparm.wp_nmerpc,"y","")
WP_NMHOLC	iif(wparm.wp_nmholc,"y","")
WP_USEGRP	iif(wparm.wp_usegrp,"y","")
WP_CCENTRE	iif(wparm.wp_ccentre,"y","")
WP_TXTONLY	iif(wparm.wp_txtonly,"y","")
WP_P45STD	iif(wparm.wp_p45std,"y","")
WP_HOLPAYC	iif(wparm.wp_holpayc,"y","")
WP_HOLPAYE	iif(wparm.wp_holpaye,"y","")
WP_FULLSSP	iif(wparm.wp_fullssp,"y","n")
WP_SSPQDSU	iif(wparm.wp_sspqdsu,"y","")
WP_SSPQDMO	iif(wparm.wp_sspqdmo,"y","")
WP_SSPQDTU	iif(wparm.wp_sspqdtu,"y","")
WP_SSPQDWE	iif(wparm.wp_sspqdwe,"y","")
WP_SSPQDTH	iif(wparm.wp_sspqdth,"y","")
WP_SSPQDFR	iif(wparm.wp_sspqdfr,"y","")
WP_SSPQDSA	iif(wparm.wp_sspqdsa,"y","")
WP_FULLSMP	iif(wparm.wp_fullsmp,"y","n")
WP_SER	iif(wparm.wp_ser,"y","")
WP_ADVPEN	iif(wparm.wp_advpen,"y","")
WP_TRDISP	iif(wparm.wp_trdisp,"y","")
WP_AOADM	iif(wparm.wp_aoadm=.t.,"y","")
WP_FULLSPP	iif(wparm.wp_fullspp, 'y', ' ')
WP_SSPEOPD	iif(wparm.wp_sspeopd, 'y', ' ')
WP_SMPEOPD	iif(wparm.wp_smpeopd, 'y', ' ')
WPAYD	
WY_TYPE	wgg_exptype_c(left(wpayd.wy_cde,1),wpayd.wy_type)
WY_MASK	iif(wpayd.wy_mask,"y","")
WY_ATTCHBL	iif(wpayd.wy_attchbl=.t.,'yes','')
WY_AOTYPE	iif(wpayd.wy_aotype=1,'priority aeo', iif(wpayd.wy_aotype=2,'non-priority aeo',iif(wpayd.wy_aotype=3,'deo',iif(wpayd.wy_aotype=4,'ccaeo',iif(wpayd.wy_aotype=5,'ctaeo',iif(wpayd.wy_aotype=6,'ea',iif(wpayd.wy_aotype=7,'cma',iif(wpayd.wy_aotype=8,'admin fee
WPNSC	
WPS_EEBTAX	iif(wpnsc.wps_eebtax,"yes", "no ")
WPS_ALLPAY	iif(wpnsc.wps_allpay=.t.,"yes","no ")
WPS_EOYCLR	iif(wpnsc.wps_eoyclr=.t.,"yes","no ")
WPS_SHP	iif(wpnsc.wps_shp=.t.,"yes","no ")
WPS_ERBTAX	iif(wpnsc.wps_erbtax=.t.,"yes","no ")
WPROF	
WF_COMP	iif(wprof.wf_comp,"y","")
WF_PAYMTH	allt(subs("autopaybacs cash giro bobs cheque ecb ",at(wprof.wf_paymth,"abcgoqe")*7-6,7))
WF_DIR	iif(wprof.wf_dir,"y","")
WF_CAR	iif(wprof.wf_car,"y","")
WF_TRF2CB	iif(wprof.wf_trf2cb,"y","")
WF_TRDISP	iif(wprof.wf_trdisp,"y","")
WF_KEEPNPR	iif(wprof.wf_keepnpr,"y","")
WF_KEEPNJB	iif(wprof.wf_keepnjb,"y","")
WSQDDT	
WO_SSPQDSU	iif(wsqddt.wo_sspqdsu,"y","n")
WO_SSPQDMO	iif(wsqddt.wo_sspqdmo,"y","n")
WO_SSPQDTU	iif(wsqddt.wo_sspqdtu,"y","n")
WO_SSPQDWE	iif(wsqddt.wo_sspqdwe,"y","n")

WO_SSPQDTH	iif(wsqddt.wo_sspqdth,"y","n")
WO_SSPQDFR	iif(wsqddt.wo_sspqdfr,"y","n")
WO_SSPQDSA	iif(wsqddt.wo_sspqdsa,"y","n")
WSSMP	
WM_MINVAL	iif(wssmp.wm_minval,"y","n")
WM_MPPST	iif(wssmp.wm_mppst, 'y', ' ')
WTRAN	
WR_TCORIDE	iif(wtran.wr_tcoride,"y","")
PERSONNEL	
HABTRN	
HB_CERTIF	iif(habtrn.hb_certif,"y","")
HB_FHALF	iif(habtrn.hb_fhalf,"y","")
HBENFT	
HF_F1FUELT	expandfueltype(hbenft.hf_f1fuelt)
HF_F2FUELT	expandfueltype(hbenft.hf_f2fuelt)
HF_F3FUELT	expandfueltype(hbenft.hf_f3fuelt)
HF_MSHARE	iif(hbenft.hf_mshare,'y',' ')
HF_OETRAD	iif(hbenft.hf_oetrad,'y',' ')
HF_F1CO2NA	iif(hbenft.hf_f1co2na, 'y', ' ')
HF_F2CO2NA	iif(hbenft.hf_f2co2na, 'y', ' ')
HF_F3CO2NA	iif(hbenft.hf_f3co2na, 'y', ' ')
HF_FTREIN	iif(hbenft.hf_ftrein, 'y', ' ')
HDTTRN	
HP_COMPLET	iif(hdttrn.hp_complet,"y","")
HJHTRN	
HJ_PRMTMP	iif(hjhtrn.hj_prmtmp,"y","")
HJ_FULPRT	iif(hjhtrn.hj_fulprt,"y","")
HNAME	
HR_VERDOB	iif(hname.hr_verdob,"y","")
HR_DISABL	iif(hname.hr_disabl,"y","")
HR_CAROWNR	iif(hname.hr_carownr,"y","")
HR_COMPCAR	iif(hname.hr_compcar,"y","")
HR_LICCLN	iif(hname.hr_liccln,"y","")
HR_C1NXTK	iif(hname.hr_c1nxtk,"y","")
HR_C1EMERG	iif(hname.hr_c1emerg,"y","")
HR_C2NXTK	iif(hname.hr_c2nxtk,"y","")
HR_C2EMERG	iif(hname.hr_c2emerg,"y","")
HR_SMOKER	iif(hname.hr_smoker,"y","")
HR_CCL	iif(hname.hr_ccl,"y","")
HR_CFBL	iif(hname.hr_cfbl,"y","")
HR_TASL	iif(hname.hr_tasl,"y","")
HR_PML	iif(hname.hr_pml,"y","")
HR_SUBSL	iif(hname.hr_subsl,"y","")
HR_EDSUBL	iif(hname.hr_edsubl,"y","")
HR_TELL	iif(hname.hr_tell,"y","")
HR_NURSEL	iif(hname.hr_nursel,"y","")
HR_VOUCHL	iif(hname.hr_vouchl,"y","")
HR_BLOANL	iif(hname.hr_bloanl,"y","")
HR_RELOCL	iif(hname.hr_relocl,"y","")
HR_MISCL	iif(hname.hr_miscl,"y","")
HR_EDCHK	iif(hname.hr_edchk,"y","")
HR_FURTHCH	iif(hname.hr_furthch,"y","")
HR_HOURLY	iif(hname.hr_hourly,"y","")
HR_CARINSB	iif(hname.hr_carinsb,'y',' ')
HR_MOBPER	iif(hname.hr_mobper,'y',' ')

HRPARM	
HM_HRSSAL	iif(hrparm.hm_hrssal,"y","")
HM_HRSDORM	iif(hrparm.hm_hrsdorm,"y","")
HM_HOLENT	iif(hrparm.hm_holent,"y","")
HM_HCACTIV	iif(hrparm.hm_hcactiv,"y","")
HTRTRN	
HN_FHALF	iif(htrtrn.hn_fhalf,"y","")
HWKPAT	
HC_IGBNK	iif(hwkpat.hc_igbnk,'y',' ')
COSTING	
JBHEAD	
JG_STATUS	jcg_expstat_c(jbhead.jg_status)
JG_BTYPE	jcg_expbtype_c(jbhead.jg_btype)
JG_HRSMIN	jcg_hrsmin_c(jbhead.jg_hrsmin)
JBLINE	
JB_CSTTYPE	iif(seek(jbline.jb_csttype,"jctype","jcjecsttyp"),jctype.je_desc,"")
JB_TRTYPE	jcg_exptrtype_c(jbline.jb_trtype)
JB_QTY	rgexpquan(jbline.jb_qty,jccode.ja_factor,jccode.ja_factdps)
JCCODE	
JA_CSTTYPE	iif(seek(jccode.ja_csttype,"jctype","jcjecsttyp"),jctype.je_desc,"")
JA_FACTDSC	iif(empty(jccode.ja_factdsc),"units",jccode.ja_factdsc)
JA_QTY	rgexpquan(jccode.ja_qty,jccode.ja_factor,jccode.ja_factdps)
JA_USED	iif(jccode.ja_used,"y","")
JA_WGVAL	iif(jccode.ja_wgval,"y","")
JCCOST	
JO_EXTCOST	jcg_expextcst_c(jccost.jo_extcost)
JO_OHMULT	iif(jccost.jo_ohmult,"y","")
JCHEAD	
JH_CCFIXED	iif(jchead.jh_ccfixed,"y","")
JH_OPNTYPE	iif(jchead.jh_opntype,"y","")
JH_BUDFIX	iif(jchead.jh_budfix,"y","")
JH_PREVEST	iif(jchead.jh_prevest,"y","")
JCLINE	
JD_QTY	rgexpquan(jcline.jd_qty,jccode.ja_factor,jccode.ja_factdps)
JD_QTYBG	rgexpquan(jcline.jd_qtybg,jccode.ja_factor,jccode.ja_factdps)
JD_QTYRV	rgexpquan(jcline.jd_qtyrv,jccode.ja_factor,jccode.ja_factdps)
JD_WIPR	iif(jcline.jd_wipr,"y","")
JEHEAD	
JH_OPNTYPE	iif(jehead.jh_opntype,"y","")
JH_PREVEST	iif(jehead.jh_prevest,"y","")
JELINE	
JD_QTY	rgexpquan(jeline.jd_qty,jccode.ja_factor,jccode.ja_factdps)
JPARM	
JP_BACKDAT	iif(jparm.jp_backdat,"y","")
JP_HRSMINS	iif(jparm.jp_hrsmins,"y","")
JP_CNTRACT	iif(jparm.jp_cntract,"y","")
JP_PHASE	iif(jparm.jp_phase,"y","")
JP_BATCHC	iif(jparm.jp_batchc,"y","")
JP_BATCHT	iif(jparm.jp_batcht,"y","")
JP_USEPER	iif(jparm.jp_useper,"y","")
JP_HEADBUD	iif(jparm.jp_headbud,"y","")
JP_CRCODES	iif(jparm.jp_crcodes,"y","")
JP_CSTPRN	iif(jparm.jp_cstprn,"y","")
JPLINE	
JR_QTY	rgexpquan(jpline.jr_qty,jccode.ja_factor,jccode.ja_factdps)

JSCTR	
JU_PTAX	iif(jsctr.ju_ptax,"y","")
JU_YTAX	iif(jsctr.ju_ytax,"y","")
JWIPR	
JW_CSTTYPE	iif(seek(jwipr.jw_csttype,"jctype","jcjecsttyp"),jctype.je_desc,"")
JW_TRTYPE	jcg_exptrtype_c(jwipr.jw_trtype)
JW_QTY	rgexpquan(jwipr.jw_qty,jccode.ja_factor,jccode.ja_factdps)
JW_CONFIRM	iif(jwipr.jw_confirm,"y","")
JW_ALLOC	iif(jwipr.jw_alloc,"y","")
JW_TRF2WG	iif(jwipr.jw_trf2wg,"y","")
JW_FRBOM	iif(jwipr.jw_frbom,"y","")
JW_PTAX	iif(jwipr.jw_ptax,"y","")
JW_YTAX	iif(jwipr.jw_ytax,"y","")
STOCK	
CBATCH	
CB_QTY	rgexpquan(cbatch.cb_qty , cfact.cf_factor,cfact.cf_dps)
CB_MOVED	rgexpquan(cbatch.cb_moved , cfact.cf_factor,cfact.cf_dps)
CB_QTYRET	rgexpquan(cbatch.cb_qtyret , cfact.cf_factor,cfact.cf_dps)
CB_STATUS	iif(cbatch.cb_status,"y","")
CB_RETURN	iif(cbatch.cb_return,"y","")
CB_COST	sqfcval(cbatch.cb_cost, cfact.cf_selldps, .t.)
CBATSE	
CA_RELSED	iif(cbatse.ca_relsed,"y","")
CDETL	
CD_SELL	sqfcval(cdetl.cd_sell , cdetl.cd_fundec,.t.)
CD_SSELL	sqfcval(cdetl.cd_ssell ,cdetl.cd_fundec,.t.)
CD_NSELL	sqfcval(cdetl.cd_nsell ,cdetl.cd_fundec,.t.)
CD_SETT	iif(cdetl.cd_sett,"y","")
CD_LINE	iif(cdetl.cd_line,"y","")
CD_OVER	iif(cdetl.cd_over,"y","")
CD_QUAN1	rgexpquan(cdetl.cd_quan1 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN2	rgexpquan(cdetl.cd_quan2 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN3	rgexpquan(cdetl.cd_quan3 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN4	rgexpquan(cdetl.cd_quan4 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN5	rgexpquan(cdetl.cd_quan5 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN6	rgexpquan(cdetl.cd_quan6 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN7	rgexpquan(cdetl.cd_quan7 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN8	rgexpquan(cdetl.cd_quan8 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN9	rgexpquan(cdetl.cd_quan9 , cfact.cf_factor,cfact.cf_dps)
CD_QUAN10	rgexpquan(cdetl.cd_quan10 , cfact.cf_factor,cfact.cf_dps)
CD_SELL1	sqfcval(cdetl.cd_sell1 ,cdetl.cd_fundec,.t.)
CD_SELL2	sqfcval(cdetl.cd_sell2 ,cdetl.cd_fundec,.t.)
CD_SELL3	sqfcval(cdetl.cd_sell3 ,cdetl.cd_fundec,.t.)
CD_SELL4	sqfcval(cdetl.cd_sell4 ,cdetl.cd_fundec,.t.)
CD_SELL5	sqfcval(cdetl.cd_sell5 ,cdetl.cd_fundec,.t.)
CD_SELL6	sqfcval(cdetl.cd_sell6 ,cdetl.cd_fundec,.t.)
CD_SELL7	sqfcval(cdetl.cd_sell7 ,cdetl.cd_fundec,.t.)
CD_SELL8	sqfcval(cdetl.cd_sell8 ,cdetl.cd_fundec,.t.)
CD_SELL9	sqfcval(cdetl.cd_sell9 ,cdetl.cd_fundec,.t.)
CD_SELL10	sqfcval(cdetl.cd_sell10 ,cdetl.cd_fundec,.t.)
CD_SSELL1	sqfcval(cdetl.cd_ssell1 ,cdetl.cd_fundec,.t.)
CD_SSELL2	sqfcval(cdetl.cd_ssell2 ,cdetl.cd_fundec,.t.)
CD_SSELL3	sqfcval(cdetl.cd_ssell3 , cdetl.cd_fundec,.t.)
CD_SSELL4	sqfcval(cdetl.cd_ssell4 ,cdetl.cd_fundec,.t.)
CD_SSELL5	sqfcval(cdetl.cd_ssell5 , cdetl.cd_fundec,.t.)

CD_SSELL6	sqfcval(cdetl.cd_ssell6 ,cdetl.cd_fundec,.t.)
CD_SSELL7	sqfcval(cdetl.cd_ssell7 , cdetl.cd_fundec,.t.)
CD_SSELL8	sqfcval(cdetl.cd_ssell8 ,cdetl.cd_fundec,.t.)
CD_SSELL9	sqfcval(cdetl.cd_ssell9 , cdetl.cd_fundec,.t.)
CD_SSELL10	sqfcval(cdetl.cd_ssell10 ,cdetl.cd_fundec,.t.)
CD_NSELL1	sqfcval(cdetl.cd_nsell1 ,cdetl.cd_fundec,.t.)
CD_NSELL2	sqfcval(cdetl.cd_nsell2 ,cdetl.cd_fundec,.t.)
CD_NSELL3	sqfcval(cdetl.cd_nsell3 ,cdetl.cd_fundec,.t.)
CD_NSELL4	sqfcval(cdetl.cd_nsell4 ,cdetl.cd_fundec,.t.)
CD_NSELL5	sqfcval(cdetl.cd_nsell5 ,cdetl.cd_fundec,.t.)
CD_NSELL6	sqfcval(cdetl.cd_nsell6 ,cdetl.cd_fundec,.t.)
CD_NSELL7	sqfcval(cdetl.cd_nsell7 , cdetl.cd_fundec,.t.)
CD_NSELL8	sqfcval(cdetl.cd_nsell8 ,cdetl.cd_fundec,.t.)
CD_NSELL9	sqfcval(cdetl.cd_nsell9 ,cdetl.cd_fundec,.t.)
CD_NSELL10	sqfcval(cdetl.cd_nsell10 , cdetl.cd_fundec,.t.)
CDISC	
SD_QUAN1	rgexpquan(cdisc.sd_quan1 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN2	rgexpquan(cdisc.sd_quan2 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN3	rgexpquan(cdisc.sd_quan3 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN4	rgexpquan(cdisc.sd_quan4 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN5	rgexpquan(cdisc.sd_quan5 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN6	rgexpquan(cdisc.sd_quan6 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN7	rgexpquan(cdisc.sd_quan7 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN8	rgexpquan(cdisc.sd_quan8 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN9	rgexpquan(cdisc.sd_quan9 , cfact.cf_factor,cfact.cf_dps)
SD_QUAN10	rgexpquan(cdisc.sd_quan10 , cfact.cf_factor,cfact.cf_dps)
CFACT	
CF_SPLIT	iif(cfact.cf_split,"y","")
CF_FIFO	iif(cfact.cf_fifo,"y","")
CF_AVER	iif(cfact.cf_aver,"y","")
CF_STOCK	iif(cfact.cf_stock,"y","")
CF_ISQTY	rgexpquan(cfact.cf_isqty , cfact.cf_factor,cfact.cf_dps)
CF_BATCH	iif(cfact.cf_batch,"y","")
CF_SERIAL	iif(cfact.cf_serial,"y","")
CF_LABOUR	iif(cfact.cf_labour,"y","")
CGHEAD	
CH_CREATOR	iif(empty(cghead.ch_creator) or cghead.ch_creator="drcpt ","direct receipts",iif(cghead.ch_creator="dgrn ","grn receipts",iif(cghead.ch_creator="dpop ","pop receipts","")))
CGLINE	
CI_QTYRCV	rgexpquan(cgline.ci_qtyrcv , cfact.cf_factor,cfact.cf_dps)
CI_ASSGND	rgexpquan(cgline.ci_assgnd , cfact.cf_factor,cfact.cf_dps)
CI_QTYREL	rgexpquan(cgline.ci_qtyrel , cfact.cf_factor,cfact.cf_dps)
CI_QTYRET	rgexpquan(cgline.ci_qtyret , cfact.cf_factor,cfact.cf_dps)
CI_QTYMAT	rgexpquan(cgline.ci_qtymat , cfact.cf_factor,cfact.cf_dps)
CI_COST	sqfcval(cgline.ci_cost ,cfact.cf_selldps,.t.)
CI_VALUE	sqfcval(cgline.ci_value , cfact.cf_selldps, .t.)
CI_SFIED	iif(cgline.ci_sfied,"y","")
CI_CREATOR	iif(empty(cgline.ci_creator) or cgline.ci_creator="drcpt ","direct receipts",iif(cgline.ci_creator="dgrn ","grn receipts",iif(cgline.ci_creator="dpop ","pop receipts",)))
CLIST	
CL_CURR	iif(rgseek(clist.cl_curr,"zxchg"),zxchg.xc_curr+" "+zxchg.xc_desc,clist.cl_curr)
CMATCH	
CM_QTY	rgexpquan(cmatch.cm_qty , cfact.cf_factor,cfact.cf_dps)
CNAME	

CN_CATAG	iif(rgseek(cname.cn_catag,"ccatg"),ccatg.cg_code+" "+ccatg.cg_desc,cname.cn_catag)
CN_SELL	sqfcval(cname.cn_sell , cfact.cf_selldps, .t.)
CN_COST	sqfcval(cname.cn_cost , cfact.cf_selldps, .t.)
CN_LCOST	sqfcval(cname.cn_lcost, cfact.cf_selldps, .t.)
CN_SSELL	sqfcval(cname.cn_ssell, cfact.cf_selldps, .t.)
CN_NSELL	sqfcval(cname.cn_nsell ,cfact.cf_selldps,.t.)
CN_SETT	iif(cname.cn_sett,"y","")
CN_LINE	iif(cname.cn_line,"y","")
CN_OVER	iif(cname.cn_over,"y","")
CN_OANAL	iif(cname.cn_oanal,"y","")
CN_INSTOCK	rgexpquan(cname.cn_instock , cfact.cf_factor,cfact.cf_dps)
CN_FREEST	rgexpquan(cname.cn_freest , cfact.cf_factor,cfact.cf_dps)
CN_ALLOC	rgexpquan(cname.cn_alloc , cfact.cf_factor,cfact.cf_dps)
CN_ONORDER	rgexpquan(cname.cn_onorder, cfact.cf_factor,cfact.cf_dps)
CN_SALEORD	rgexpquan(cname.cn_saleord, cfact.cf_factor,cfact.cf_dps)
CN_SETTSAL	iif(cname.cn_settsal,"y","")
CN_OVERSAL	iif(cname.cn_oversal,"y","")
CN_LINESAL	iif(cname.cn_linesal,"y","")
CN_LINEOVR	iif(cname.cn_lineovr,"y","")
CN_OPANAL	iif(cname.cn_opanal,"y","")
CN_ASSMFLG	iif(cname.cn_assmflg,"y","")
CN_MATCOST	sqfcval(cname.cn_matcost, cfact.cf_selldps, .t.)
CN_LABCOST	sqfcval(cname.cn_labcost , cfact.cf_selldps, .t.)
CN_WOALLOC	rgexpquan(cname.cn_woalloc , cfact.cf_factor,cfact.cf_dps)
CN_WORKORD	rgexpquan(cname.cn_workord , cfact.cf_factor,cfact.cf_dps)
CN_COMPWIP	rgexpquan(cname.cn_compwip , cfact.cf_factor,cfact.cf_dps)
CN_KITTING	iif(cname.cn_kitting,"y","")
CN_ASSMMIN	rgexpquan(cname.cn_assmmin , cfact.cf_factor,cfact.cf_dps)
CN_PHASSM	iif(cname.cn_phassm,"y","")
CN_MODEL	iif(cname.cn_model,"y","")
CPARM	
CP_TRACK	iif(cparm.cp_track,"y","")
CP_LOCFLAG	iif(cparm.cp_locflag,"y","")
CP_WOPRINT	iif(cparm.cp_woprint,"y","")
CP_SUBWKO	iif(cparm.cp_subwko,"y","")
CP_FRCGRN	iif(cparm.cp_frcgrn,"y","")
CP_ASGNRCT	iif(cparm.cp_asgnrct=1,"don't prompt" , iif(cparm.cp_asgnrct=2,"prompt for assignment","don't prompt but assign direct"))
CRLINE	
RI_QTYRET	rgexpquan(crline.ri_qtyret , cfact.cf_factor,cfact.cf_dps)
RI_COST	sqfcval(crline.ri_cost , cfact.cf_selldps, .t.)
RI_VALUE	sqfcval(crline.ri_value, cfact.cf_selldps, .t.)
CSERNO	
CE_RELSED	iif(cserno.ce_relsed,"y","")
CE_ISSUED	iif(cserno.ce_issued,"y","")
CE_STATUS	iif(cserno.ce_status,"y","")
CE_RETURN	iif(cserno.ce_return,"y","")
CE_DISCARD	iif(cserno.ce_discard,"y","")
CE_COST	sqfcval(cserno.ce_cost, cfact.cf_selldps, .t.)
CSPLIT	
CZ_QTY	rgexpquan(csplit.cz_qty , cfact.cf_factor,cfact.cf_dps)
CZ_QTYISS	rgexpquan(csplit.cz_qtyiss , cfact.cf_factor,cfact.cf_dps)
CZ_QTYRET	rgexpquan(csplit.cz_qtyret , cfact.cf_factor,cfact.cf_dps)
CZ_RETURN	iif(csplit.cz_return,"y","")
CZ_DISCARD	rgexpquan(csplit.cz_discard , cfact.cf_factor,cfact.cf_dps)

CSTWH	
CS_INSTOCK	rgexpquan(cstwh.cs_instock, cfact.cf_factor,cfact.cf_dps)
CS_FREEST	rgexpquan(cstwh.cs_freest, cfact.cf_factor,cfact.cf_dps)
CS_ALLOC	rgexpquan(cstwh.cs_alloc, cfact.cf_factor,cfact.cf_dps)
CS_ORDER	rgexpquan(cstwh.cs_order, cfact.cf_factor,cfact.cf_dps)
CS_SALEORD	rgexpquan(cstwh.cs_saleord, cfact.cf_factor,cfact.cf_dps)
CS_REORDL	rgexpquan(cstwh.cs_reordl, cfact.cf_factor,cfact.cf_dps)
CS_REORDQ	rgexpquan(cstwh.cs_reordq, cfact.cf_factor,cfact.cf_dps)
CS_MINST	rgexpquan(cstwh.cs_minst, cfact.cf_factor,cfact.cf_dps)
CS_SUPCOS1	sqfcval(cstwh.cs_supcos1, cfact.cf_selldps, .t.)
CS_SUPCOS2	sqfcval(cstwh.cs_supcos2, cfact.cf_selldps, .t.)
CS_SUPCOS3	sqfcval(cstwh.cs_supcos3, cfact.cf_selldps, .t.)
CS_ISS1	rgexpquan(cstwh.cs_iss1, cfact.cf_factor,cfact.cf_dps)
CS_ISS2	rgexpquan(cstwh.cs_iss2 , cfact.cf_factor,cfact.cf_dps)
CS_ISS3	rgexpquan(cstwh.cs_iss3 , cfact.cf_factor,cfact.cf_dps)
CS_ISS4	rgexpquan(cstwh.cs_iss4 , cfact.cf_factor,cfact.cf_dps)
CS_ISS5	rgexpquan(cstwh.cs_iss5 , cfact.cf_factor,cfact.cf_dps)
CS_ISS6	rgexpquan(cstwh.cs_iss6 , cfact.cf_factor,cfact.cf_dps)
CS_ISS7	rgexpquan(cstwh.cs_iss7 , cfact.cf_factor,cfact.cf_dps)
CS_ISS8	rgexpquan(cstwh.cs_iss8 , cfact.cf_factor,cfact.cf_dps)
CS_ISS9	rgexpquan(cstwh.cs_iss9 , cfact.cf_factor,cfact.cf_dps)
CS_ISS10	rgexpquan(cstwh.cs_iss10 , cfact.cf_factor,cfact.cf_dps)
CS_ISS11	rgexpquan(cstwh.cs_iss11 , cfact.cf_factor,cfact.cf_dps)
CS_ISS12	rgexpquan(cstwh.cs_iss12 , cfact.cf_factor,cfact.cf_dps)
CS_ISS13	rgexpquan(cstwh.cs_iss13 , cfact.cf_factor,cfact.cf_dps)
CS_SHLDCST	sqfcval(cstwh.cs_shldcst, cfact.cf_selldps, .t.)
CS_COST	sqfcval(cstwh.cs_cost ,cfact.cf_selldps,.t.)
CS_SELL	sqfcval(cstwh.cs_sell , cfact.cf_selldps, .t.)
CS_LCOST	sqfcval(cstwh.cs_lcost , cfact.cf_selldps, .t.)
CS_WOALLOC	rgexpquan(cstwh.cs_woalloc , cfact.cf_factor,cfact.cf_dps)
CS_WORKORD	rgexpquan(cstwh.cs_workord , cfact.cf_factor,cfact.cf_dps)
CS_COMPWIP	rgexpquan(cstwh.cs_compwip , cfact.cf_factor,cfact.cf_dps)
CTITEM	
CU_ALLISS	iif(ctitem.cu_alliss,"y","")
CTRACK	
CK_QTY	rgexpquan(ctrack.ck_qty , cfact.cf_factor,cfact.cf_dps)
CTRAN	
CT_TYPE	expandst(ctran.ct_type)
CT_QUAN	rgexpquan(ctran.ct_quan , cfact.cf_factor,cfact.cf_dps)
CT_MOVED	rgexpquan(ctran.ct_moved , cfact.cf_factor,cfact.cf_dps)
CT_COST	sqfcval(ctran.ct_cost ,cfact.cf_selldps,.t.)
CWARE	
CW_REC	iif(cware.cw_rec,"y","")
CW_ISS	iif(cware.cw_iss,"y","")
CW_RET	iif(cware.cw_ret,"y","")
CW_TRANF	iif(cware.cw_tranf,"y","")
CW_RESNEG	iif(cware.cw_resneg,"y","")
CW_QUANT	iif(cware.cw_quant,"y","")
BILL OF MATERIALS	
CHEAD	
CX_CANCEL	iif(chead.cx_cancel,"y","")
CX_ORDQTY	rgexpquan(chead.cx_ordqty , cfact.cf_factor,cfact.cf_dps)
CX_MADEQTY	rgexpquan(chead.cx_madeqty , cfact.cf_factor,cfact.cf_dps)
CX_KITTING	iif(chead.cx_kitting,"y","")
CX_WIPQTY	rgexpquan(chead.cx_wipqty , cfact.cf_factor,cfact.cf_dps)

CX_ALLOQTY	rgexpquan(chead.cx_alloqty , cfact.cf_factor,cfact.cf_dps)
CX_DISCQTY	rgexpquan(chead.cx_discqty , cfact.cf_factor,cfact.cf_dps)
CLINE	
CY_REQQTY	rgexpquan(cline.cy_reqqty,cfact.cf_factor,cfact.cf_dps)
CY_CMPLQTY	rgexpquan(cline.cy_cmplqty,cfact.cf_factor,cfact.cf_dps)
CY_PRICE	stg_strprice_c(cline.cy_price,cline.cy_cnref,.t.,12)
CY_ISASSM	iif(cline.cy_isassm,"y","")
CY_PHASSM	iif(cline.cy_phassm,"y","")
CY_TOMAKE	rgexpquan(cline.cy_tomake , cfact.cf_factor,cfact.cf_dps)
CY_FROMST	rgexpquan(cline.cy_fromst , cfact.cf_factor,cfact.cf_dps)
CY_ALLOQTY	rgexpquan(cline.cy_alloqty , cfact.cf_factor,cfact.cf_dps)
CY_WIPQTY	rgexpquan(cline.cy_wipqty , cfact.cf_factor,cfact.cf_dps)
CY_COQUANT	rgexpquan(cline.cy_coquant , cfact.cf_factor,cfact.cf_dps)
CY_TOPROCE	rgexpquan(cline.cy_toproce , cfact.cf_factor,cfact.cf_dps)
CY_ASSMMIN	rgexpquan(cline.cy_assmmin , cfact.cf_factor,cfact.cf_dps)
CY_SHOW	iif(cline.cy_show,"y","")
CY_LABOUR	iif(cline.cy_labour,"y","")
CY_STOCK	iif(cline.cy_stock,"y","")
CSTRUC	
CV_COQUANT	rgexpquan(cstruc.cv_coquant , cfact.cf_factor,cfact.cf_dps)
CV_SUBASSM	iif(cstruc.cv_subassm,"y","")
CV_PHASSM	iif(cstruc.cv_phassm,"y","")

FAQS

Question	Answer
Can I use a SEEK Expression in a Reporter report?	Not really. Due to the Data Environment the Reporter Report is created in, a SEEK or SQOPEN would not offer the correct information or outcome. It does depend on what the SEEK is being used for – but be aware this could publish incorrect information or force the report to be wrong.
Can I use ? in the expression builder?	Yes you can. This is covered in the Expression Builder section.
Where are the joins for the tables?	These are held in the SEQJOINS table.
Will Reporter Reports pick up bespoke tables and fields?	Yes as long as they are designed to update the Data Dictionary and a UDS has been run – this will pull through into Reporter Reports.
Can I turn a Standard VFP report into a Reporter Report?	No, these are pre-defined Reports assigned to the respective module. Some hold some complex expressions and some work with Arrays etc.

Appendix B - Report Base & Linked Tables

The choice of 'Report Base' when designing a report, will dictate which data tables and fields are available on the report. The following summarises which data tables can be reported upon, depending upon the 'Report Base' table selected.

COMPANY		
Notes	VAT	
Notes Accounts (NL) Names Accounts (PL) Accounts (SL) Customised Fields	VAT Data Collection Exchange Rates Countires Commodities Supplier Profile Terms Profiles (PL) Customer Profile Terms Profiles (SL) Accounts (SL) Accounts (PL) Calculations Customised Fields	

SYSTEM		
Payroll AO Rates	People Table	
Payroll AO Rates Customised Fields	People Table Customised Fields	

CASHBOOK		
Cashbook Audit File	Entries	Transactions Lines
Customised Fields	Banks Cost Centres Factors Accounts Budgets Cashbook Types Customised Fields	Banks Cost Centres Factors Accounts Budgets Entries Customised Fields
Recurring Trans	Nominal Transfer	
Recurring Entries Customised Fields	Customised Fields	

NOMINAL		
Accounts	Banks	Pre-Journals
Types Sub Types Banks Cost Centres Factors Budgets Calculations Customised Fields	Cost Centres Accounts Exchange Rates ECB Software Names Customised Fields	Types Sub Types Banks Cost Centres Factors Accounts Budgets Pre-Jrl Headers Customised Fields
Nominal VAT Transfer	Transactions	History
Customised Fields	Transactions Types Sub types Banks Cost Centres Factors Accounts (NL) Budgets Nominal Projects Nominal Department Calculations Customised Fields	Nominal History Types Sub Types Cost Centres Accounts (NL) Nominal Project Nominal Department Customised Fields

PURCHASE LEDGER		
Accounts	Allocations	Analysis
Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles Calculations Customised Fields	Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles Accounts Customised Fields	Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles Accounts Customised Fields
History	Nominal Transfer	PIR Header
Supplier Profiles Supply Codes Trade Terms Profiles Accounts Customised Fields	Types Sub Types Banks Cost Centres Factors Accounts Budgets Customised Fields	Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles Accounts Customised Fields
PIR Lines Detail	PIR Match Table	Transactions
Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles PIR Header Names Profiles Accounts Customised Fields	Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles PIR Header Order Lines Names Profiles PIR Lines Detail Accounts Customised Fields	Exchange Rates Countries Supplier Profiles Supply Codes Trade Terms Profiles Accounts Customised Fields
Supplier Audit File		
Customised Fields		

POP		
Authorities	Individuals	Purchasing Department
Authorities Individuals Purchasing Department Customised Fields	Individuals People Table Customised Fields	Purchasing Department Customised Fields
Order Lines	Model Order Lines	Requisition Lines
Order Lines Exchange Rates Supplier Profiles Terms Profiles (PL) Requisition Headers Requisition Lines Supplier Products Uncommitted Document Purchase Orders Warehouse Stock Per Warehouse Names Profiles Accounts (PL) Nominal Project Nominal Department Calculations Customised Fields	Model Order Lines Exchange Rates Supplier Profiles Terms Profiles (PL) Supplier Products Madel Order Header Warehouse Stock Per Warehouse Names Profiiles Accounts Nominal Project Nominal Department Calculations Customised Fields	Requisition Lines Supplier Profiles Terms Profiles (PL) Requisition Headers Names Profiles Accounts (PL) Calculations Customised Fields
Supplier Products		
Supplier Products Exchange Rates		

Supplier Profiles Terms Profiles Names Profiles Accounts (PL) Calculations Customised Fields		

SALES LEDGER		
Accounts	Allocations	Analysis
Exchange Rates Countries Warehouses Customer Profiles Trade Terms Profiles Calculations Customised Fields	Exchange Rates Customer Profiles Trade Terms Profiles Accounts Customised Fields	Exchange Rates Countries Commodities Sales Codes Names Profiles Categories Customer Profiles Trade Terms Profiles Accounts Customised Fields
Transactions	History	Nominal Transfer
Exchange Rates Customer Profiles Trade Terms Profiles Accounts Customised Fields	Sales Codes Names Profiles Customer Profiles Trade Terms Profiles Accounts Customised Fields	Types Sub Types Banks Cost Centres Factors Accounts Budgets Customised Fields
Customer Audit File		
Customised Fields		

Invoicing SOP		
Price Details	Transactions	Allocations
Price Details Exchange Rate Names Profiles Price List Caclulations Customised Fields	Transactions Exchange Rates Countires Commodities Sales Codes Warehouse Stock Per Warehosue Names Profiles Customer Profiles Terms Profiles (SL) Headers Accounts (SL) Nominal Project Nominal Department Caclulations Customised Fields	Allocations Countries Commodities Sales Codes Warehouse Names Profiiles Headers Accounts (SL) Calculations Customised Fields
Names		
Names Profiles Customised Field		

STOCK		
Batch Numbers	Batch Splits	Carriers
G.R.N. Lines Names Profiles G.R.N. Header	Warehouses Names Profiles Categories	Customised Fields

Customised Fields	Customised Fields	
G.R.N. Header	G.R.N. Lines	G.R.N. Lines Matched
Carriers Customised Fields	Sales Codes Supply Codes Warehouses Names Profiles G.R.N. Header Carriers Categories Accounts Customised Fields	G.R.N. Lines Sales Codes Supply Codes Order Lines Warehouses Names Profiles Categories Accounts Customised Fields
Names	Price Details	R.T.V. Header
Countries Commodities Sales Codes Profiles Calculations Customised Fields	Exchange Rates Names Profiles Price List Categories Customised Fields	Carriers Customised Fields
R.T.V. Lines	Stock Batched/Serial	Stock Parameters
Sales Codes Supply Codes Warehouses Names R.T.V. Header Profiles Carriers Categories Accounts Customised Fields	Customised Fields	Customised Fields
Stock Per Warehouse	Stock Traceable Item	Tracking of Items
Exchange Rates Countries Commodities Sales Codes Supplier Profiles Supply Codes Trade Terms Profiles Warehouses Names Profiles Accounts Customised Fields	Customised Fields	Warehouses Stock Per Warehouse Names Profiles Customised Fields

BOM		
Assembly Components	W/Order Lines	BOM W/Orders
Assembly Components Countries Commodities Sales Codes Names Profiles Calculations Customised Fields	W/Order Lines Countries Commodities Sales Codes Assembly Components Names Profiles Works Orders Calculations Customised Fields	Works Orders Countries Commodities Sales Codes Assembly Components Names Profiles Calculations Customised Fields

COSTING		
Batch Details	Job Detail	Job Headers
Batch Details	Job Detial	Job Headers

Profiles Exchange Rates Payments and Deductions Countries Sales Codes Supply Codes Contracts Subcontractors Batch Header Cost codes Job Headers Job Details Warehouses Stock Per Warehouse Names Profiles (STK) Customer Profiels Terms Profiles (SL) Employees Accounts (SL) People Table Calculations Customerised	Exchange Rates Coutnries Commodities Sales Codes Contracts Framework Cost Codes Cost Code Rates Job Headers Names Profiles Customer Profiles Terms Profiles (SL) Accounts People Table Department Calculations Customised Fields	Payments And Deductins Sales Codes Supply Codes Contracts Framework Cost Codes Cost Codes Rates Names Profiles Accounts (SL) People Table Department Calculations Customised Fields
Estimate Details	Estimate Headers	Work In Progress
Estimate Details Exchange Rates Countires Sales codes Contracts Estimate Headers Framework Cost Codes Names Profiles Customer Profiles Terms Profiles (SL) Accounts (SL) People Table Department Calculations Customised Fields	Estimate Headers Payments And Deductions Sales Codes Supply Codes Contracts Framework Cost Codes Cost Codes Rates Names Profiles Accounts (SL) People Table Department Calculations Customised Fields	Work In Progress Profiiles Exchange Rates Payments and Deductions Countries Sales Codes Supply Codes Terms Profiles (PL) Contracts Subcontractors Cost Codes Cost Codes Rates Job Header Job Detail Warehouse Stock Per Warehouse Names Profiles Customer Profiles Terms Profiles (SL) Employees Accounts Accounts (SL) People Tables Calculations Customised Fields
Cost Codes	Contracts	Subcontractors Transactions
Cost Codes Sales Codes Supply Codes Names Profiles Calculations Customised Fields	Contracts Exchange Rates Countires Customer Profiles Terms Profikesl (SL)# Accounts People Tables Calculations Customerised Fields	Subcontractor Transactions Work In Progress Subcontractors Cost Codes People Table Calculations Customised Fields
Subcontractors	Framework Details	
Subcontractors People Table Customised	Framework Details Sales Codes Supply Codes Framework Cost Codes Names	

	Profiles Calculations Customised Fields	

PAYROLL		
Employees	History	Nominal Transfer
Cost Centres Options Profiles Departments Groups Bank Details Training Absence Reasons Discipline Ethnic Personnel Employees Customised Fields	Cost Centres Profiles Departments Bank Details Employees Customised Fields	Types Sub Types Banks Cost Centres Factors Accounts Budgets Customised Fields
SCON Transactions	Transactions	
Cost Centres Profiles Departments Bank Details Employees Customised Fields	Cost Centres Profiles Payments & Deductns Departments Bank Details Employees Customised Fields	

Personnel		
Absence Records	Discipline Records	Employee Patterns
Absence Records Cost Centres Options Profiles Departments Groups Bank Details Absence Reasons Employees Personnel Employees Customised Fields	Dicipline Records Cost Centres Options Profiles Departments Groups Bank Details Dicipline Employees Personnel Employees Customised Fields	Employee Patterns Cost Centres Options Profiles Departments Groups Bank Details Employees Personnel Employees Pattern Headers Customised Fields
Job Details	Pattern Details	Personnel Employees
Job Details Cost Centres Options Profiles Departments Groups Bank Details Jobs File Employees Personnel Employees Customised Fields	Pattern Details Pattern Headers Customised Fields	Personnel Employees Cost Centres Options Profiles Departments Groups Bank Details Ethnic Marital Status Files Employees Benefits Customised Fields
Salary History	Training Records	
Salary History Cost Centres Options Departments Groups Bank Details Employees Personnel Employees Customised Fields	Training Records Cost Centres Options Profiles Departments Groups Bank Details Training Employees Personnel Employees Customised Fields	

Accreditation Training Courseware

Other Guides and Help

The following user documents are also provided with the respective software. Guides are available from various sources including Answerlink, the Opera 3 CD and from the Opera 3 Client installation's main folder, normally *C:\Program Files\Pegasus\Client XXX\User documents (where XXX is either VFP or SQL)*. Some guides are also available from the Opera 3 Help menu or respective software Help menu.

Readme File

The Readme file is an HTML document. It contains late breaking news and any other release specific important information.

Note: This file can also be accessed from the Opera 3 CD Installation Menu with regards to Opera 3.

Release Guide

When new versions are released such as Opera 3, they are normally accompanied by a Release Guide that contains information such as What's New in this Release, Release History, and Functional Differences between versions.

Note: This guide can also be accessed from the Opera 3 CD Installation Menu with regards to Opera 3.

Installation Guide

The Installation Guide is provided to help you install the application in your chosen Windows environment. It contains technical information and advice to ensure the installation process is successful.

Demonstration Data Guide

This guide is provided to assist the use of the demonstration data and contains pertinent information that is not contained elsewhere in the product documentation. It is recommended that the Help be used at the same time.

Implementation Guide

The Implementation Guide provides a high-level view of an application such as Opera 3 and how to approach an implementation. It details how the modules are integrated, and recommends the order in which elements should be set up.

System Requirements Guide

The System Requirements Guide details the minimum and recommended computer hardware and software requirements that are required to install and use software such as Opera 3 Enterprise and Opera 3 Enterprise SQL.

It also includes guidance on using Opera 3 Enterprise and Opera 3 Enterprise SQL in a Terminal Services environment run in Windows Server 2003 and Windows Server 2008 (32-bit & 64-bit).

Help

F1 Help provides details of all the functions available in the application.

Document Management & Executive Dashboards Guides

Installation Guides are available for both Document Management and Executive Dashboards. A user guide is also available for Executive Dashboards. Help for Document Management is in the Opera 3 Help.

Note: The guides are in the respective folders on the Opera 3 CD and can be accessed from the Opera 3 CD Installation Menu.