Getting Started

COPYRIGHT
1989 - 2000 by **ZUCCHETTI TOOLS** S.r.l.
All rights reserved.
This publication contains information protected by copyright. This publication may not be reproduced, stored, or transmitted, in any form or by any means without the prior permission of the publisher.

TRADEMARKS
All produced trademarks are ownership of the holder and are acknowledged by this publication.

ZUCCHETTI TOOLS S.r.l. SOFTWARE TECHNOLOGY
PADOVA - BELLARIA - RIMINI
E-mail: clabrn@codelab.it
Web Address:
http:\\www.zucchettitools.com
http:\\www.codepainter.com
http:\\www.codelab.it

Index

Capitolo 1

Introduction

For years hardware development has aimed at improving processing speed and memory space, changing its focus from mainframes to personal computers.

1.1 Software Application Development

Today businesses require global IT solutions, which include powerful software. Software application development concerns defining 'WHAT' needs to be done and 'HOW' things need to be done. In addition software needs 'MAINTENANCE', maybe due to changes in the environment, changed requirements, or software bugs.

A successful software generation takes the commissioner into account. The 'What' definition will take longer, because analyst and commissioner must understand each other and accurate data must be passed on to the programmer. Further the analyst must understand the impact the solution will have on the business and communicate it to the commissioner so that if required corrective action can be taken in an early stage of the project.

A software generation project usually goes through the classical six steps:

Analysis, where minimum performance, information requirements, needed functionalities, existing boundaries, validation criteria, and interfaces to other software applications, etc. are defined.

Project, where project specifications are written basing on information collected during customer interviews, the data structure is defined and the problem is divided into smaller sub-problems.

Coding, where project specifications are coded, program lines written, documents structured, etc.

Testing, where technical and functional specifications are compared with running programs. Further smaller software blocks are tested to verify that they work together smoothly. Testing programs is about finding as many errors as possible.

Software **Delivery**. During normal day-to-day running of the software users will find errors that are unlikely to be found in a testing environment. End-users therefore must be trained thoroughly and users reference guides must be delivered and accessible.

Maintenance, where errors are fixed, and adaptations and/or improvements are made.

1.2 Life Cycle Models

The six steps can have different timings and durations. This is the reason why different life cycle models have been developed.

The Classic Model

With this method steps occur one after the other. A phase cannot start before the previous has been completed. Even if widely used, this model is too simplistic, because each phase is seen as a stand-alone phase. Misunderstandings and errors are often detected once the program is developed and used. Going back to the phase in which the error was generated can become costly.

The Rapid Prototype

After an initial problem analysis a prototype is developed, i.e. a reduced version of a program, in which efficiency and control are not considered, but that allows revising and finalizing the analysis quickly. Once the analysis is completed, the prototype is abandoned and the real program can be developed. Using this model implies having tools to write programs quickly.

The Cyclic Model

This method is often used in dynamic environments. It suites products that grow fast during maintenance, and allows going through all life cycle stages again. Once the commissioner sees specifications tangibly, additional requirements are often defined. Adequate tools must support the programmer, because a too fast software development with program patches could make the data architecture soon inefficient.

1.2.1 The CODEPAINTER Methodology

CODE**PAINTER REVOLUTION** is a C.A.S.E. (Computer Aided Software Engineering) tool, which guides and coordinates each phase of software development.

Object Oriented Analysis

An Object Oriented Tool supports the analysis in CodePainter. Using this tool allows dividing the issue in various objects, recognizing object attributes (i.e. data) and methods (i.e. procedures), describing object relationships, and applying object inheritance correctly. This tool has a set of predefined object classes, such as Master File, Detail File, Master/Detail, Dialog Window, and Routine, which are the most frequently used in business software application.

The analysis methodology used by CodePainter is different from the one used by general object based software. There is no need to define new classes; indeed an object is assigned to one of the predefined classes making the analyst's job easy as known tools are used. The drawback of this methodology being that its use is limited to business issues.

Developing a software project in CodePainter means starting with the *Design* phase. Objects representing the problem are defined. Each object is assigned to one of the predefined classes. In order to lead the code generation, a table will be assigned to each class. Relationships can be established for CodePainter objects, which from now on will be called *Entities*. Completing the software design means obtaining a plan summarizing the entire application, which is ready for the next stage: prototyping. The software design can be changed at any time during the software development. Entities can be changed or new ones can be added to the plan.

The Prototype

Prototypes can be considered bridges between design and coding. The CodePainter prototyping tool generates a code exploiting standard definitions established for classes, to which the defined entities are assigned; e.g. Master Files have templates for data input and output.

On one hand prototypes disclose defined specifications very quickly; on the other hand they represent the first step towards coding. Coding indeed must not be regenerated, but only requires improving what the prototyping tool generated. In addition prototypes help finalizing the software design, as they can be developed so quickly that can be used to discuss requirements with clients.

This tool supports the Rapid Prototype Software Life Cycle.

Template Driven Code Generator

In the 'Codify Phase' CodePainter provides specialized tools for each predefined class. With these tools the programmer can work on entities' masks, and can define specifications for each entity's element implementing the WYSIWYG (What You See Is What You Get) methodology with the mouse. Specifications are translated in code, which can be run using templates. Templates are program skeletons, which together with entities definitions form the complete code.

Entities can use standard templates or, if the issue is complex, specialized ones can be defined. One advantage of this feature is that code generated by a template has been tested and verified before in other similar programs and therefore is error-free. In addition it is strongly standardized and programmers easily understand the code, being able to perform all required maintenance. Programmers are therefore no longer bound by problems encountered with individual programming languages. The main advantage of CodePainter 'Codify Tools' is that both code and modifications are generated in a few seconds only, a time frame, which is indeed, a fraction compared to other similar methods.

Code Integration

Auto-generated coding does not overcome classic issues completely. CodePainter has many options supporting code generation, but for complex software applications, programmers may have to add some code manually. The auto-generated code is an ASCII file, which is identical to manually written files. Integrating the code is therefore possible, but what happens with the integrated code if the program is re-generated? Simple generators would regenerate the program loosing manually integrated code, this having heavy repercussions on the software life cycle. CodePainter overcomes the issue using the 'manual areas' technique. In each auto-generated program there are plain areas, in which the programmer can introduce code manually. When the code is re-generated these areas are maintained.

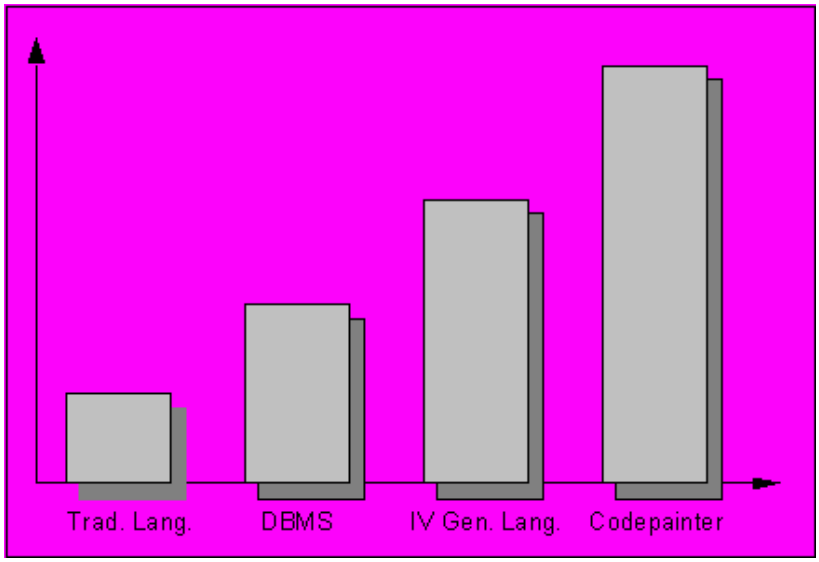
Documentation

CodePainter has an overall understanding of the project at different detail levels. Using specialized tools and integrating definitions specified at various project levels the programmer can develop user reference guides. Technical user reference guides are developed from design level. These include software project expectations, database description, and the list of procedures, which need being created. Technical user reference guides are generated before actual programs are developed and therefore can be used for producing cost estimates and to focus on client's specifications.

End-user reference guides are written combining design definitions with actual masks, and are always up-dated with the latest program version. This functionality saves time and costs in producing manuals and reference guides, and also helps programmers that are often more at ease with programming language, writing documentation, which is understandable to the end-user.

1.3 Increasing Productivity

CodePainter fundamental objective is increasing programmers' productivity (as shown in Picture 4) while improving software quality.



Picture 1 - Source Zucchetti Tools - Estimating Increasing Productivity (which can be obtained using CODEPAINTER as opposed to other traditional development methods)

CODEPAINTER REVOLUTION integrates complexity and specificity of traditional programming languages with the data structure versatility given by databases. Software development is guided and managed by object oriented programming, which is the cornerstone of this product.

Capitolo 2

Creating a Software Application

2.1 Introduction

This user reference guide leads you through the creation of a simple software application using CodePainter tools. The underlying principle of this guide is that hands-on experience is the best way to learn using this tool.

Getting Started will guide you step by step through the logical flow of procedures required to create a small warehouse management software and the following implementation and maintenance.

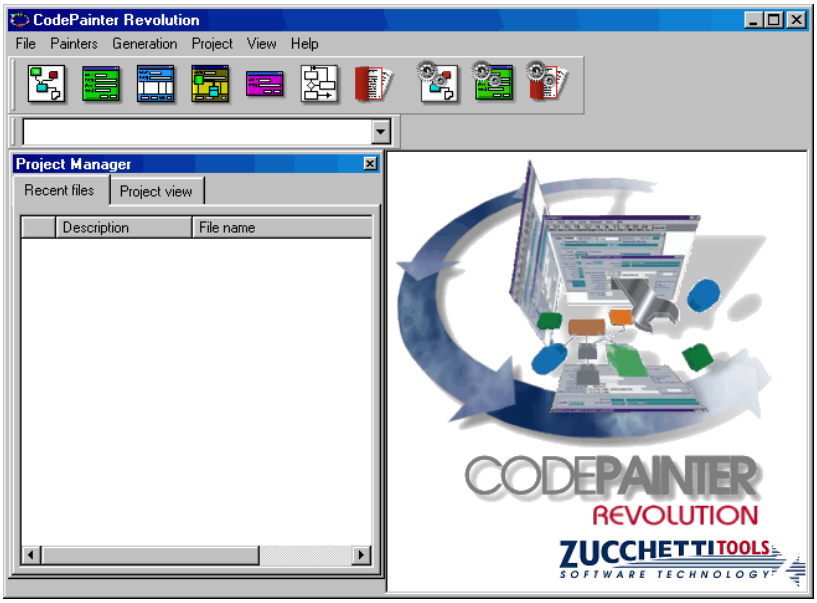
To guarantee the success of this exercise you need to install the source language MS Visual FoxPro 6.x

2.2 First Steps

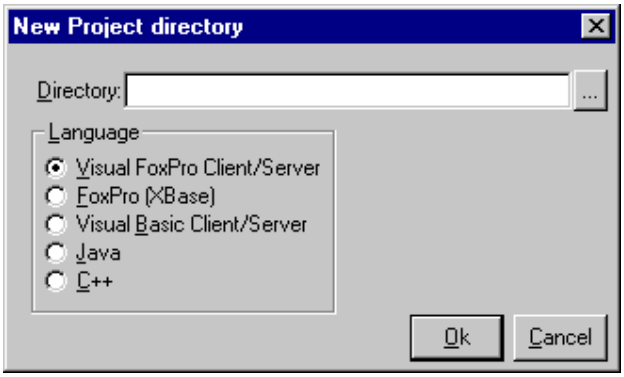
CodePainter can be started by clicking the icon in the CodePainter group. This will work only if you have followed the installation notes.

The first screen is the CodePainter Front-End (for further detail please refer to the "User's Reference Guide"). Starting from here you can access all the available tools.

Picture 2 - CodePainter's Front-End



First of all you must create a directory in which you will save your software application. Open the 'File' menu and select 'New Project'. You have opened the 'New Project Directory' window, where you will define the directory name in which you want to develop your software application, and the source language.



Picture 3 - Directory Definition

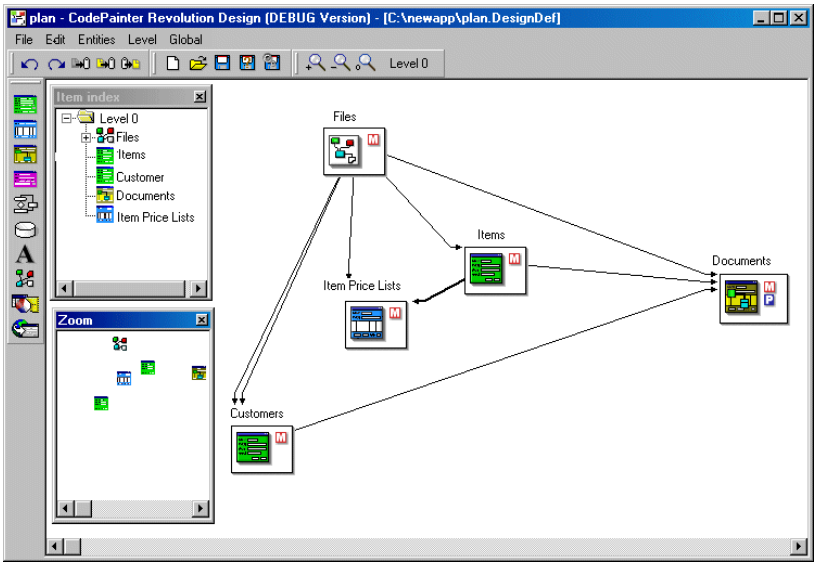
Using the '…' button, you can use the Window browser to select the desired path. Alternatively you can create a new path writing directly in the corresponding window field. Then select the source language. For the purpose of this exercise please assume you have chosen 'Visual FoxPro Client/Server'. The directory is created and all files you will require for development are copied in this directory. You have gone back to the CodePainter front-end. Please note the icons on the toolbar, which represent the various components required to develop the software application.

2.3 The Design Methodology

The Design Tool helps you defining the underlying structure, i.e. the databases.

On the toolbar click this icon to activate the Design Painter. The various represented objects allow you acting on single entities, defining type, properties, and links with other entities.

Picture 4 - Example of a Design



You can select from the following predefined entity types: Master File, Detail File, Master/Detail, Dialog Window, Routine and Database table. You can insert descriptive strings, create groups organizing the project plan, insert visual entities and recall external entities. Let's quickly review the entity types.

2.3.1 Master File Entities

Master File entities are made of a database table and a program, which allow to input and modify the data. They have a flat structure, i.e. without repeated elements. They manage databases with one-by-one record processing, e.g. they are used to manage customer files, in which one record per customer will be created. When you work on these files, records will be modified one-by-one.

Warehouse items are another typical example of Master Files, as shown in the table below:

Item	Description
01	Item 1
02	Item 2
03	Item 3

2.3.2 Master/Detail Entities

Business transactions are usually made using documents, like orders, invoices, etc. These documents are made of three distinct blocks: header, body and footer. This structure is managed processing a record containing header and footer, and a second record containing the body. Fields defined in the body are said to be 'repeated', because the same document may have more instances of the same field, e.g. an invoice will have instances of the item number field and item description field. Despite the two record types, Master/Detail entities appear as a single structure.

Invoice No.	Invoice Date	Customer Code	Total
01	10/01/2000	01	540
02	20/02/2000	05	2070

Invoice No.	Item No.	Quantity	VAT	List Price	Amount
01	01	150	20	1	180
01	02	100	20	3	360
02	04	50	20	0.50	30
02	06	1	20	650	780
02	07	30	20	35	1260

The two files are linked trough the 'Invoice Number' field. To the end-user the two files will remain separate and transparent.

2.3.3 Detail File Entities

The use of *Detail* File entities is preferred when a document has little information in header and footer, and all repeated fields are part of the database table's primary key containing the body. Warehouse transactions are a typical example:

Item No.	Flow Code	Flow Type	Date	Quantity
01	01	Inflow	10/01/2000	1000
01	02	Outflow	20/01/2000	150
01	03	Outflow	15/02/2000	500
02	01	Inflow	08/01/2000	800
02	02	Outflow	15/01/2000	300

2.3.4 Dialog Windows

Dialog Window entities are graphic user interfaces (GUI) entailing variables, strings, icons, etc. They do not have fields, and therefore no associated files. Dialog Windows are used for data input or to display transaction results.

2.3.5 Routines

Routine entities are procedures or functions called into programs. The Routine Painter helps you creating procedures for data work-out, or for importing/ exporting files or functions, e.g. for field validation.

2.3.6 Database Tables

Database Table entities are files defined in the design phase that have no dialog window. These could be used as support files, as export result or as an import basis.

2.3.7 Strings

Strings allow writing notes against other entities in the plan.

2.3.8 Groups

Groups could be compared to folders, allowing organizing entities logically. A plan could have a group for accounting files, one for customer files and one for warehouse item files.

2.3.9 Output Entities

With Output entities you could refer to procedure entities such as print, query, graphs, etc. These are defined during the execution of the software application using visual tools.

2.3.10 External Entities

External entities allow you to use entities of other design plans. They could also be used to link databases of different software applications.

You can maintain entities in general using tools, called Painters. You can link entities together to highlight relationships and the actual flow of data.

2.4 The Project Plan

The warehouse management software application you are about to create, will have a warehouse item file, a file with different document types, and, last but not least, will manage these documents.

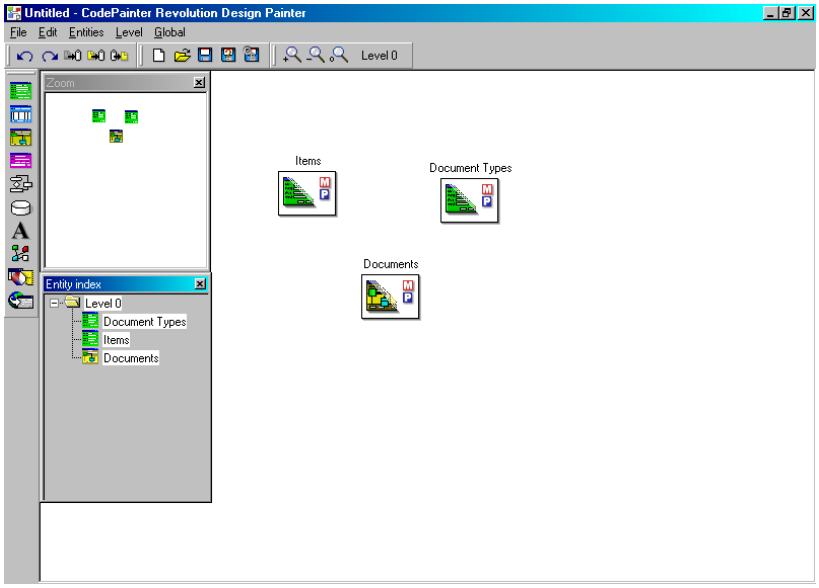
To recap, you are in the Design Painter, which has many toolbars. The one on the left screen side has icons representing the entities, available within the Design Painter.

Click the first, green icon, the Master File. The mouse has changed into a 'plus' with the description 'ADD'. Click the white background to add a Master File. The added entity is displayed with a half table on a white background with the string tagged "NoName" and the two flags "M" and "P". The half table tells you that not all compulsory parameters have been defined for this entity.

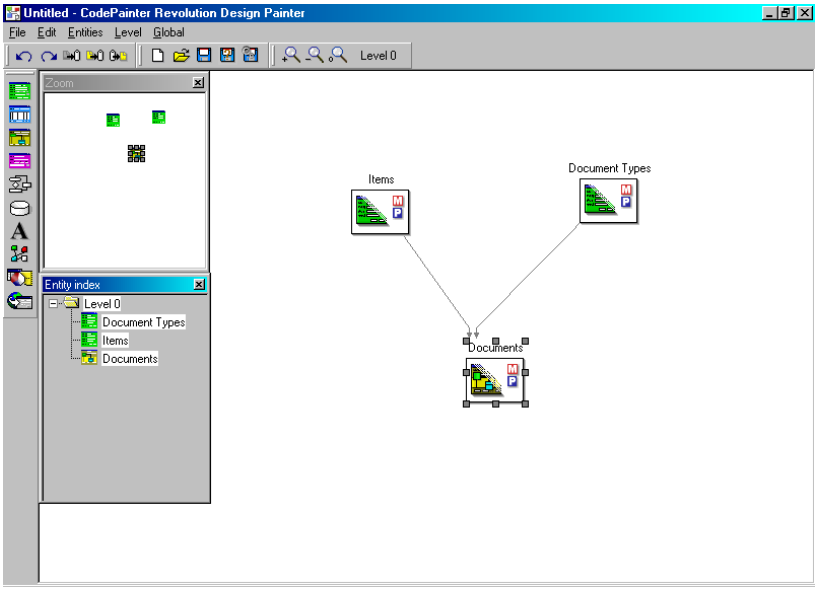
Double click the entity. A window opens up. Write "Items" in the "Name" field, and confirm clicking 'OK'.

Follow the same steps to open a second Master Entity that you will name 'Document Types' and a Master/Detail Entity (the third, yellow-brownish icon) that you will name 'Documents'.

Picture 5 - Three entities in the Design Plan



You now need to define how these entities interact with each other, e.g. 'Documents' receive data from both 'Items' and 'Document Types'. Right click 'Documents' to open the drop-down menu. Choose Link and straight after Relationship. The mouse changes into a plus with the description 'LINK'. Go to 'Items' and click again. Create the same link between 'Documents' and 'Document Types'.



Picture 6 - Creating Links

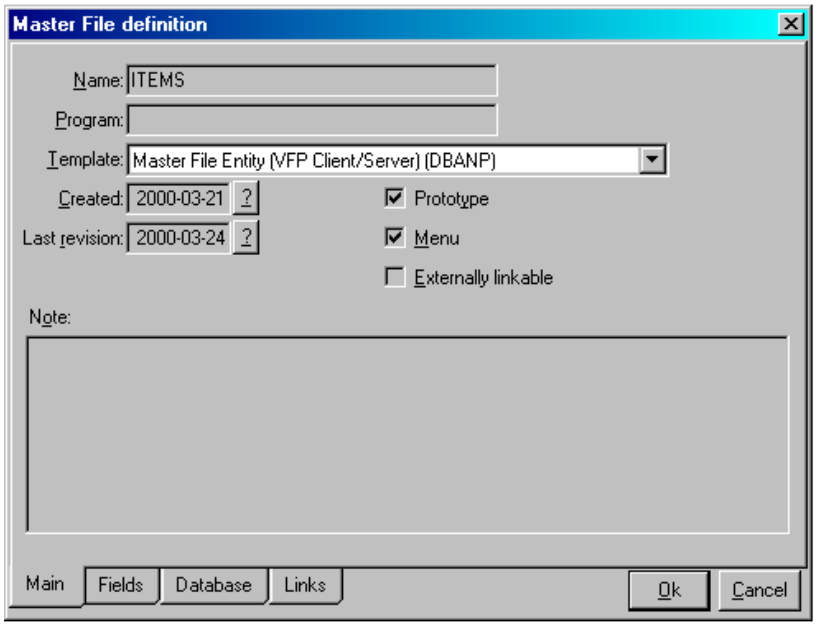
You can now pass on to the entities definition.

2.4.1 Defining Entities

You now need to define each entity, i.e. define database structure, index formulas, and other parameters, which are essential for prototyping.

Items

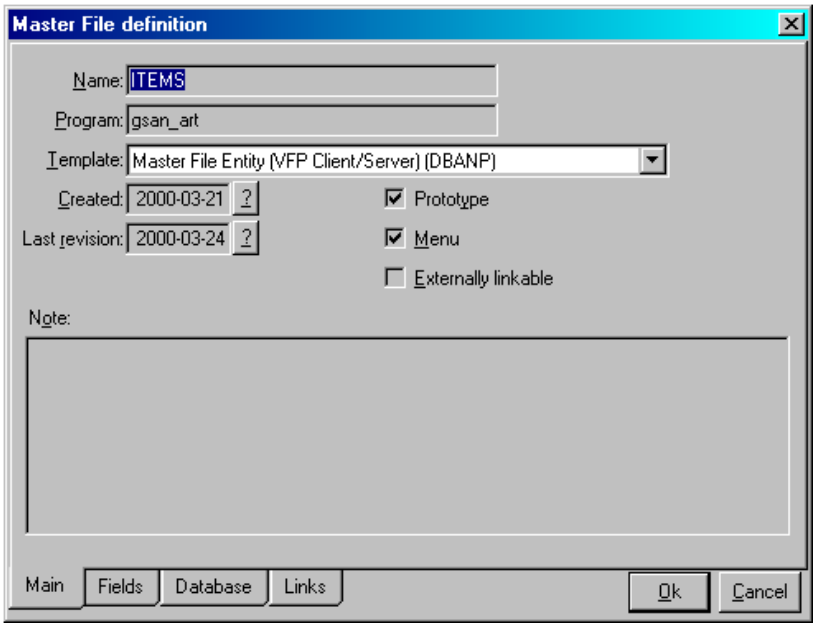
Double click the 'Item' entity to open the definition dialog window.



Picture 7 - Definition Window of the Item Master File

The fields 'Creation Date' and 'Last Revision Date', the flags 'Prototype' and 'Menu' are defaulted. You need to define the managing procedure name. Go to the 'Program' field and digit a string, in this example 'gsan_art'.

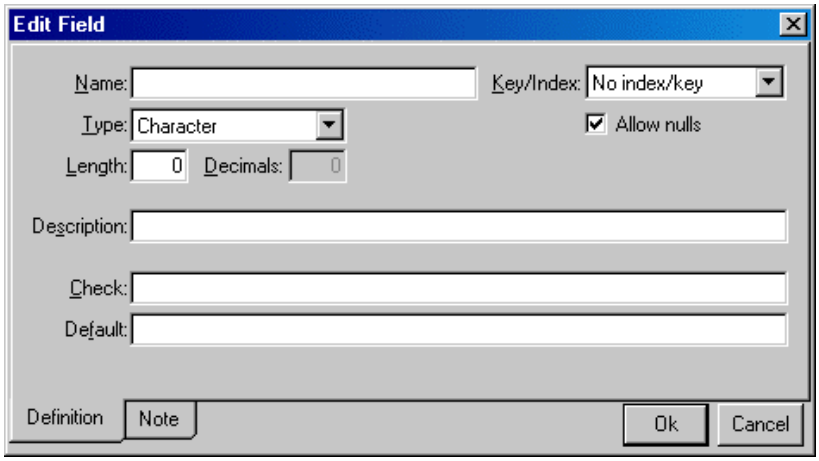
Click the combo-box next to the 'Template' field and select the first template displayed 'Master File Entity (VFP Client Server)(DBANP)'. This template is the skeleton used to automatically generate the source code for this entity.



Picture 8 - Defining General Parameters

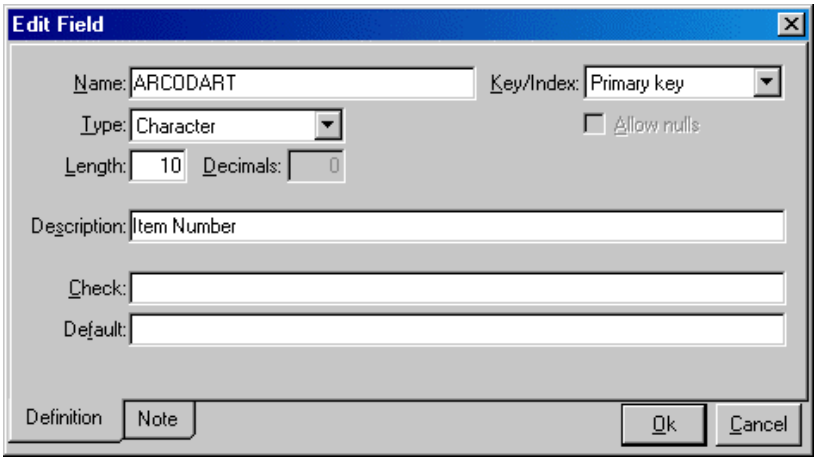
Go to the second dialog window sheet ('Fields'), to define the database structure.

Click the "+" button to open the windows in which you will define single field parameters.



Picture 9 - Field Definition Window

First of all define the primary key: write 'ARCODART' on the 'Name' field. Go to the field with the description 'No Key' and select 'Primary Key'. This field is now part of the primary key. Go to the 'Type' field and select 'Character'. Go to the 'Length' field and write 10, meaning that the field will accept up to ten characters. Go to the 'Description' field and write the description 'Item Number'. Click 'OK' to confirm your entries.



Picture 10 - Complete Field Definition Window

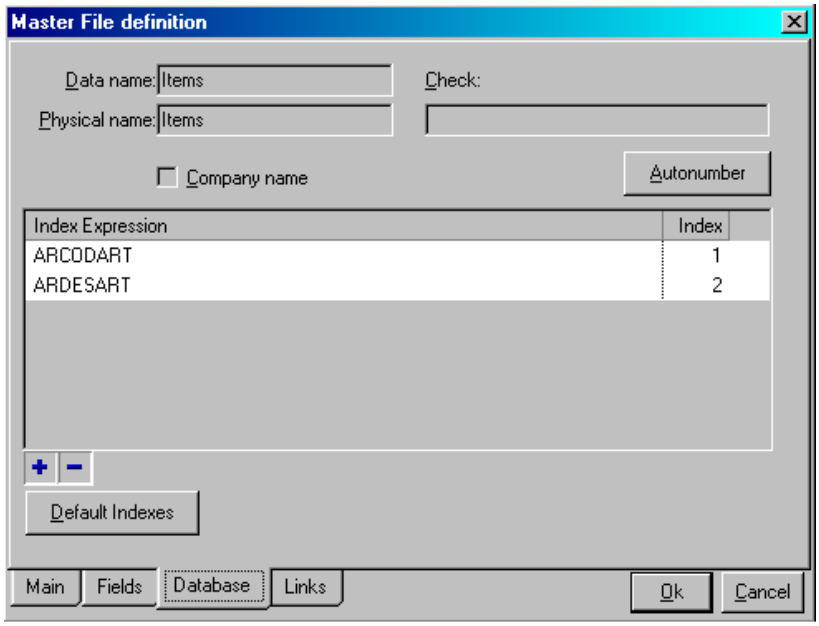
Follow the same steps to add the fields detailed in the table below.

Name	Type	Len	Dec	Key	Descr.
ARCODART	C	10		Prim.	Item No.
ARDESART	C	30		Sec.	Descr.
ARPRZART	N	10	2	No	Price
ARACQART	N	10		No	Purchased Q.
ARVENART	N	10		No	Sold Q.
ARORDART	N	10		No	Ordered Q.
ARIMPART	N	10		No	Booked Q.

Go to the third window sheet ('Database') to define the database name and key structure.

On the 'Data Name' field write the string 'Items'. Press <TAB>, the next field 'Physical Name' will be automatically defined. The Data Name is the name used by the procedure to define the database. The Physical Name is the name used to define the actual file. Click 'Default Indexes' to define default research criteria.

Picture 11 - Database Window Sheet



Click 'OK' to confirm your entries. The 'Item' definition dialog window closes down.

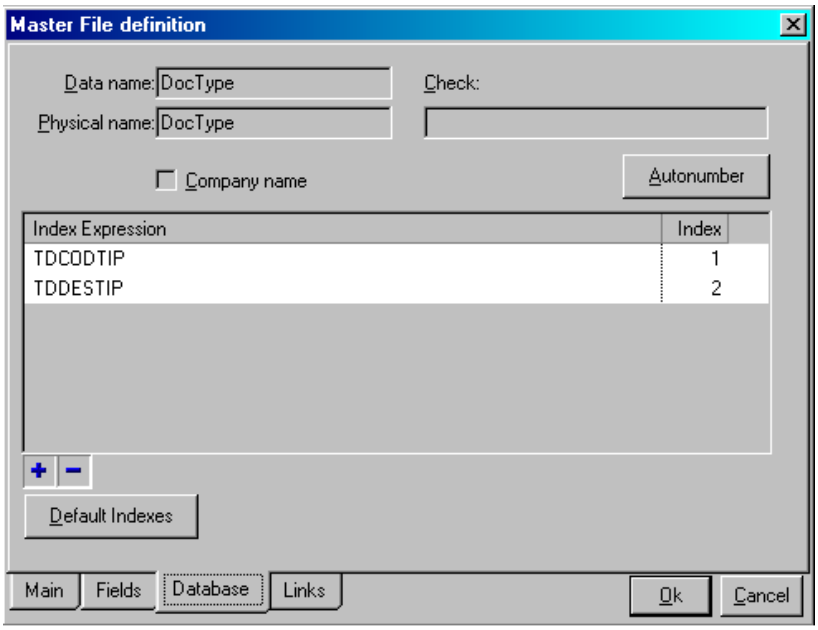
Document Types

Double click the 'Document Types' entity. Write "gsan_tip" in the field 'Program'. Click the combo-box next to the 'Template' field and select 'Master File Entity (VFP Client Server)(DBANP)'.

Go to the 'Field' sheet and define the following fields:

Name	Type	Len	Dec	Key	Descr.
TDCODTIP	C	2		Prim.	Doc. Type No.
TDDESTIP	C	30		Sec.	Descr.
TDACQTIP	C	1		No	Purchased F.
TDVENTIP	C	1		No	Sold Flag
TDORDTIP	C	1		No	Ordered F.
TDIMPTIP	C	1		No	Booked F.
TDFORTIP	C	1		No	Suppl. Y/N

Go to the 'Database' sheet and write 'Doc. Type' in the 'Data' and 'Physical Name'.



Picture 12 - Defining The Document Type Database Sheet

Documents

Double click the 'Documents' entity. Write "gsmd_doc" in the field 'Program'. Click the combo-box next to the 'Template' field and select Master/Detail Entity (VFP Client Server)(DBANP).

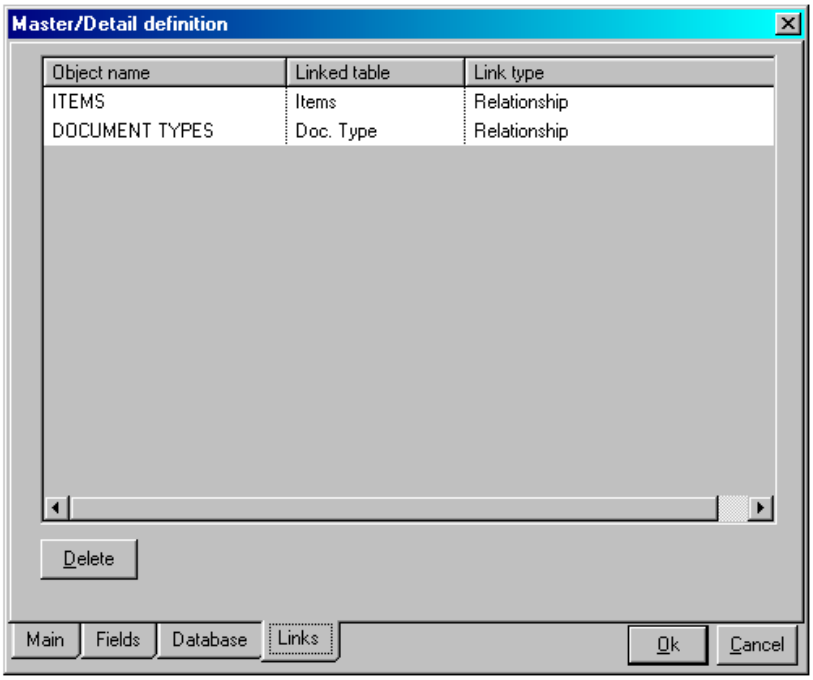
Go to the 'Fields' sheet and define the fields shown in the table below. You also have to activate the 'Repeated' flag for repeated fields.

Name	Type	Len	Dec	Rep.	Key	Descr.
DCCODTIP	C	2			Prim.	Doc. Type No.
DCCODDOC	C	10			Prim.	Doc. No.
DCDATDOC	D	8			No	Date
CPROWNUM	N	6		R	Prim.	Line No.
DCARTDOC	C	10		R	No	Item
DCPRZART	N	10	2	R	No	Price
DCQTADOC	N	4		R	No	Quantity
DCACQOPE	C	1			No	Purchased
DCVENOPE	C	1			No	Sold
DCORDOPE	C	1			No	Ordered
DCIMPOPE	C	1			No	Booked

N.B.

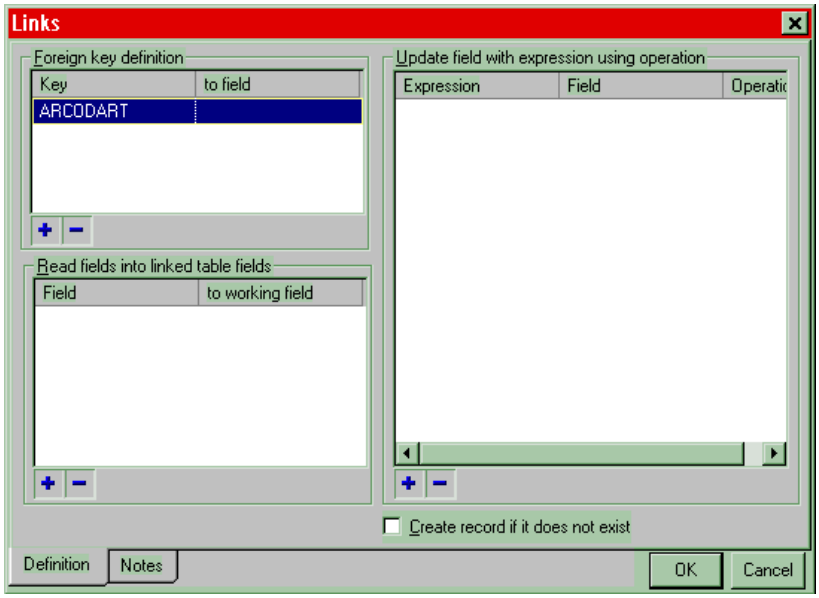
'CPROWNUM' is a system generated field and is usually defined as primary key, so that each line of Detail or Master/Detail files can be identified as unique.

Go to the 'Database' sheet and write 'Document' in the field 'Data Name'. Press <TAB> and all other fields are automatically entered. Click 'Default Indexes' to define default research criteria.



Picture 13 - Active Links Window

Go to the 'Links' sheet and click the first line (Items). In the new dialog window you can define the link between the two entities.

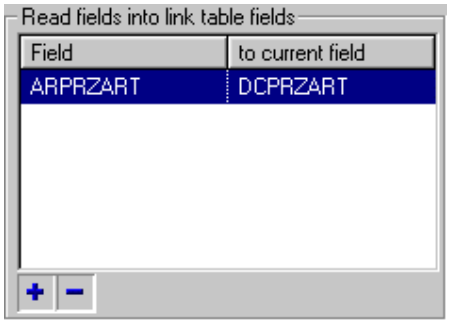


Picture 14 - Link Definition for the 'Item' File.

The links in this example will work as follows: item code and price are taken from the 'Item' file. When a document is saved or cancelled the quantities defined on the body's lines must up-date the fields 'Purchased Quantity', 'Sold Quantity', 'Ordered Quantity', and 'Booked Quantity' of the 'Item' file. Quantity up-dates will be made basing on the mathematical signs defined in the 'DCxxxOPE' fields of the 'Item' file. This is the reason why the field 'ARCODART', the item number, has been set as the item file's primary key.

Go back to the Link sheet, double click the 'ARCODART' field, so that you can establish the link with the corresponding field. Click the right '?' button, in order to obtain the field list of the current file. Select the field double clicking 'DCARTDOC'. Click outside the dialog window area to confirm your selection.

Go to the 'Read Fields into linked table fields' area and click the '+' button to insert a new line. Click the left '?' button to select the field 'ARPRZART" from the 'Item' file. Then click the right '?' button to select the field 'DCPRZART".

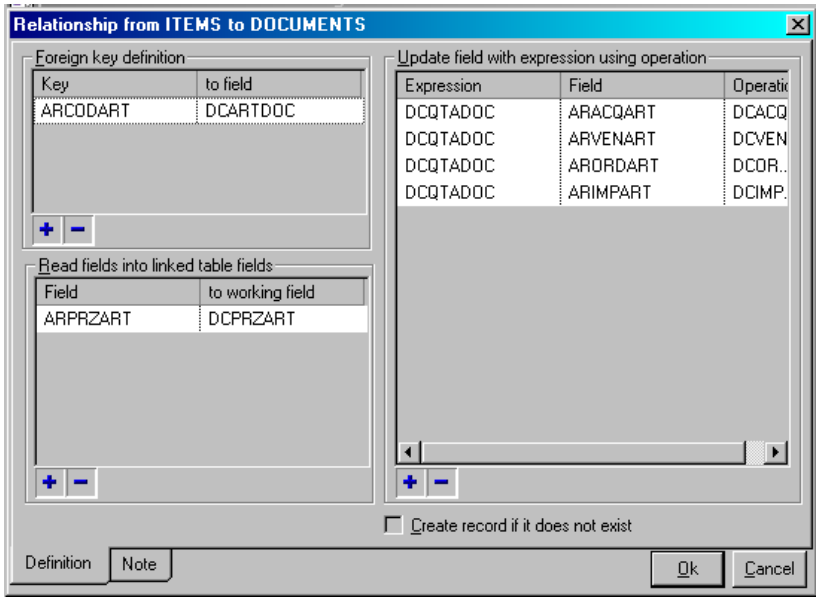


Picture 15 - Price Decoding

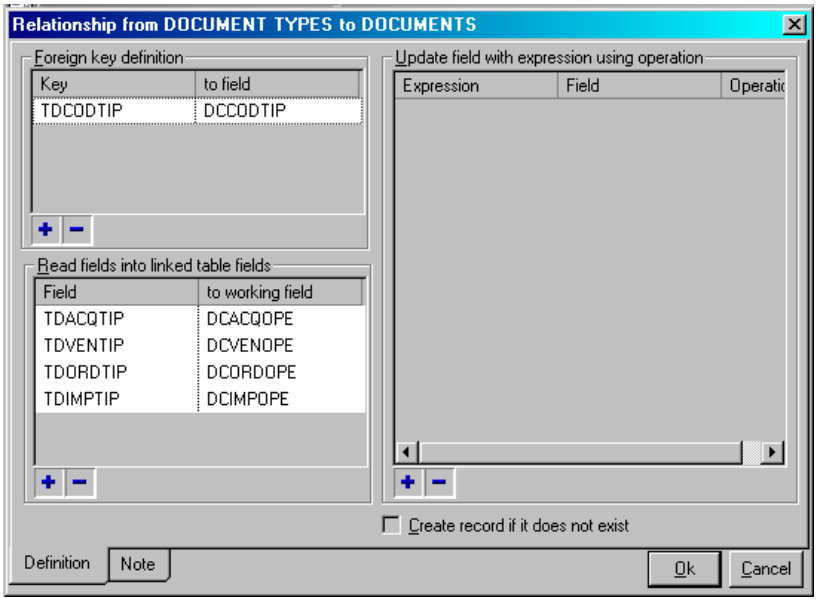
Go to the 'Update field with expression using operation' area to define the field names that will update the 'Item' file. Click the '+' button to insert a new line. Click the left '?' button and select the field 'DCQTADOC'. Click the middle '?' button and select the field 'ARACQART". Click the right '?' button and select the field 'DCACQOPE', that will contain the mathematical sign (+, -, = or <blank> as neutral sign). Repeat these steps to obtain the same result as show in the table below.

Update Field	Expression	Operation
DCQTADOC	ARACQART	DCACQOPE
DCQTADOC	ARVENART	DCVENOPE
DCQTADOC	ARORDART	DCORDOPE
DCQTADOC	ARIMPART	DCIMPOPE

Picture 16 - Completed Link Definition Window



Click 'OK' to confirm your entries. You have come back to the links window. Now select 'Document Types' and define the following: in the 'Foreign key definition' define the fields "TDCODTIP - DCCODTIP', in the 'Read Field into linked table Fields' the fields "TDACQTIP - DCACQOPE', "TDVENTIP - DCVENOPE', 'TDORDTIP - DCORDOPE', and "TDIMPTIP - DCIMPOPE'.



Picture 17 - Completed Link Window for The Documents File

Confirm all dialog windows clicking 'OK' until you come back to the plan.

Open the 'File' menu, select 'Save' and name it 'plan'.

Open the 'File' menu again and select 'Exit' to go back to the Front End.

2.4.2 The Application Prototype

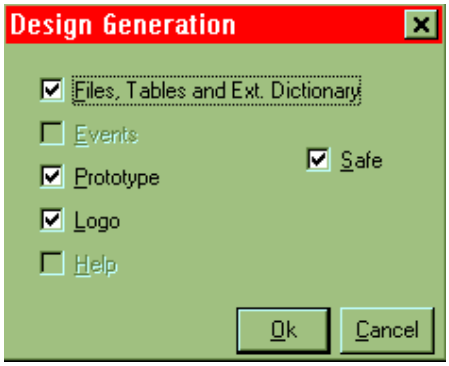
A prototype is a rough procedure, with barely any control, with the exception of link files. Prototypes can help you verifying whether your project is working and is complaint with requirements. You will improve the Software until you obtain the optimal prototype.

Open the 'Generation' menu and select 'Build Application'. This functionality will guide you through the procedure generation.

Design Generation

In the first screen you can create menus, files and templates, the software logo and the procedure's prototypes, i.e. files with the extension .xxxDEF

Picture 18 - Design Generation Mask



The 'Safe' flag allows you to regenerate prototypes for changed entities only, ignoring the ones you have previously consolidated.

Select the following flags: 'Files, Tables and Ext. Dictionary', 'Prototype' and 'Logo'. Click 'OK' to confirm.

The procedure prototype is written in the definition file (.xxxDEF), which is compatible with the 'Codify Painter' that uses standard windows and does not require controls.

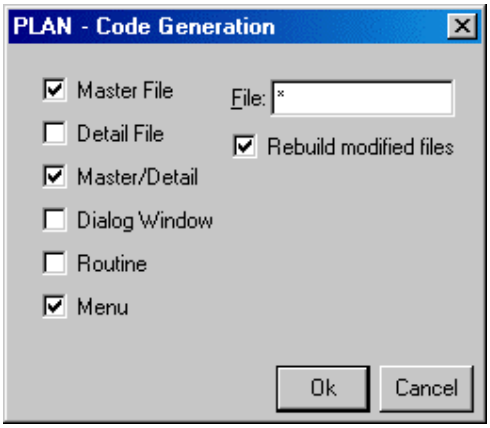
The Master File entity generation, creates a dialog window containing fields and field descriptions defined at design level. Depending on total field number, fields will be displayed on one or more pages. Master/Detail and Detail entities have Header, Body and Footer, and therefore must be generated using the following criteria. Unrepeated fields ('REPEATED flag [ ]' inactive) become part of the HEADER, maintaining the defined sequence. Repeated fields ('REPEATED flag [*]' active) become part of the BODY, maintaining the defined sequence. Depending on field length and total number, one or more horizontal lines are created. Remaining unrepeated fields become part of the FOOTER.

System generated .xxxDEF files can be maintained and completed using coding painters. To change a Master File entity you need to launch the Master File Painter and open the corresponding definition file.

Be careful when you re-generate prototypes. If you want to maintain changes and improvements for certain entities, you need to deactivate the 'Prototype' flag in each entity's definition dialog window.

Code Generation

Once the design generation is terminated, the 'Code Generation' window is displayed. This window creates the actual program code, basing on the definition files you have just created.



Picture 19 - Code Generation Window

A series of flags allow you to select entity classes for which you want to generate the code. The 'Rebuild Modified Files' option allows selecting entities amongst selected classes, which have been modified since the last code generation.

In this exercise, select 'Master File', 'Master/Detail', and 'Menu', then click 'OK' to confirm.

Once the Code Generation is terminated, the 'Documentation Generation' window is displayed. This topic will be addressed in later chapters, therefore simply click 'Cancel'.

Documentation Generation

The 'Documentation Generation is the last dialog window for building your application.

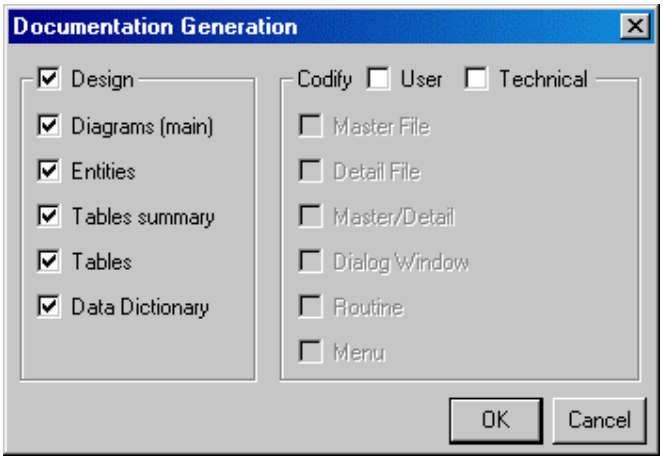


Figura 1 - The Documentation Generation Dialog Window

Most flags are selected by default. You only need to select the 'Design' to generate the documentation for all project elements. Click 'OK' to confirm.

The project is still in an initial phase and thus the only documentation that can be generated is the Technical one.

Producing documentation for SW applications is one of the main required effort in terms of time and resources. Once the documentation has been produced you constantly need to keep it up to date and in line with the evolving product.

The documentation is produced during the Prototype phase so that it can be used to discuss the product with the client, or as a communication tool between analyst and programmer, or as a basis for user reference guides.

Once the Documentation generation is terminated you can check the created .HTML documents opening the 'Project' menu and selecting 'View Documentation' and then 'Design'.

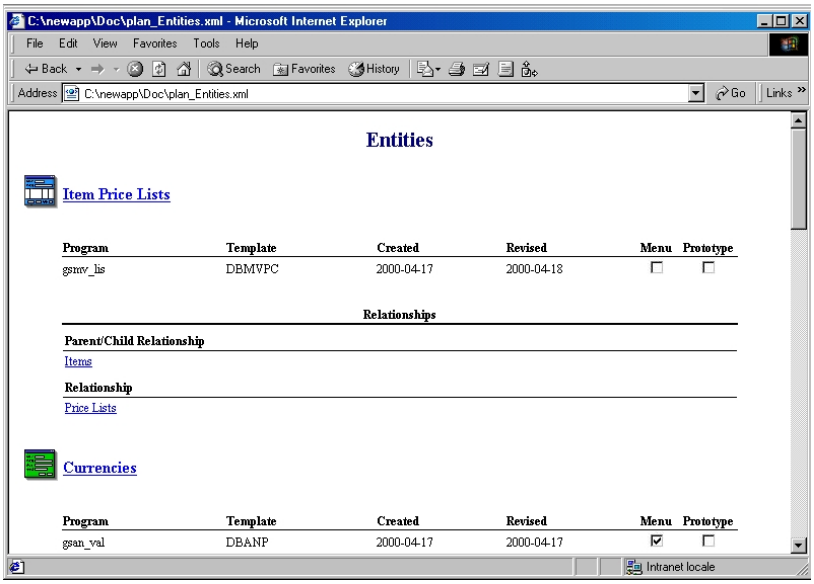


Figura 2 - Design Documentation - Extract

Capitolo 3

Prototype Audit

3.1 Introduction

The prototype you have created must now be audited. The prototype audit allows highlighting structural or functional errors. Further, functionalities can be changed or added. Using the coding tool and changing the design when and where required you will obtain the optimal prototype and thus the final Software.

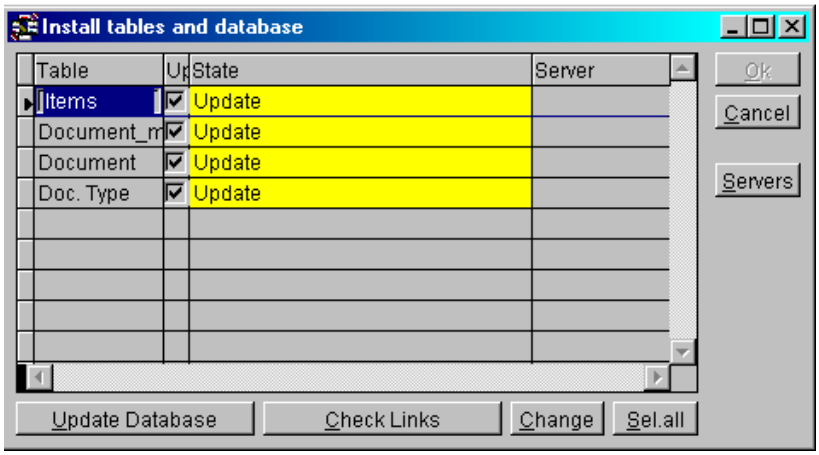
This chapter guides you through a prototype audit using MS Visual FoxPro.

3.2 Prototype Audit using MS Visual FoxPro

Go to CodePainter Front-End, open the 'Project' menu and select 'Run'; otherwise start Microsoft Visual FoxPro, open the 'Program' menu and select 'Do'. In the dialog window open the directory in which you saved your plan and select the program name 'CP3START.PRG'.

You have launched the interpreted execution of the prototype. The procedure starts displaying the Logo window containing the following information: procedure's title (default .PROG), author, commissioner, version (that you can define in Design selecting 'Definition' from the 'Global' menu), and a dialog window, in which the database installation must be defined.

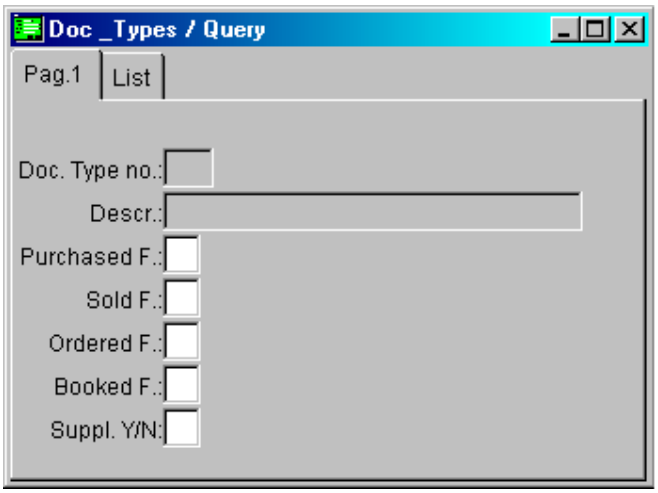
Picture 20 - Database Update Window



Click 'Update Database' to automatically generate the database in SQL language. Two dialog windows will follow. Confirm them clicking 'OK' or pressing <Enter>. The Software's main menu has been prototyped basing on the entities you have defined in the project plan. There are a toolbar and an applicationbar, which are named after the entities. You now need to enter data in the databases 'Document Types' and 'Items'.

'Document Types' Audit

Open the 'Document Types' menu. The corresponding procedure is activated.



Picture 21 - 'Document Types' Window

Select the 'Create' icon or press <F4> to insert new records starting from the first window field.

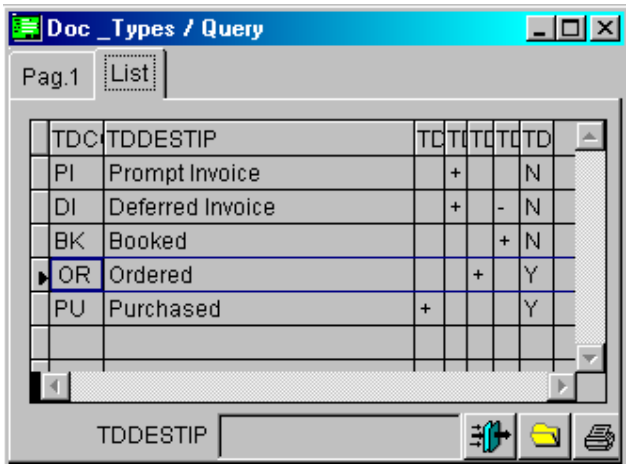
Follow the table below and enter document type codes, descriptions, and mathematical signs reflecting warehouse transactions. After having entered a record save it clicking 'Save' or pressing <F10>.

Code	Description	Purchase F.	Sold Flag	Ordered F.	Booked F.	Supplier F.
PI	Prompt Invoice		+			N
DI	Deferred Invoice		+		-	N
BK	Booked				+	N
OR	Ordered			+		Y
PU	Purchased	+				Y

Once you have completed data entry, press <Esc> to come back to display functionality.

The cursor defaults on the field 'Document Type Code'. If you click 'Zoom', or press <F9>, or if you select the window sheet 'List', the complete record list is displayed.

Picture 22 - Record List of The 'Document Type' Table



Select a record using <Arrows> and pressing <Enter>, or clicking on it directly. The selected record is entered in the previous window.

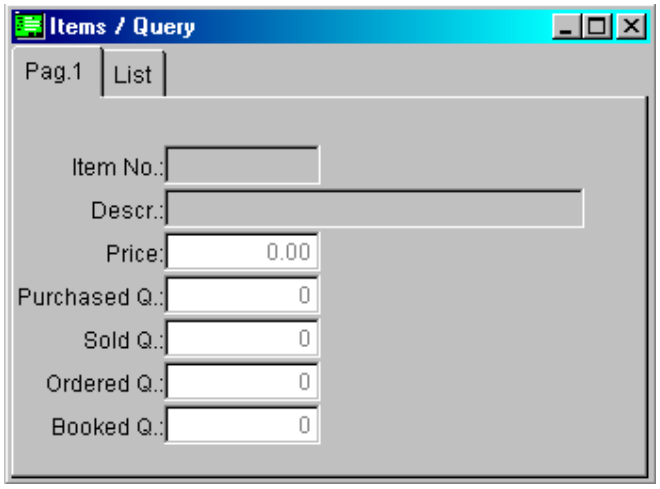
The record can be modified clicking

'Change', or pressing <F3>. The record can also be deleted clicking

'Delete', or pressing <F5>. Select <Esc> to go back to the main menu.

'Items' Audit

Open the 'Items' menu. A dialog window is opened:.



Picture 23 - 'Item' Dialog Window

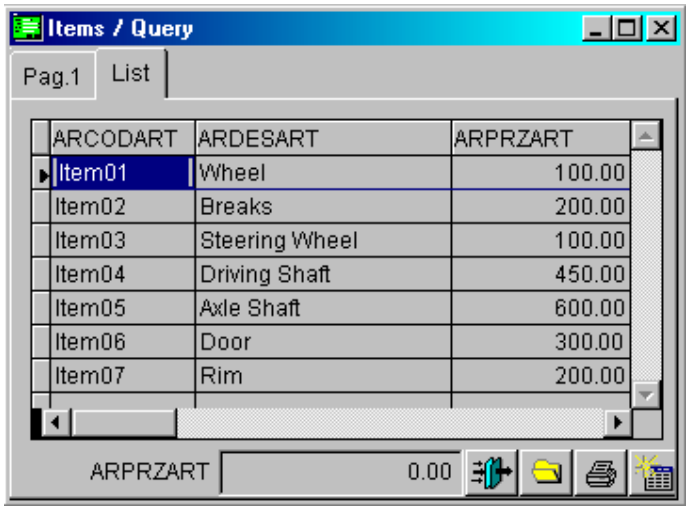
Select 'Create' or press <F4> to enter the 'items' detailed in the table below.

Code	Description	Price
Item01	Wheel	100
Item02	Breaks	200
Item03	Steering Wheel	100
Item04	Driving Shaft	450
Item05	Axle Shaft	600
Item06	Door	300
Item07	Rim	200

After each record has been entered, click 'Save', or press <F10>.

Once you have completed data entry, press <Esc>.

The cursor defaults on the field 'Item Number'. If you click 'Zoom', or press <F9> or if you select the window sheet 'List', the complete record list is displayed.



Picture 24 - Zoom on the 'Item' File

To change the sort code double click the column name. Clicking again you can choose whether to sort data in ascending or descending order.

To find the item 'Door' double click the column 'ARDESART', write the string 'Door' in the field at the bottom of the window, and press <Enter>. All records having the 'Code Description' starting with 'Door' are displayed. Select the desired record and press <Enter>. The selected record is entered in the previous window.

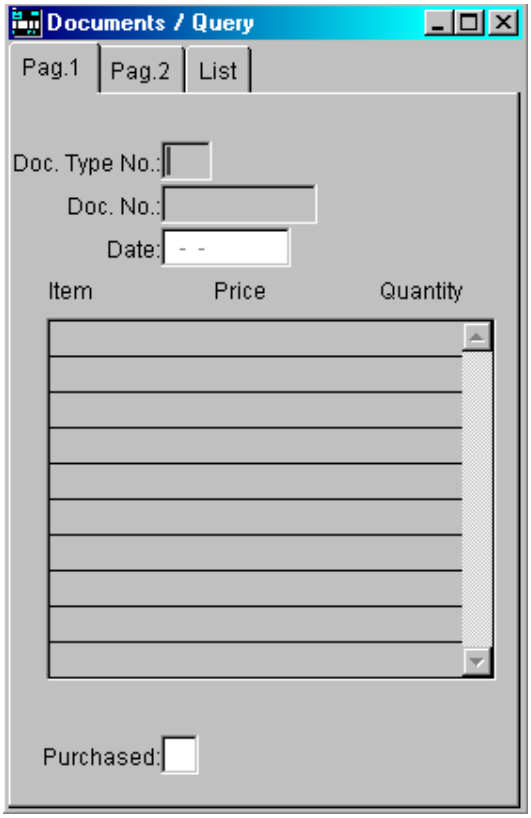
You could also search for an item using the 'Item Number'. Activate the 'Display' mode and position the cursor on the primary field next to 'Item Number'.

You can activate the search functionality on editable fields, clicking 'Filter' or pressing <F12>, go to the field 'Price' and digit '100'. Save with <F10>. You can scroll records having the value '100' using <F7> and <F8>. In this exercise you will find 'Wheel' and 'Steering Wheel'.

Click <Esc> to come back to the main menu.

'Documents' Audit

Now you will audit the Master/Detail Entity. Select 'Documents' from the menu. A dialog window is opened.



Click 'Create' or press <F4>.

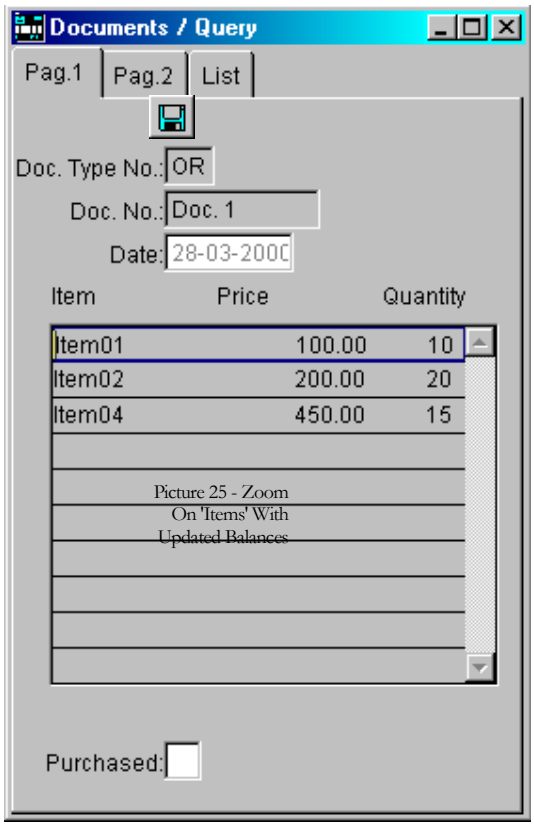
Remember: in the design phase you linked the entity 'Document Types' with the field 'Doc. Type', in order to read data from the 'Document Types' table.

Select the first field clicking 'Zoom', or double clicking it or pressing <F9>. A dialog window is opened. Select 'Ordered'. In the 'Document Number' field enter 'Doc. 1' and a date in the field 'Date'. Enter the value 'Trial' in the 'Item' column and press <Enter>. An error message appears: 'Value not correct'. This happens, because in the design phase you linked 'Documents' with 'Items', creating a validation on input data. You basically have defined a referential integrity.

Press <Enter> to by-pass the message. Press <F9> or click 'Zoom' to open the dialog window containing 'Items'. Should the searched item not be available, you could enter it now. Pressing <F9> or clicking 'Zoom' again, you access the 'Item' database. The dialog window is the same you would have accessed using the Items menu and the item can be entered, changed or deleted as usual. Pressing <Esc> you come back to the previous Zoom window. Now enter records as detailed in the table below, so that content is added to the document's body.

Item	Quantity
Item01	10
Item02	20
Item03	13
Item04	15

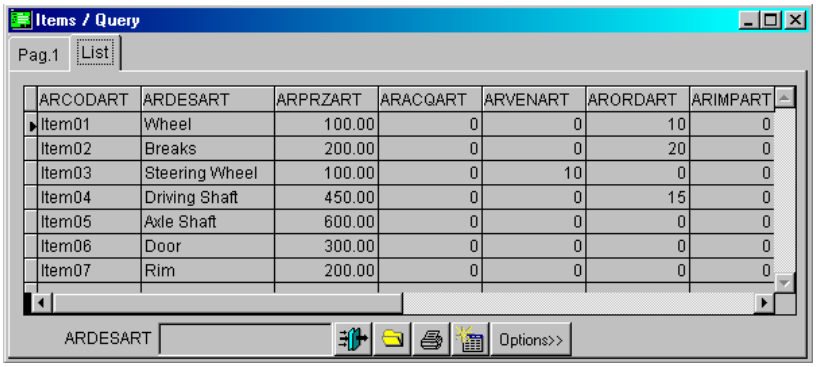
Go to the third record of the document's body and delete it pressing <Del.> or <F6>.



Picture 25 - Zoom On 'Items' With Updated Balances

Save the document clicking 'Save' or pressing <F10>. Enter another record containing the transaction description 'Prompt Invoice', 'Document Type 2', a date, and in the document's body enter 'Item3' with quantity '10'. Save the document clicking 'Save' or pressing <F10>.

Press <Esc> to come back to the main menu. Open the Items menu and select the Zoom functionality pressing <F9>. Notice that item balances are no longer zero, but have been updated according to the documents you have created.



In the design phase you defined how the value in 'DCQTADOC' must update the value 'xxxART' using mathematical signs defined in 'xxxOPE'. This mere description allows the system to automatically perform balance transactions on the database.

Press <Esc>, open the 'Utility' menu and select 'Exit'. You have come back to the Visual FoxPro environment.

Select the active CodePainter session.

You have completed the Prototype Audit! You have seen that it works, and that it is coherent with the original plan. You have also seen the functionalities and the layout. You could improve your software application adding a database for VAT codes, or adding a client database, which would allow adding a default currency and payment methods on documents. Further you could define a price-list database and, using a Detail File, link it to the 'item' database. To do this you need to go back to the 'Design Painter', change the Project Plan and go through the prototyping phase once again. Please refer to chapter 4.

Capitolo 4

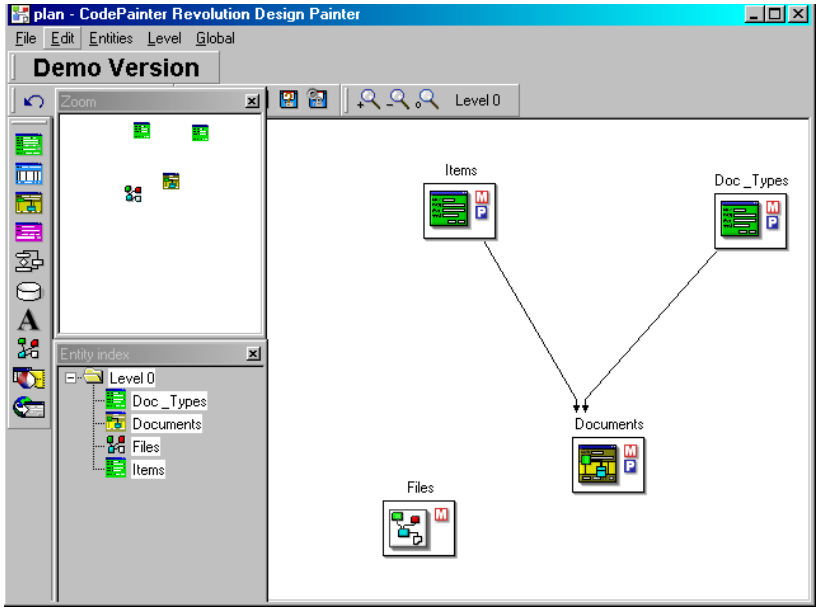
Changing The Project Plan

Open the 'Painters' menu and select 'Design'. Open the 'File' menu, select 'Open' and choose the project plan you have created (PLAN.DEF).

4.1 Changes to The Project Plan

For a better plan overview open the 'Entities' menu, select

'Group' and place the entity on your plan. Right click it and select 'Edit' to open the definition dialog window. Write the string 'Files' next to the field 'Name' and click 'OK'.



Picture 26 - Adding a Group Entity on The Project Plan

Position the mouse on the 'Document Types' entity, right click it and select 'Group to' to move it in the 'Files' entity. A new dialog window is opened displaying the project structure. Click the icon representing the group 'Files' and click 'OK'.

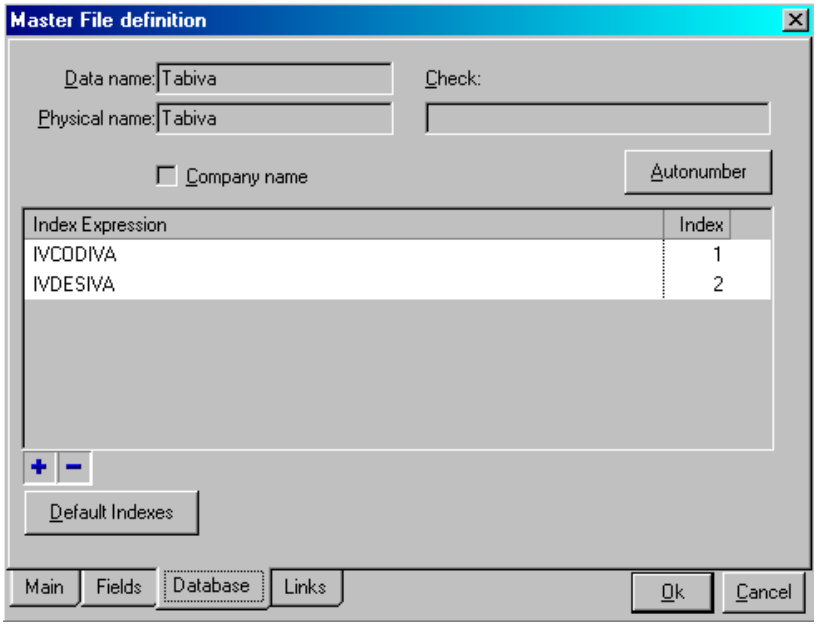
4.1.1 Value Added Tax Rates

Edit the group entity and add a new Master File Entity. Double click the Master entity to open the definition dialog window. Write 'VAT-Rates' next to the field 'Name'. Write 'gsan_iva' next to the field 'program'. Click 'Template' and select the 'Master File Entity (VFP Client/Server)(DBANP)'.

Go to the 'Fields' sheet and build the database structure basing on the table defined below:

Name	Type	Len	Dec	Key	Description
IVCODIVA	C	2		1	VAT Code
IVDESIVA	C	30		2	Description
IVPERIVA	N	2	0		VAT %

Go to the 'Database' sheet and write 'TABIVA' next to the 'Data name' field. By pressing <TAB> the 'Physical Name' field is automatically entered. Click 'Default Indexes' and confirm with 'OK'.



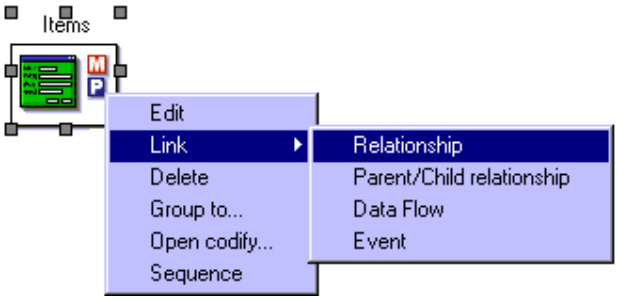
Picture 27 - The Database Sheet

4.1.2 Changing the 'Item' Entity

You now need to add the field 'VAT %' in the 'Items' entity and to link 'Items' to 'VAT Rates'.

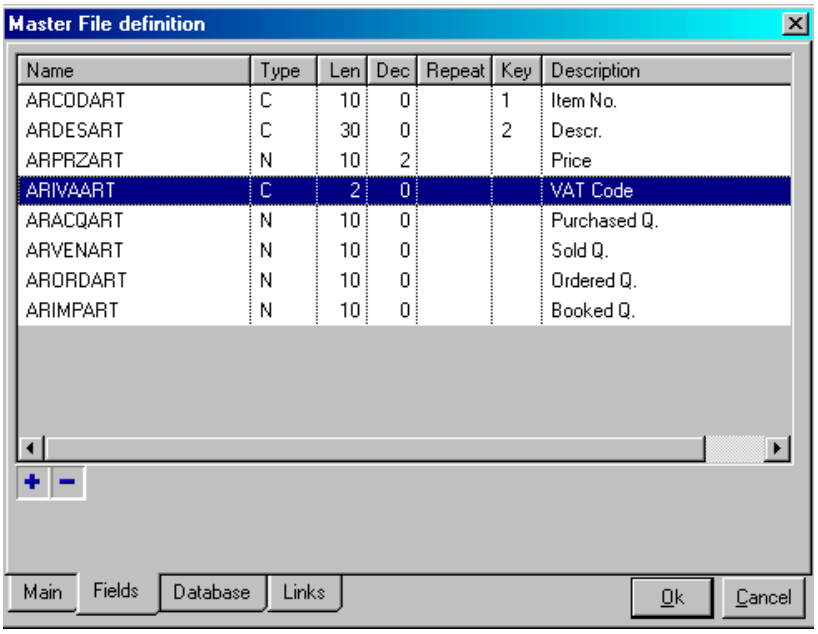
Click this icon to go back to the plan's top level. Right click 'Items', open the 'Link' menu and select 'Relationship'.

Picture 28 - Selecting the Link Type



The cursor has changed into '+'. Click the group 'File' and then 'VAT Rates' to build the link. Double click 'Items', select the 'Fields' sheet, click the '+' button and digit 'ARIVAART" next to the 'Name' field, select 'Character' type, digit 2 as field lengths and write the description 'VAT Code'.

Move the mouse between the field name 'ARIVAART" and the left window frame to change the cursor into a hand. You can move the field within the dialog window clicking and dragging the field up and/or down. Move the field under 'ARPRZART".



Picture 29 - Added Fields

Select the 'Links' sheet, where build links are displayed. Double click the link to open the definition dialog window. Double click the field 'IVCODIVA' and click the right '?' button. Select the field 'ARIVAART' from the linked entity. Click outside the window and confirm with 'OK'.

4.1.3 Customers File

On the project plan add a new Master File Entity. Open the definition dialog window, write 'Customers' next to the name field, write 'gsan_cli' next to 'program', click the 'Template' line and select the 'Master File Entity (VFP Client/Server)(DBANP)'.

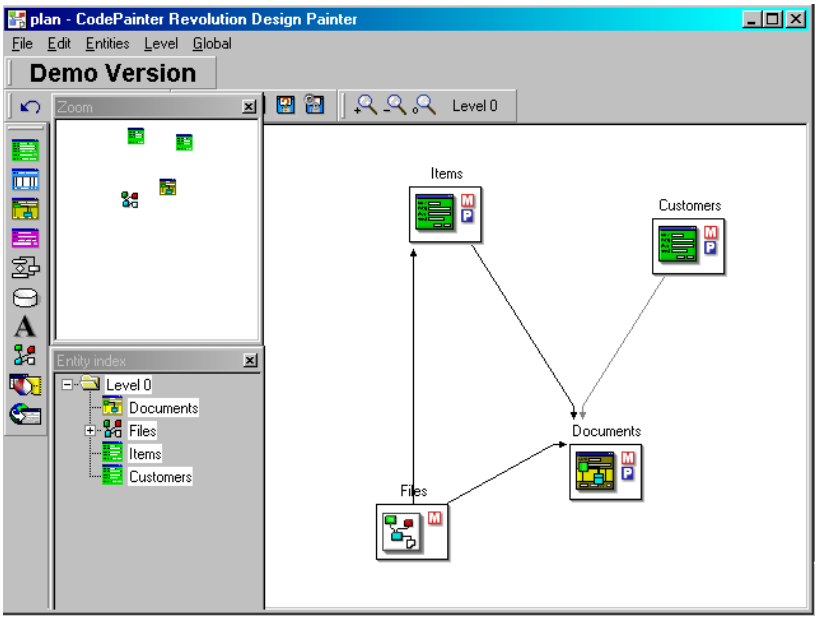
Go to the 'Fields' sheet and add fields as defined in the following table:

Name	Type	Len	Dec	Key	Description
CLCODCLI	C	10		1	Customer Key
CLRAGCLI	C	30		2	Company Name
CLINDCLI	C	50			Address
CLCITCLI	C	20			Town
CLCAPCLI	C	5			Post Code
CLPROCLI	C	2			Country
CLCODPAG	C	3			Payment Method
CLCODVAL	C	3			Currency
CLFLAFOR	C	20			Supplier Flag

Click 'Database', write the string 'customers' next to 'Data Name'. Press <TAB> to enter the 'Physical Name'. Click 'Default Indexes' and confirm your entries clicking 'OK'.

4.1.4 Changes to The 'Documents' Entity

Right click 'Documents', open the 'Links' menu, select 'Relationships', and click 'Customers' to define the link.

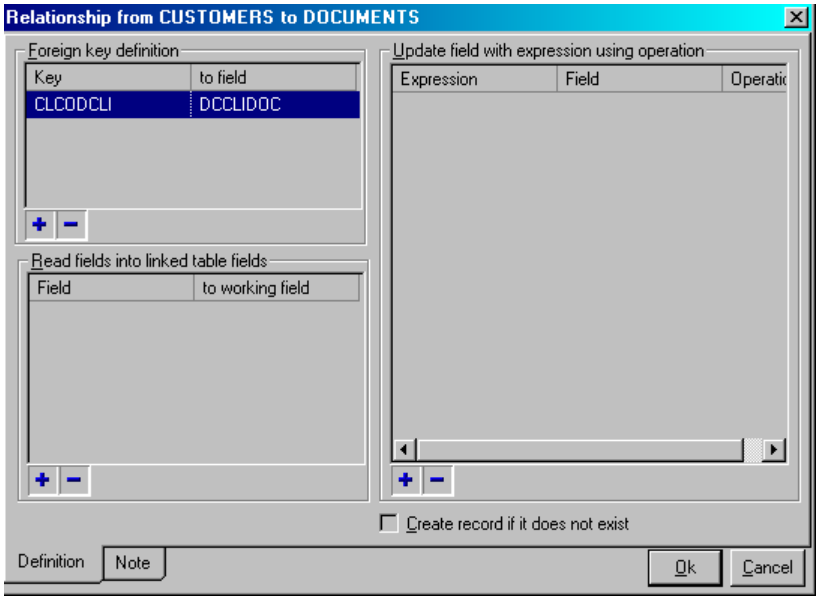


Picture 30 - The New Link Between 'Documents' And 'Customers'

To add the customer key field on 'Documents', double click the entity, go to the 'Fields' sheet and add the field 'DCCLIDOC' as follows:

Name	Type	Len	Dec	Key	Description
DCCLIDOC	C	10		1	Customer Key

Move the added field under 'DCDATDOC' and go to the 'Links' sheet. Select the link with 'Customers' to activate the definition dialog window. Double click the first 'Foreign Key Definition' line. The field 'CLCODCLI' is defaulted. Click the right '?' button and select 'DCCLIDOC'.



Picture 31 - Link Between 'Documents' and 'Customers'

You have just entered a new Primary Field. Go to the 'Database' sheet, and update 'Default Indexes' clicking the button. Confirm your entries to go back to the project plan.

4.1.5 Currencies File

You now need to define the entity containing information on currencies customers will use for payments.

Add a new Master entity in the 'Files' group. Write 'Currencies' next to the 'Name', define the 'Program' as 'gsan_val, and select the template 'Master File Entity (VFP Client/Server)(DBANP)'. Go to the 'Fields' sheet and add fields as defined in the following table:

Name	Type	Len	Dec	Key	Description
VACODVAL	C	3		1	Currency Code
VADESVAL	C	10		2	Description
VACMBVAL	N	7	2		Exchange Rate
VASIMVAL	C	1			Symbol

Go to the 'Database' sheet and digit 'Currencies' next to 'Data Name'. Press <TAB> to enter the 'Physical Name'. Click 'Default Indexes' and confirm you entries with 'OK'.

4.1.6 Payments File

You are still in the 'Files' group. Add a new

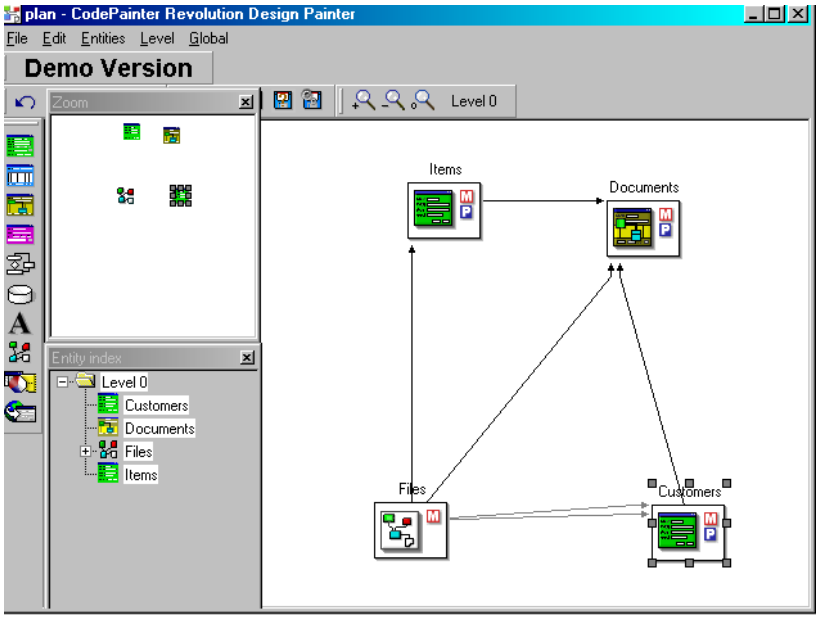
Master File entity, containing information on payment methods. Double click the added entity to open the definition dialog window. Write 'Payments' next to 'Name', define the 'Program' as 'gsan_pag', and select the 'Master File Entity (VFP Client/Server)(DBANP)'. Go to the 'Fields' sheet and add fields as defined in the following table:

Name	Type	Len	Dec	Key	Description
PACODPAG	C	3		1	Payment Code
PADESPAG	C	30		2	Description

Go to the 'Database' sheet, digit 'Payments' next to 'Data Name', press <TAB> to enter the 'Physical Name' and click 'Default Indexes'. Confirm you entries clicking 'OK'.

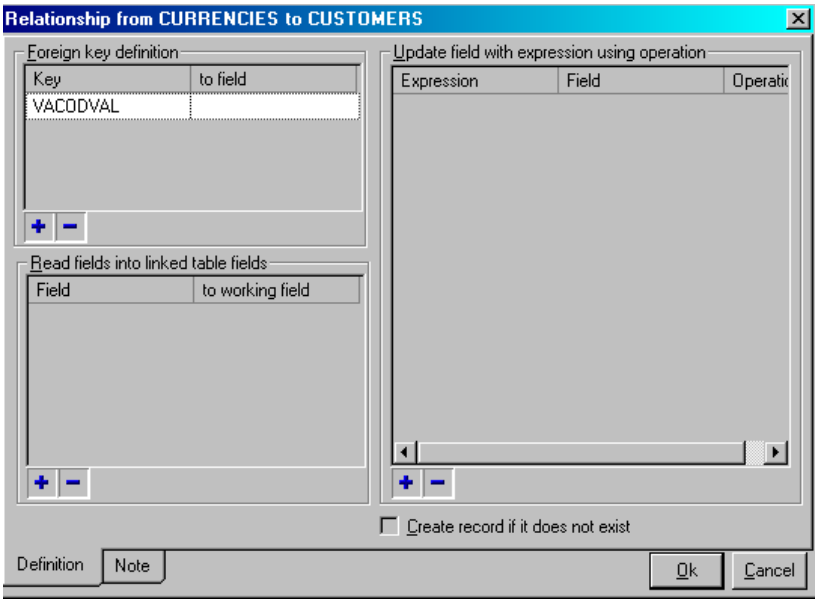
4.1.7 Defining Links for 'Customers'

Open the 'Level' menu and select 'Top Level'. You now need to define links between the 'Customers' file and the files 'Currencies' and 'Payments'. Right click 'Customers', open the 'Link' menu and select 'Relationship'. Click 'Files' first and 'Currencies' straight after. Do the same between 'Files' and 'Payments'.



Picture 32 - Links between the 'Customers' file and the files 'Currencies' and 'Payments'

Double click 'Customers', go to the 'Links' sheet, and select the 'Currencies' link to open the definition dialog window.

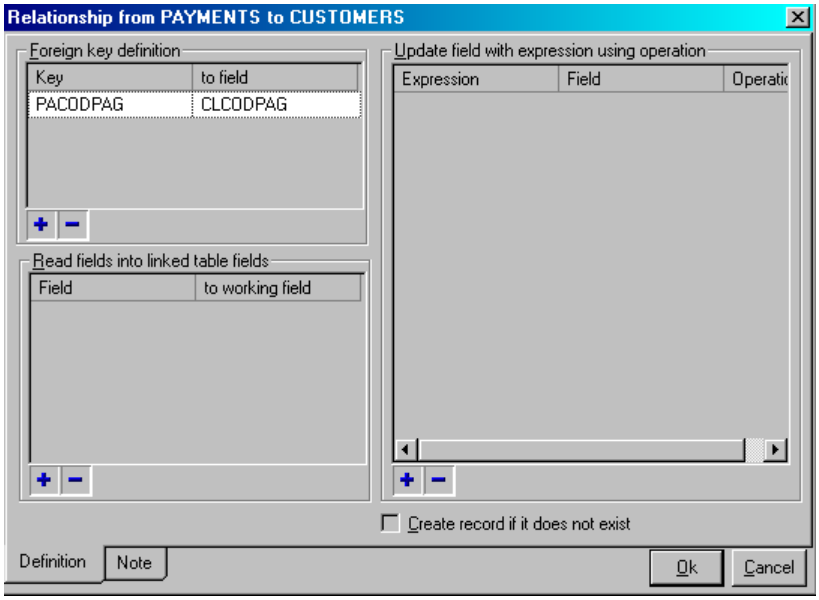


Picture 33 - Defining Currencies Links

Double click the first 'Foreign Key Definition' row. The field 'VACODVAL' is defaulted. Click the right '?' button and select 'CLCODVAL'. Confirm with 'OK'.

Picture 34 - Currency Code Decoding

Select the link with 'Payments' and double click the first 'Foreign Key Definition' row. The field 'PACODPAG' is defaulted. Click the right '?' button and select the field 'CLCODPAG'. Confirm all opened dialog windows clicking 'OK'.



Picture 35 - Defining 'Payments' Link

4.1.8 Price Lists File

Open the 'Files' group and add a new

Master File entity. Double click the added entity to open the definition dialog window. Write 'Price Lists' next to 'Name, define the 'Program' as 'gsan_lis', and select the template 'Master File Entity (VFP Client/Server)(DBANP)'. Go to the 'Fields' sheet and add fields as defined in the following table:

Name	Type	Len	Dec	Key	Description
LICODLIS	C	3		1	Price List Code
LIDESLIS	C	30		2	Description

Go to the 'Database' sheet, write 'PriceLists' next to 'Data Name', press <TAB> to enter the 'Physical Name', click 'Default Indexes', and confirm with 'OK'.

4.1.9 Item Price Lists

Open the 'Level' menu and select 'Top Level'. You now need to create the Detail File 'Item Price Lists', which allows associating an indefinite number of Price Lists to one single item.

Add a 'Detail File' in the plan, double click it to open the definition dialog window. Write 'Item Price Lists' next to 'Name', define the 'Program' as 'gsmv_lis', and select the template **'Child Detail** Entity (VFP Client/Server)(DBMVPC)'.

N. B.
The 'DBMVPC.TPL' template is required to define 'Parent/Child Links'. In this exercise the 'Item Price Lists' detail entity is defined as the 'Child' of the 'Items' entity.

'Item Price Lists' not being part of the menu, deselect the menu flag. Go to the 'Fields' sheet and add the fields defined in the following table:

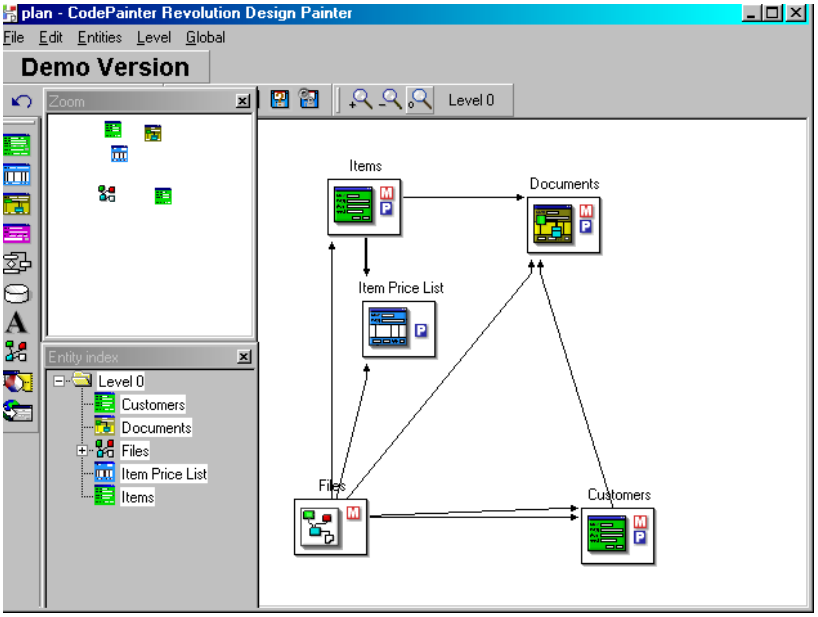
Name	Type	Len	Dec	Rep	Key	Description
LACODART	C	10			1	Item
CPROWNUM	N	6		R	1	Line no.
LACODLIS	C	3		R		Price List Code
LAPRZART	N	10	2	R		Price

Go to the 'Database' sheet, write 'lis_xart' next to 'Data Name', press <TAB> to enter the 'Physical Name', click 'Default Indexes', and confirm with 'OK'.

4.1.10 Defining Links for 'Item Price Lists'

The detail file you have just created must be linked with 'Price Lists' and with 'Items'. Right click 'Item Price Lists', open the 'Link' menu and select 'Relationships'. Click the 'Files' group first and the 'Price Lists' entity straight after.

Right click 'Item Price Lists' and open the 'Link' menu. Select the 'Parent/Child relationship', click the 'Files' group first and the 'Item' entity straight after.



Picture 36 - Links between Items and List Prices

Double click the 'Item Price Lists' to open the definition dialog window. Go to the 'Links' sheet and select the link with 'Price Lists' to open a new dialog window. Double click the first 'Foreign Key Definition' row. The field 'LICODLIS' is defaulted. Click the right '?' button, select 'LACODLIS', and confirm clicking 'OK'.

Picture 37 - Price Lists Decoding

Select the link with 'Items' to open a new definition dialog window. Being a 'Parent/Child' relationship, the 'Link' sheet has only two sections.

Picture 38 - Definition Dialog Window Of A Parent/Child Relationship

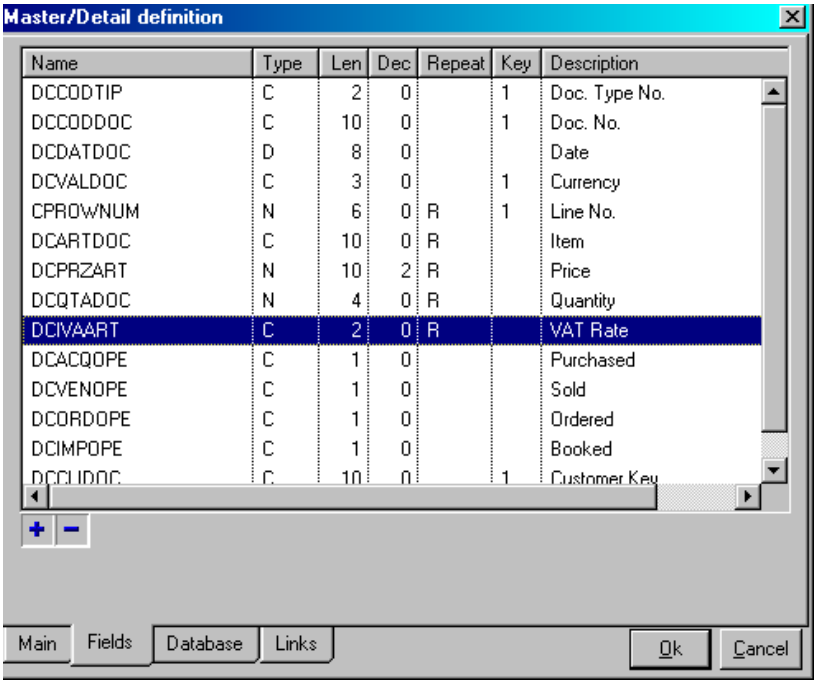
Double click the first 'Foreign Key Definition' row. Click the right '?' button, select 'LACODART", and confirm with 'OK'.

4.1.11 Changes to 'Documents'

You are now required to add in the 'Documents' entity a field for currency codes and one for the VAT rates. Double click 'Documents' and go to the 'Fields' sheet. Add the two fields as defined in the following table:

Name	Type	Len	Dec	Rep	Key	Description
DCVALDOC	C	3			1	Currency
DCIVAART	C	2		R		VAT Rate

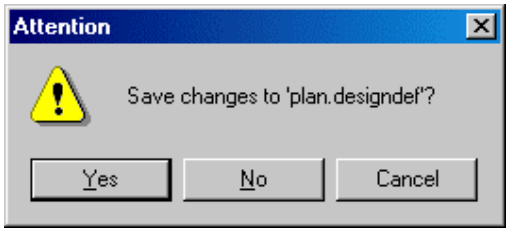
Move the 'Currency' field above 'CPROWNUM', and 'VAT Rate' under 'DCQTADOC', to obtain a logical order.



Picture 39 - Moving Fields

Open the 'File' menu and select 'Exit'. CodePainter will ask you if you want to save the changes. Click 'YES' to save.

Picture 40 - Save Changes Dialog Window



To regenerate your software application, go back to the CodePainter's Front End, open the 'Generation' menu and select 'Build Application'. During regeneration, CodePainter will ask you if you want to regenerate prototypes for entities created in the first phase. So far you have not changed dialog windows, therefore click 'Yes' on each window.

Follow the instructions given at the end of Chapter 2 ('Creating a Software Application') for the 'Codify' phase and for creating technical documentation.

Audit your changed prototype following the instructions given in Chapter 3 ('Prototype Audit'). Once you reached the optimal prototype consider the changes you need to do during the 'Codify' phase to obtain the final software application as shown in chapter 5. These are:

1) 'Item Price Lists' Entity

a) Reduce the window so that when displayed it fits in the parent entity.

b) Change the window header of the detail file, delete the description of the parent item, and hide the field 'LACODART'.

c) Move the body field descriptions.

d) Decode the 'Price Lists' description linking it to the corresponding file.

e) Reduce the window and improve the layout.

2) 'Items' Entity

a) The fields in which quantities are updated must be 'display' only.

b) Decode the 'VAT Rate' description linking it to the corresponding file.

c) Display item quantities.

d) Integrate the child entity 'Item Price Lists' in the parent dialog window.

e) Reduce the window and improve the layout moving fields on more pages.

3) 'Documents' Entity'

a) From 'Document Types' download the value of the 'TDFORTIP' flag, in order to identify whether to extract a 'customer' or a 'supplier' from the 'customers file'. Hide the mathematical sign flags.

b) Activate the 'Autonumber' option to manage progressive autonumbering on the field 'DCCODDOC'.

c) Decode the 'Company Name' and the 'Default Currency' reading from the corresponding file.

d) Decode 'Exchange Rate' and 'Currency Symbol' reading from the corresponding file.

e) Decode 'Description', 'VAT Code', and 'Item Price' reading from the corresponding file.

f) Decode 'VAT Rate' reading from the corresponding file.

g) Create a variable in which the item price can be calculated according to the chosen currency.

h) Insert a variable in which the total price (price times quantity) is input.

i) Insert a hidden variable with the 'Total VAT Amount' on each row.

l) In the Footer insert a 'Total Taxable Amount', a 'Total VAT Amount', and a 'Total Amount'.

m) Improve the windows layout.

Capitolo 5

Finalizing the Prototype

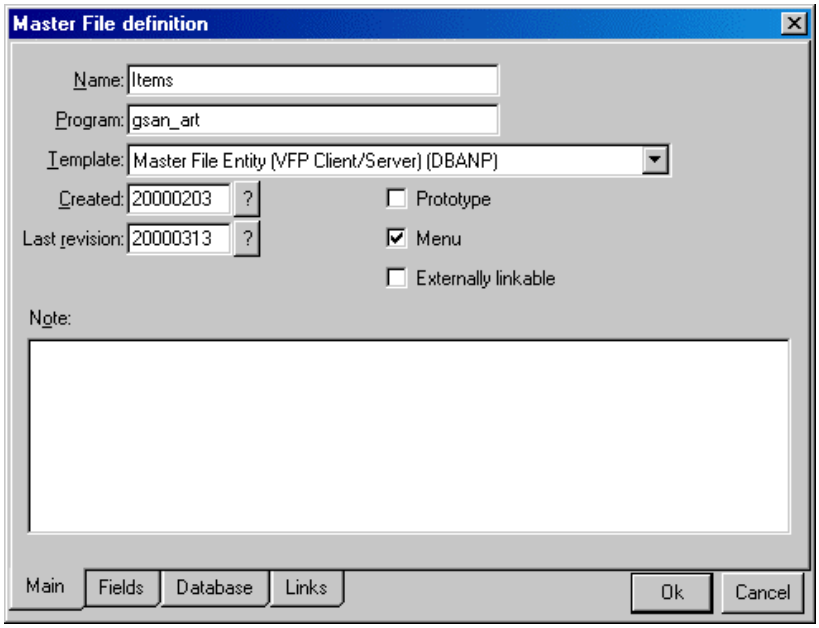
5.1 Introduction

Finalizing the Prototype means performing the 'Codify' phase. Using dedicated painters you can complete your software application, integrating it with all required controls. Further you can improve the layout, e.g.:

- Divide the plan's elements on more pages.
- Highlight key fields and their descriptions using bold Arial font size 9.
- Use line breaks and scrollbars in Detail Files and Master/Detail Files.

First of all you need to deactivate the 'Prototype Flag' from the entities you want to change; otherwise you loose the changes you made so far. Open the Design Painter, edit the definition dialog window of those entities you want to modify, and deselect the 'Prototype' flag.

Picture 41 - Deselecting The Prototype Flag



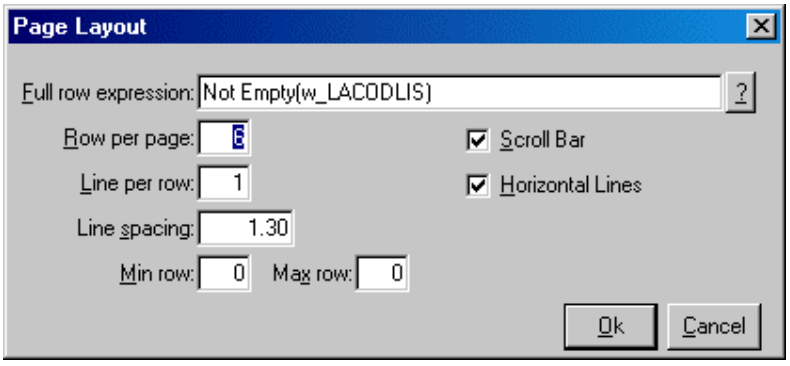
5.1.1 Changing The 'Item List Prices' Entity

Open the 'Painter' menu and select 'Detail' to change the 'Item List Prices' dialog window.

Open the 'File' menu, select 'gsmv_lis' in the dialog window and confirm with 'OK'. You will do the changes discussed at the end of Chapter 4 ('Changing The Project Plan') following the instructions seen in Chapter 3 ('Prototype Audit').

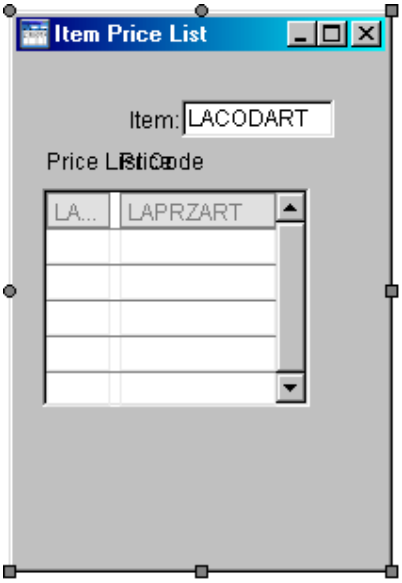
Reducing the dialog window

To define the number of rows to use for editing the body of the Detail File, open the 'Globals' menu and select 'Page Structure'. The page layout definition window is opened. Digit '6' next to 'Row per page' to define the number of rows that can be used for data input.



Picture 42 - 'Page Structure' Dialog Window

Having decreased the number of rows, you have to reduce the window's lengths and widths. Click the window border and drag it by the 'handles'.

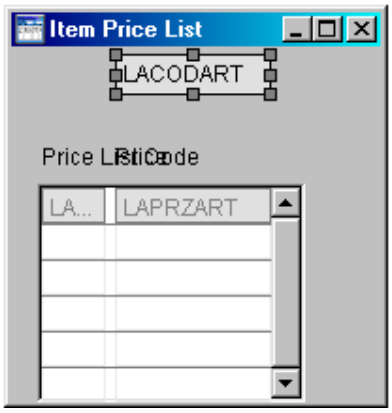


Picture 43 - Reducing The Dialog Window

Changing The Detail File Header

Right click 'Item Code', and select 'Delete'. Double click the field 'LACODART".

To hide the field, select the flag 'Hide'. Confirm with 'OK' and move the field at the top left of the page.



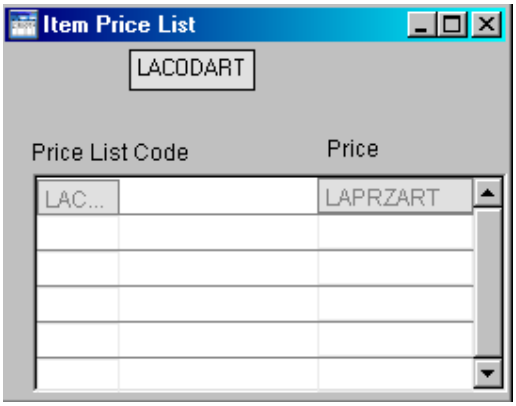
Picture 44 - First Changes

Move The Body Field Descriptions

Enlarge the window slightly, select the description field 'Price List Code' and, keeping the mouse pressed, move it to the top left. Move the description field 'Price' to the right.

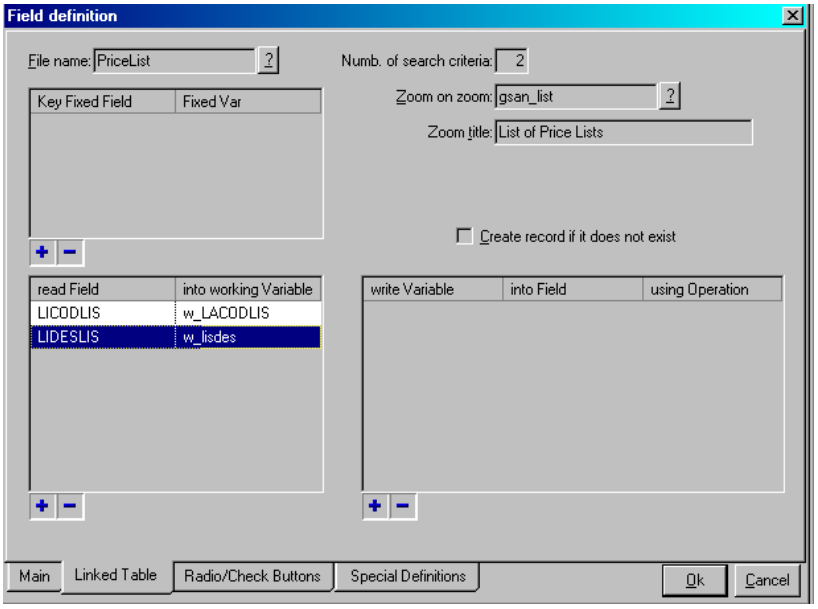
To change the body, open the 'Pages' menu and select 'Body'. Keeping the mouse pressed move the field 'LACODLIS' to the left under its description. Then move 'LAPRZART' to the right under its description. Adapt the window size.

Picture 45 - Changes in the 'Item Price Lists' Window



Decoding The 'Price List Code' Description

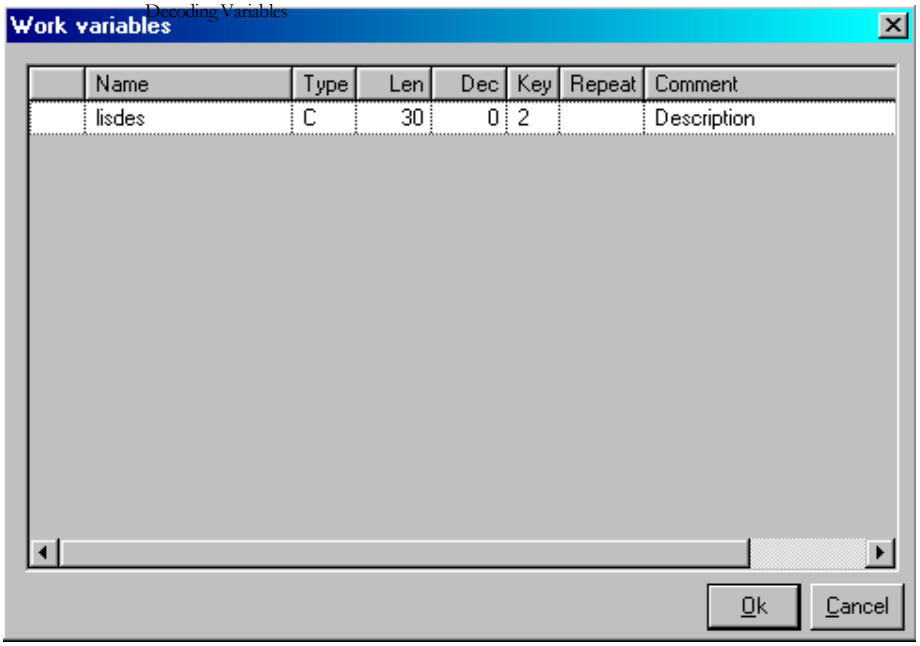
Double click 'LACODLIS' to open the definition dialog window, go to the 'Linked Table' sheet and change the field 'No. of search criteria' to '2'. The Zoom functionality will use also the secondary index. Go to the 'Zoom Title' area an input 'List of Price Lists'. Go to the left column under the field 'Read Field Into Working Variable' and click the '+' button to define a new link row. Select the field 'LIDESLIS' on the left and digit 'w_lisdes' to the right. You have just defined a variable containing the description decoding. Click outside the window area and then 'OK' to confirm your entries.



Picture 46 - Decoding The 'Price List' Description

Go in the 'Documents' body, open the 'Item' menu, select 'Variable' and click next to the field 'LACODLIS'. The Variable Definition Window opens up. Click the '?' button next to 'Name' and double click the variable you created in the previous link.

Picture 47 - List Of Decoding Variables

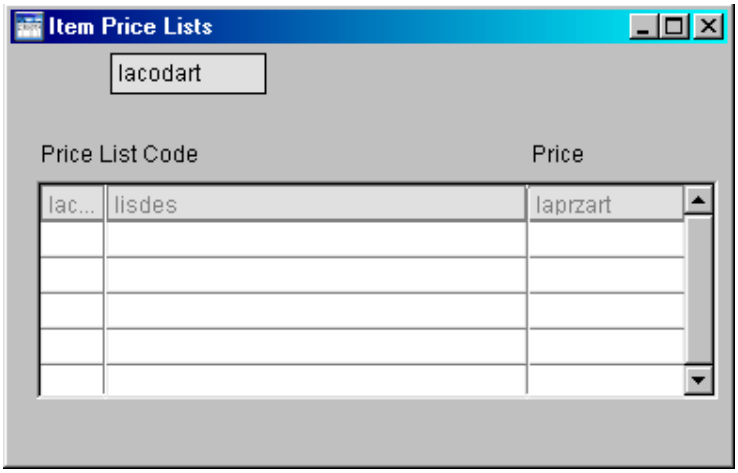


Change the status to display only activating the 'Show' flag and confirm with 'OK'. To align the variable with the field 'LACODLIS', press <Shift> and click both, variable and the field. Open the 'Align' menu and select 'Top'. To move the variable to the side simply click it and use the 'Arrows' keys.

Picture 48 - 'Item Price Lists Decoding

Change The Window's Layout

Open the 'Pages' menu and select 'Header'. To insert dividing columns open the 'Items' menu, select 'Box' and add it between 'LACODLIS' and 'LISDES'. Reduce the box width to a line, which goes along the body's rows. Do the same between "LISDES" and "LAPRZART".



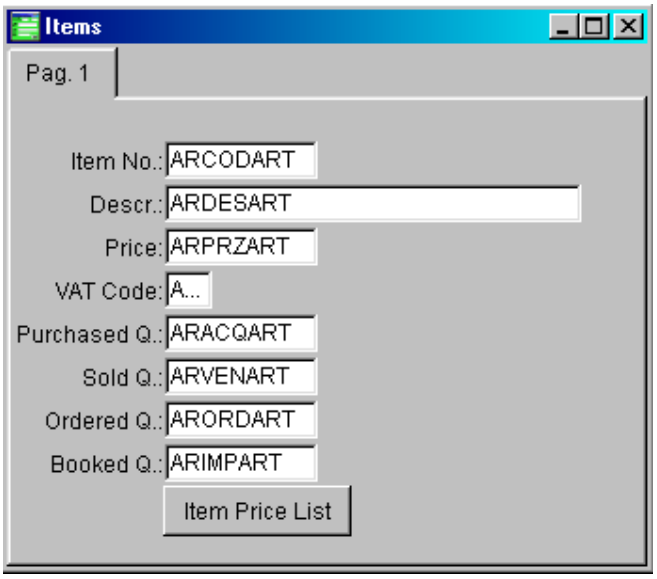
Picture 49 - Changing The Window's Layout

Align the field descriptions strings, open the 'File' menu and select 'Save' to save your changes. Open the 'File' menu and select 'Exit' to go back to the Front End.

5.1.2 Changing The 'Items' Entity

Go back to your plan in the 'Design Tool', right click the 'Item' entity. and select the 'Open Codify' option. The Master File Painter is automatically opened showing the 'Items' prototype.

Picture 50 - The 'Items' Prototype Window



Change The 'Quantities' to 'Display' Only

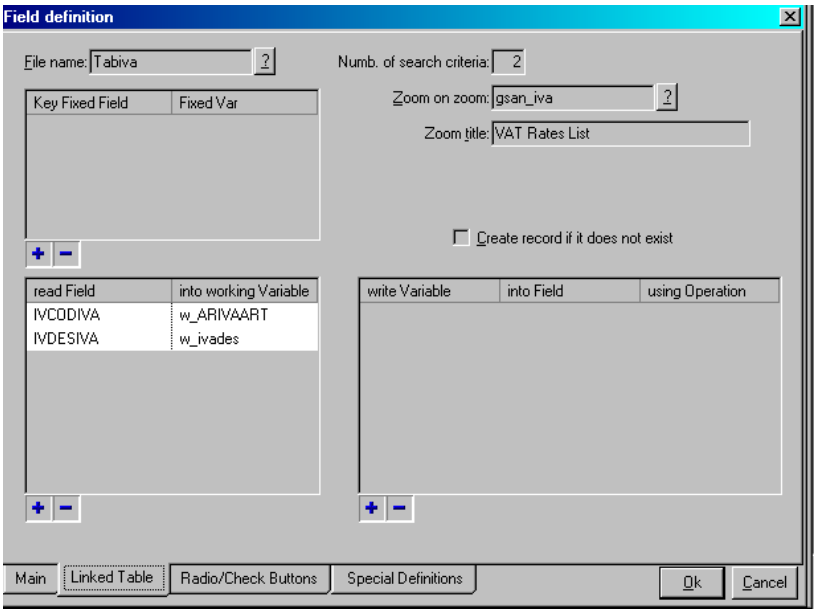
In order to change the fields to 'display' only, double click 'ARACQART', activate the 'Show' flag and confirm with 'OK'.

Picture 51 - The 'Editing' Flag Must Be Set To 'Show' For All 'Quantity' Fields' Description

Do the same for the fields 'ARVENART', 'ARORDART' and 'ARIMPART'.

Decoding The 'VAT Rates' Description

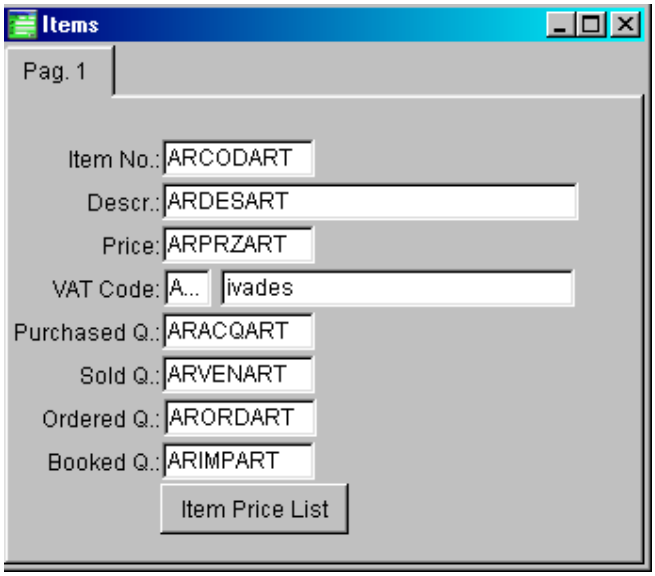
To use the secondary index in the 'Zoom' functionality double click 'ARIVAART", select the 'Linked Table' sheet and change the field 'No. of search criteria' to '2'. Go to the 'Zoom Title' and add 'VAT Rates List'. Go to 'Read Field Into Working Variable', click the '+' button to add a new row for a new link. On the left select 'IVDESIVA' and on the right digit the decoding variable 'w_ivades'.



Picture 52 - Link to the 'VAT Rates' Table

Click outside the window area and confirm with 'OK'. Open the 'Items' menu and select 'Variable'. In the Definition Window click the '?' button next to 'Name' and double click 'ivades'. You have come back to the previous window, activate the 'Show' flag and confirm with 'OK'. Align the variable next to the 'VAT Code' as you did in the previous section.

Picture 53 - Description Decoding

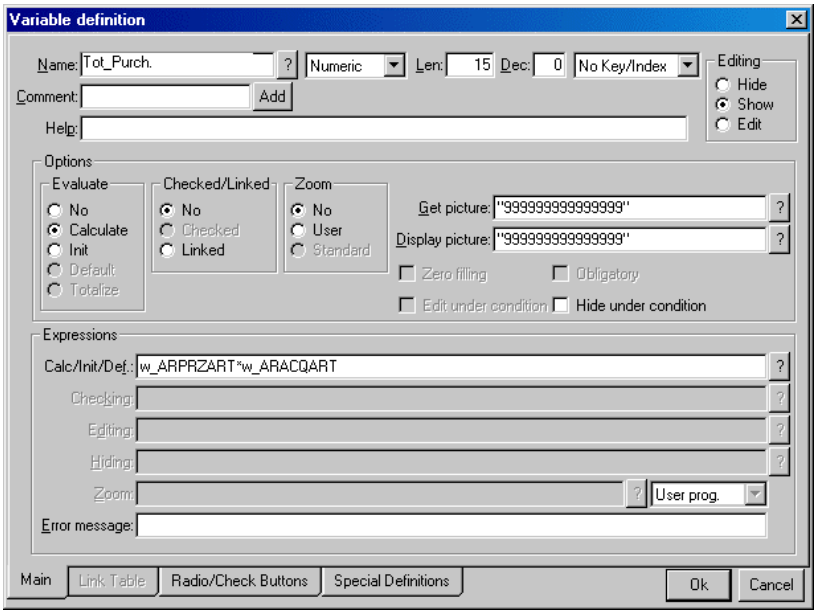


Display 'Items' Quantities

You need to create four variables to default values calculated basing on price and quantity. Add a new variable next to 'ARACQART". In the definition dialog window input 'Tot_Purch' next to 'Name', change the field type in 'Numeric' and define the lengths ('Len') as '15'.

To define display attributes of the variable go to 'Get Picture', and click the '?' button to default attributes. Do the same with 'Display Picture'.

To define the variable's value and the formula it must be calculated with, go to the 'Options' section of the window and in the 'Evaluate' box select 'Calculate'. Then go to the 'Expression' section and input the following formula next to 'Calc/Init/Def.': 'w_ARPRZART * w_ARACQART". Click 'OK' and then align the variable next to the field 'ARACQART'.



Picture 54 - 'Total Purchased' Variable

You now need to create other three variables to calculate sold, ordered and booked quantity. Follow the same steps as before naming the three variables 'Tot_Sold', 'Tot_Ord' and 'Tot_Book' and input the following formulas:

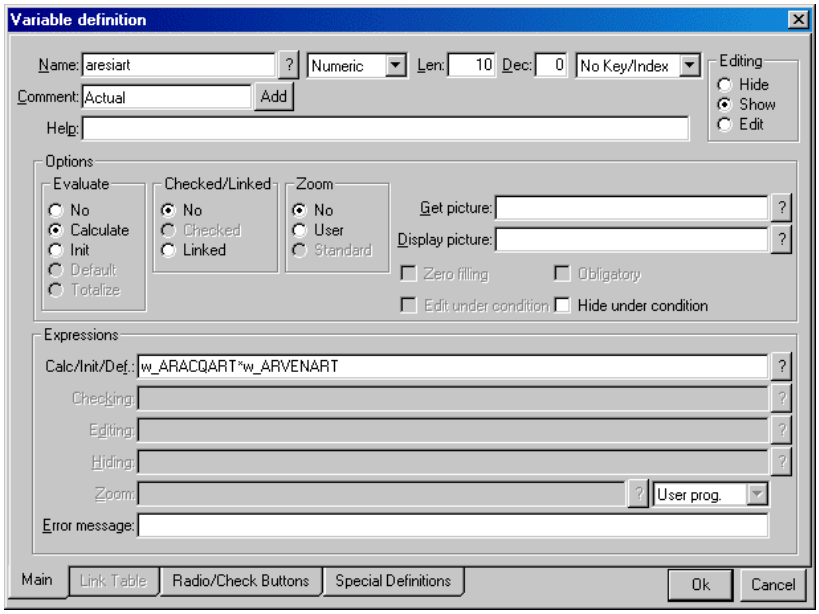
'w_ARPRZART * w_ARVENART',

'w_ARPRZART * w_ARORDART'

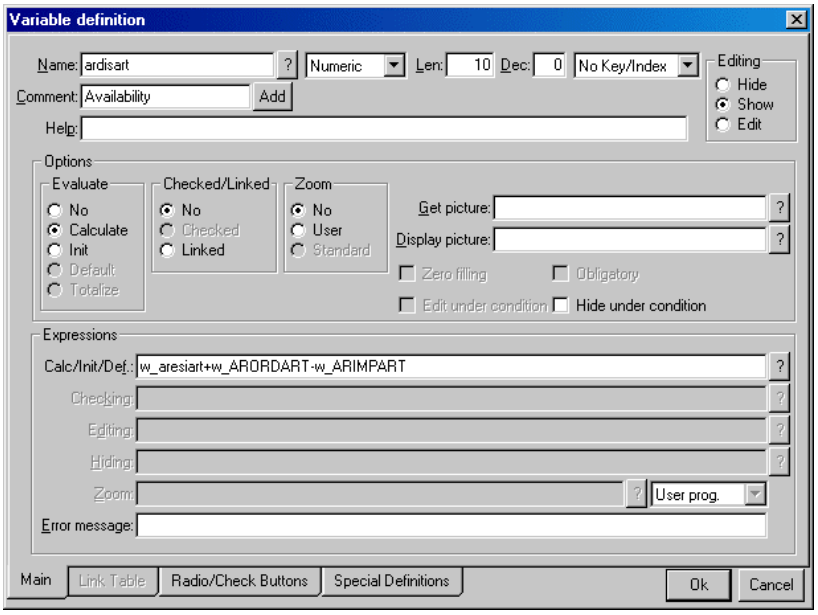
'w_ARPRZART * w_ARIMPART'.

Align the variables next to the corresponding quantity fields. Add a new variable ('aresiart') to calculate items availability. Define the variable as 'Numeric', 'Show' and long '10'. Next to 'Comment' input 'Actual' and click the 'Add' button. Go to the 'Options' section and in the 'Evaluate' box select 'Calculate'. Then go to the 'Expression' section and input the following formula next to 'Calc/Init/Def.'w_ARACQART - w_ARVENART". Click 'OK' to confirm.

Picture 55 - Actual Variable



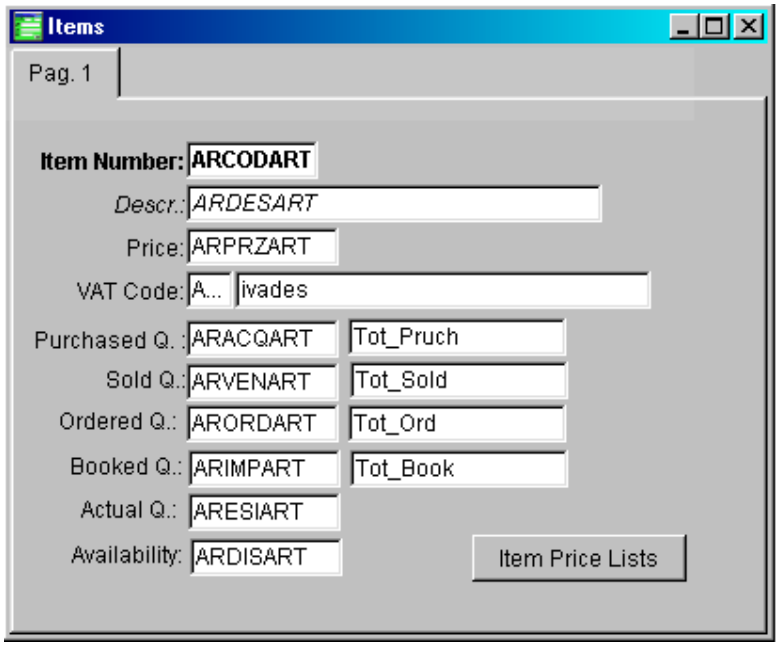
Under the variable 'aresiart' add a further variable ('ardisart') to display the items' availability. Input the same characteristics as 'w_aresiart' with the following formula: 'w_aresiart + w_ARORDART - w_ARIMPART'. In the 'Comment' field input 'Availability', click the 'Add' button, and then 'OK'.



Picture 56 - Availability Variable

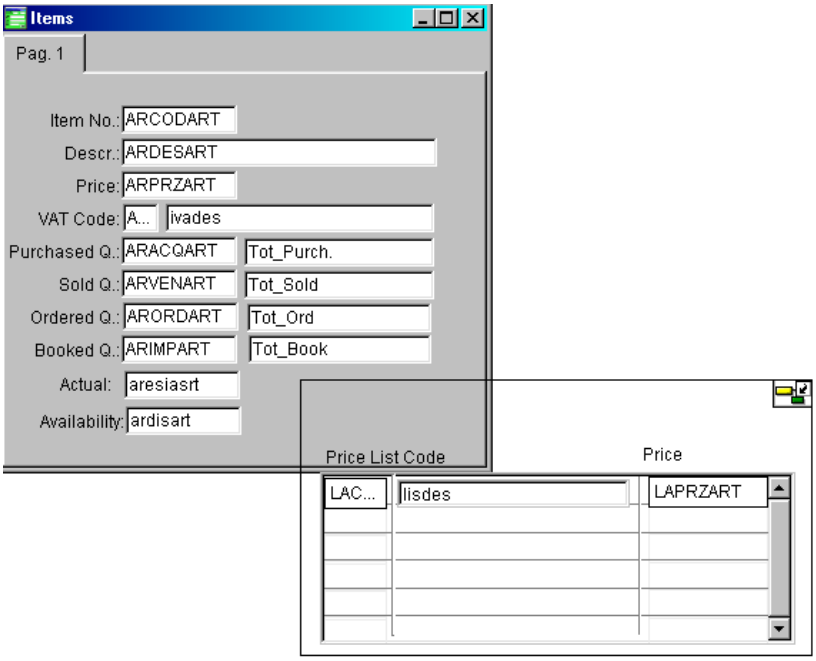
Position the two last variables under the quantity fields and move the 'Item Price Lists' button to the right.

Picture 57 - The 'Items' Window With The New Variables



Integrating The Child Entity In The Parent Entity ('Item Price Lists')

Double click the child entity 'Item Price Lists' (button), go to the 'Child Editing' area and select 'Edit'. Click the 'Get Child Size' button to activate the size. Click 'OK' to confirm. The 'Item Price Lists' image is displayed on the 'Items' entity.



Picture 58 - Integrated Child Window

Change The Window's Layout

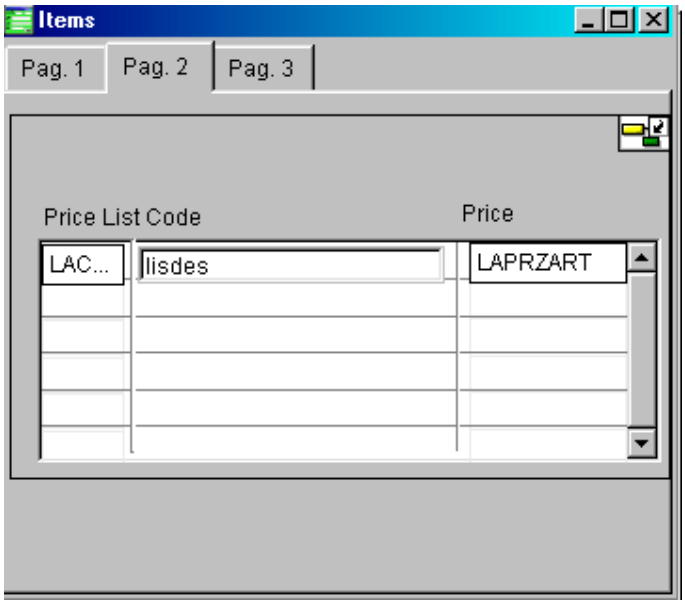
- Split the elements on more pages:

Group fields logically together, e.g. in three main groups: items, price-lists and totals.

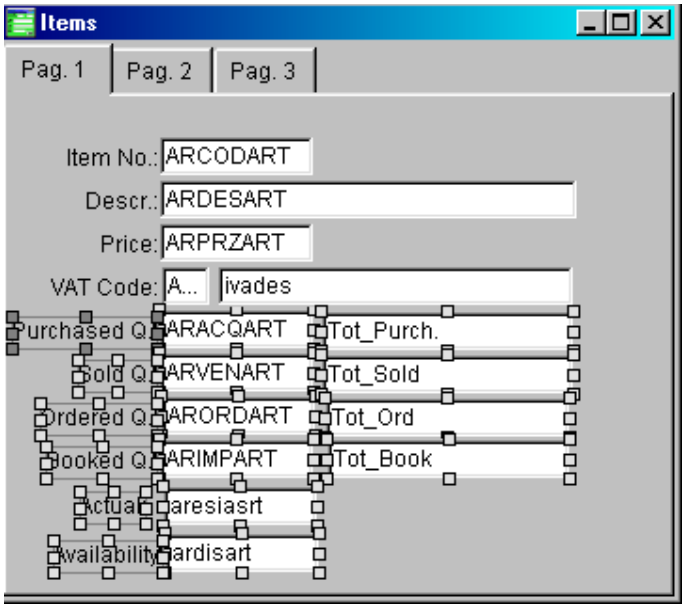
Open the 'Pages' menu and select 'Add page' to add a new page in the dialog window. Create a third page. To go back to the first page open the 'Pages' menu and select 'Page1'.

Click the 'Child' image, open the 'Edit' menu and select 'Cut' to delete it and move it temporary to memory. Open the 'Pages' menu and select 'Page2'. Open the 'Edit' menu and select 'Paste'. Click the window to paste the entity.

Picture 59 - Window Displaying The Integrated Child on 'Page2'



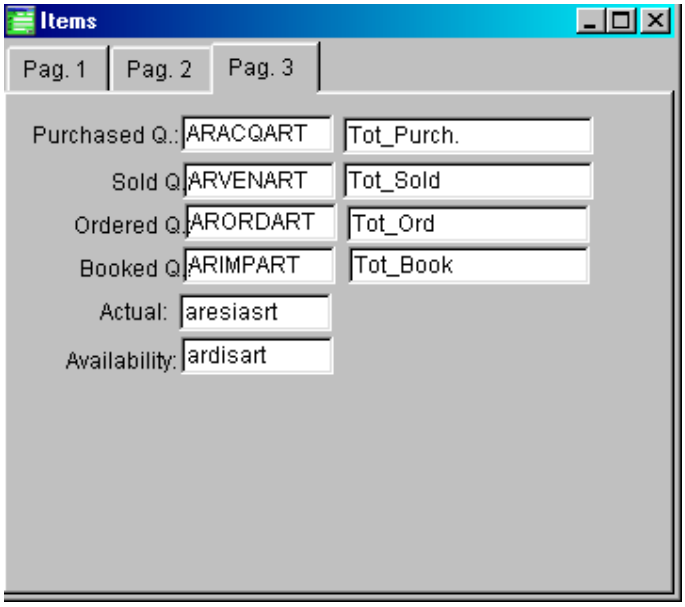
Go back to 'Page1' and pressing <Shift> select all fields and variables from 'Purchased Quantity' onwards. Open the 'Edit' menu and select 'Cut'.



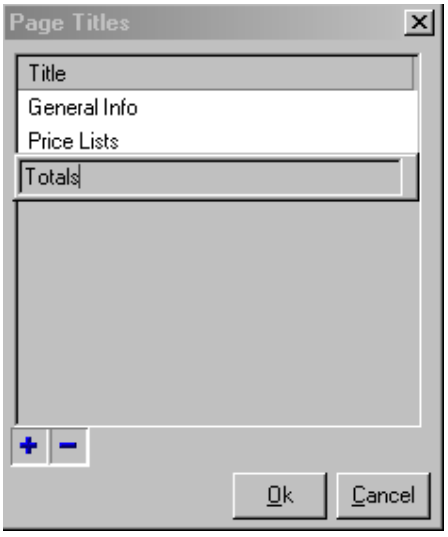
Picture 60 - Selecting Elements Of 'Page1'

Go to 'Page3', open the 'Edit' menu and select 'Paste' to paste the selection.

Picture 61 - Pasted Selection On 'Page3'



Open the 'Globals' menu and select 'Page Titles' to open a new dialog window. Click the '+' button, digit the string 'General Info' and press <Enter>. Add a second string 'Price Lists' and a third 'Totals' and click 'OK'



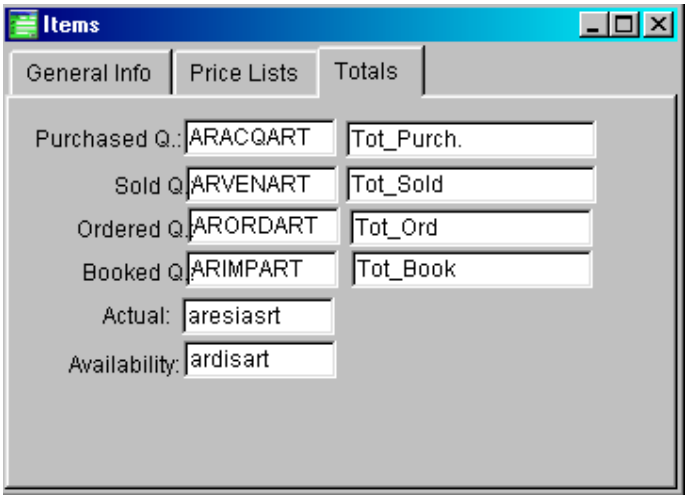
Picture 62 - Changing 'Pages' Names

- Highlight the key field and its description using the font 'Arial', Bold', sized '9':

Double click the 'Item No.' string to edit it. Deselect the 'Global Font' flag, click 'Change Font', select 'Arial', 'Bold', size '9' and confirm with 'OK'. Edit the field 'ARCODART', go to the 'Special Definitions' field and do the same.

Reduce the window size as to optimize the field arrangement in each page.

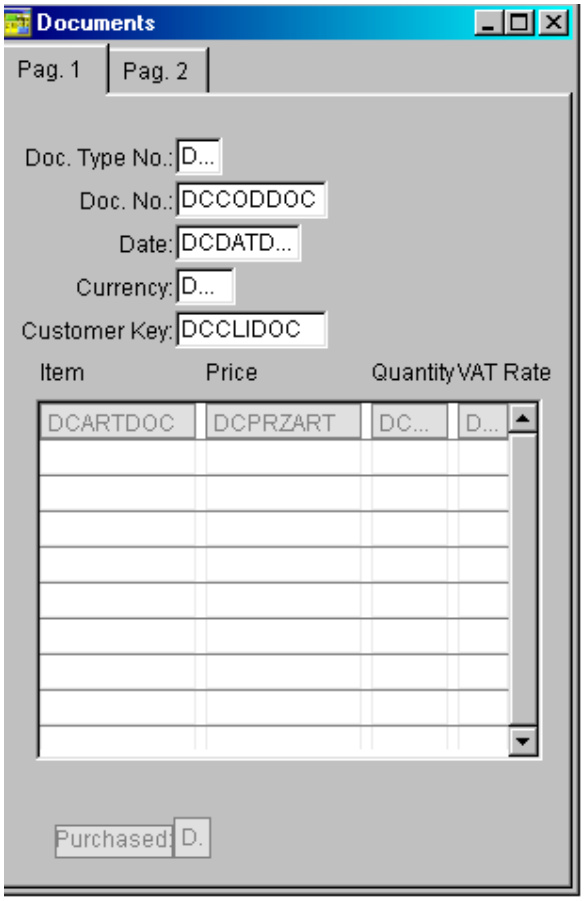
Picture 63 - 'Items' Management Definition Dialog Window



Open the 'File' menu and select 'Exit'. A dialog window asking if you want to save the changes appears. Click 'Yes' to confirm. You are back to the Front-End.

5.1.3 Changing The 'Documents' Entity

Click the Master Detail icon on the Front End, open the 'File' menu, select 'Open' and double click 'gsmd_doc'.



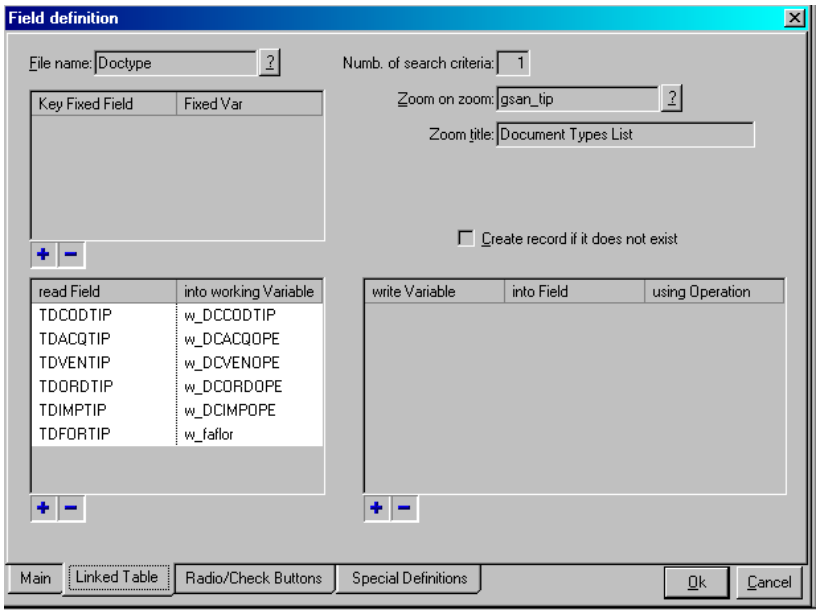
Picture 64 - Master/Detail Prototyping Window

Decoding The 'TDFORTIP' Flag in 'Document Types'

You need to define a variable for downloading from 'Document Types' the value stored in the 'TDFORTIP' flag. This allows identifying whether to extract from the 'Customers' file a customer or a supplier. Mathematical signs should not be displayed.

Double click the field 'DCCODTIP' (the first in the window) and go to the 'Linked Table' sheet. In the 'Read Field Into Working Variable' area click the '+' button. Select 'TDFORTIP' on the left, and on the right add a new variable, named 'w_flafor'. In the 'Zoom Title' digit the string 'Document Types List' and confirm with 'OK'.

Picture 65 - Link With 'Document Types'

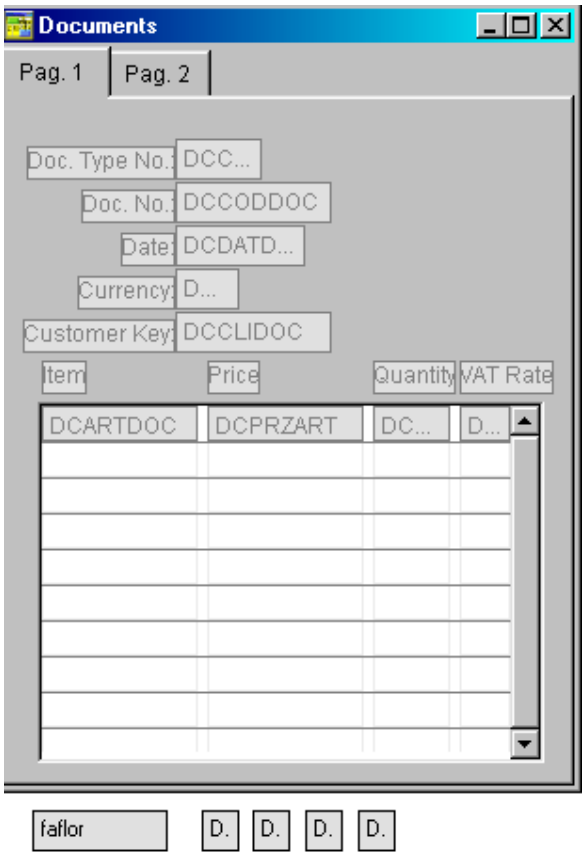


Open the 'Page' menu and select 'Footer'. Go to a blank area of the footer, add a new variable, click the '+' button next to the field 'Name', and select 'flafor'. Change the flag from 'Editing' to 'Hide' and confirm with 'OK'.

Open the 'Pages' menu and select 'Footer2'. Select the three remaining fields, open the 'Edit' menu and select 'Cut'.

Go back to the 'Footer', open the 'Edit' menu and click 'Paste'. Edit the four fields and in the 'Editing' area activate the 'Hide' flag.

Right click the the string 'Purchase', and select 'Delete'. Do the same for the next three descriptions. Reduce the window size and move the four fields, their mathematical signs and the 'flafor' variable under the window as displayed in Picture 23.

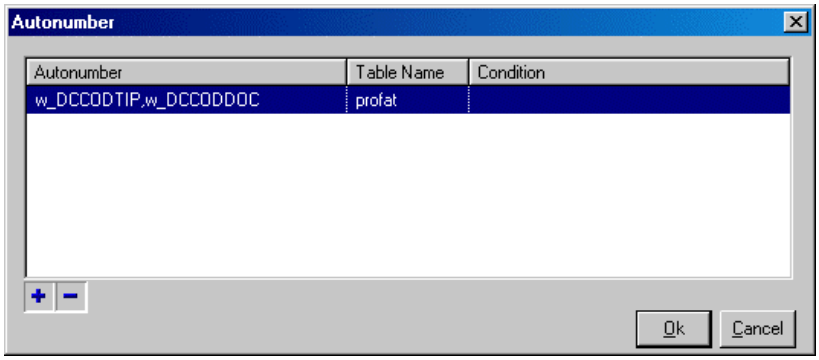


Picture 66 - Moving Mathematical Signs

Autonumbering

Open the 'Globals' menu and select 'Autonumber'. In the first row add the field defined for autonumbering. In this exercise you need to add two fields, because the primary key is made of two fields. Click the '+' button and the first '?' button. Select 'DCCODTIP' and 'DCCODDOC'. Define the table that must be used, digit 'profat' in the 'Table Name' and confirm with 'OK'.

Picture 67 - 'Autonumber' Definition Dialog Window

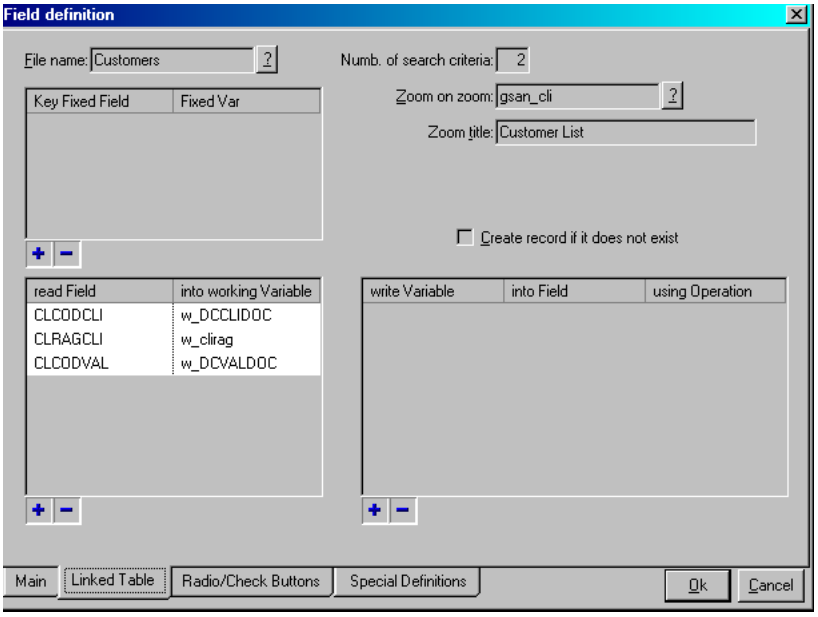


Decoding Customer Details And Currencies

When a value containing a customer key is input in the field 'DCCLIDOC', the software application must validate it and default the description and the currency defined for that customer.

Open the 'Pages' menu, select 'Header' and double click 'DCCLIDOC'. Select the 'Linked Table' sheet, go to the 'Read Field Into Variable' area, click the '+' button, click the left '?' button to activate the 'Customers' file and double click the field 'CLRAGCLI'. Go to the next field and define the decoding variable as 'w_clirag'. Click the '+' button to add a new row, select 'CLCODVAL', click the right '?' button and select 'DCVALDOC'.

Change the field 'No. of search criteria' to '2' in the 'Zoom functionality' to use the secondary index, go to 'Zoom Title and input the caption 'Customer List'. Confirm with 'OK'

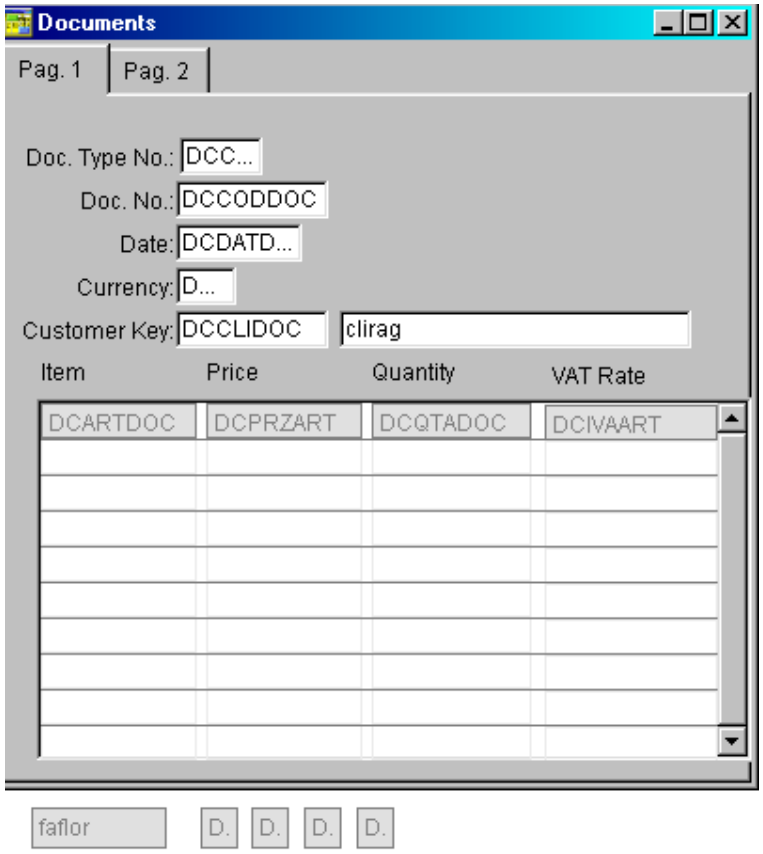


Picture 68 - Link With The Customer File

Add a variable next to the field 'DCCLIDOC'. In the definition dialog window click the '?' button next to 'Name' and select 'clirag'. Change 'Editing' to 'Show' and confirm with 'OK'.

Allocate the variables tidily and reset the size so that all elements fit in the window.

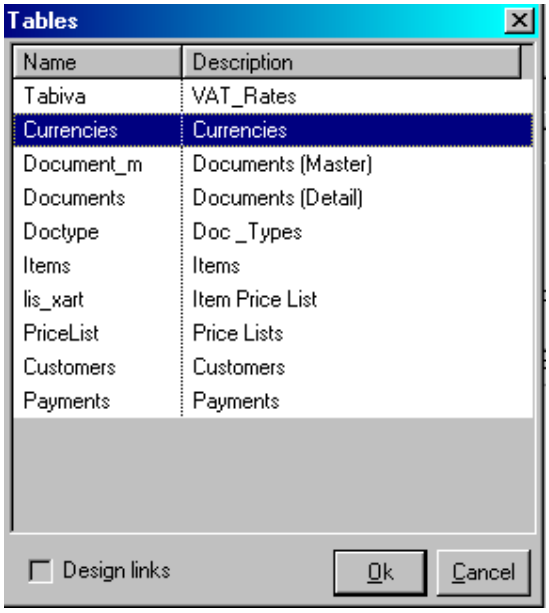
Picture 69 - Decoding The 'Company Name'



Decoding Currency Value And Symbol

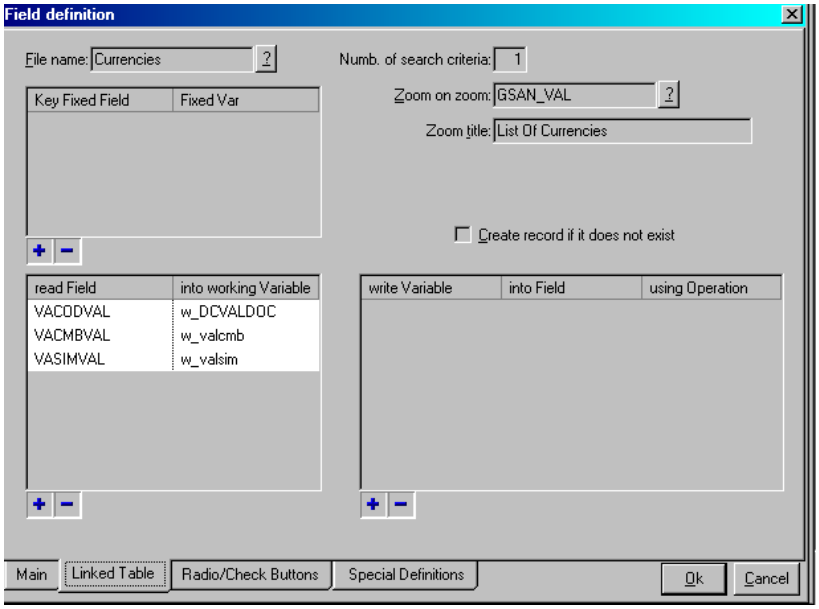
When a value containing a currency code is input in the field 'DCVALDOC', the software application must validate it and default exchange rate and currency symbol from the chosen record.

Edit the field 'DCVALDOC' in the 'Checked/LInked' area, select 'Linked', and in the 'Zoom' area select 'Standard'. Go to the 'Linked Table' sheet, click the '?' button next to 'File name', deselect the 'Design Links' flag and select 'Currencies'.



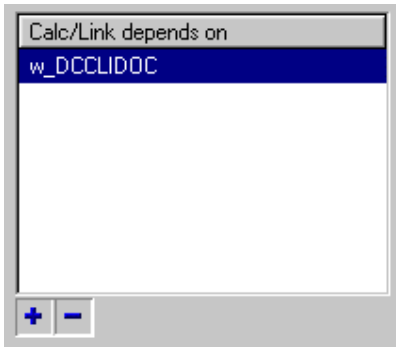
Picture 70 - Selecting The Link Table

Add a new row in the 'Read Field Into Working Variable' area. Click the left '?' button, select 'VACODVAL', and leave the defaulted value on the right. Add another row. On the left select 'VACMBVAL' and on the right write the variable 'w_valcmb' as decoding value. Add a further row. On the left select 'VASIMVAL' and on the right write 'w_valsim'. Next to 'Zoom Title' write 'List of Currencies'.



Picture 71 - Link Definition Dialog Window

Go to the 'Special Definitions' sheet, select the 'Calc/LInk Depends On' area, click the '+' button and use the '?' button to select the field 'DCCLIDOC'. Confirm with 'OK': The link will be re-executed each time this field is updated.

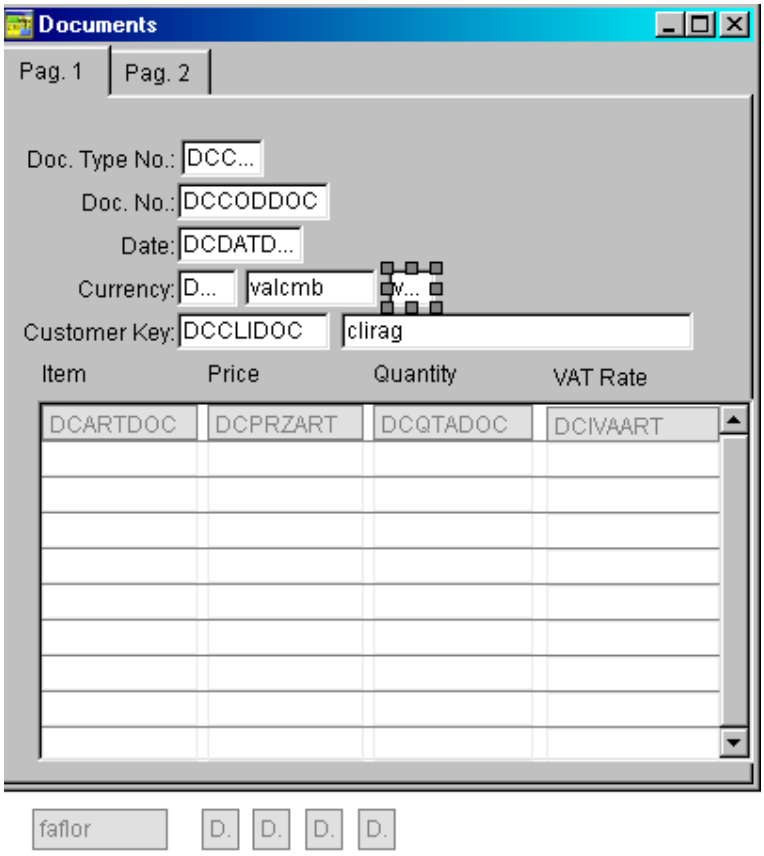


Picture 72 - 'Calc/LInk Depends On' Area

Add a new variable next to the field 'DCVALDOC'. In the definition dialog window click the '?' button next to 'Name' and select 'valcmb'. Change the flag from 'Editing' to 'Show' and confirm with 'OK'.

Add a further variable next to the previous one. In the definition dialog window click the '?' button next to 'Name' and select 'valsim'. Change the flag from 'Editing' to 'Show' and confirm with 'OK'.

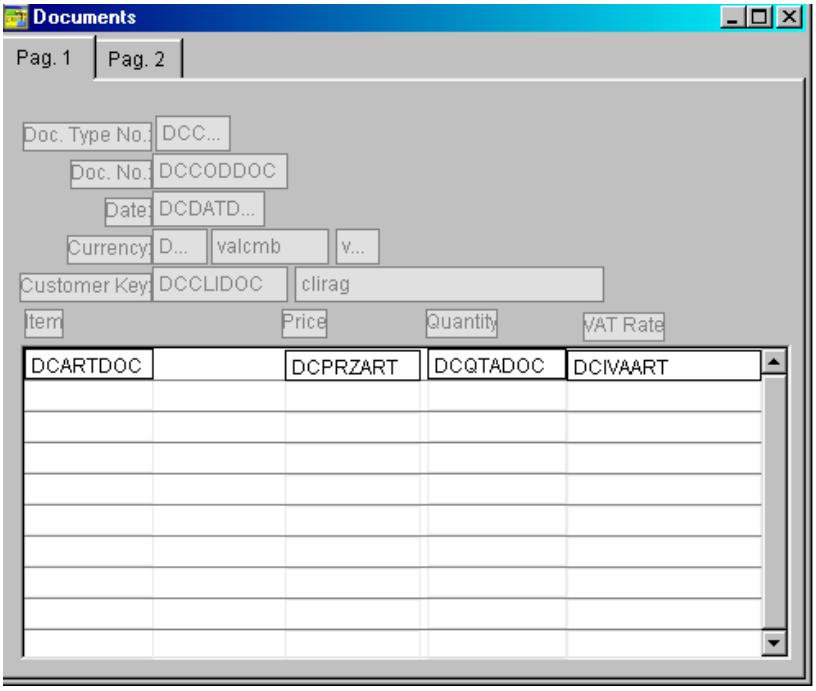
Picture 73 - Window Showing The Added Variables



Decoding 'Item Description', 'VAT Code' and 'Price'

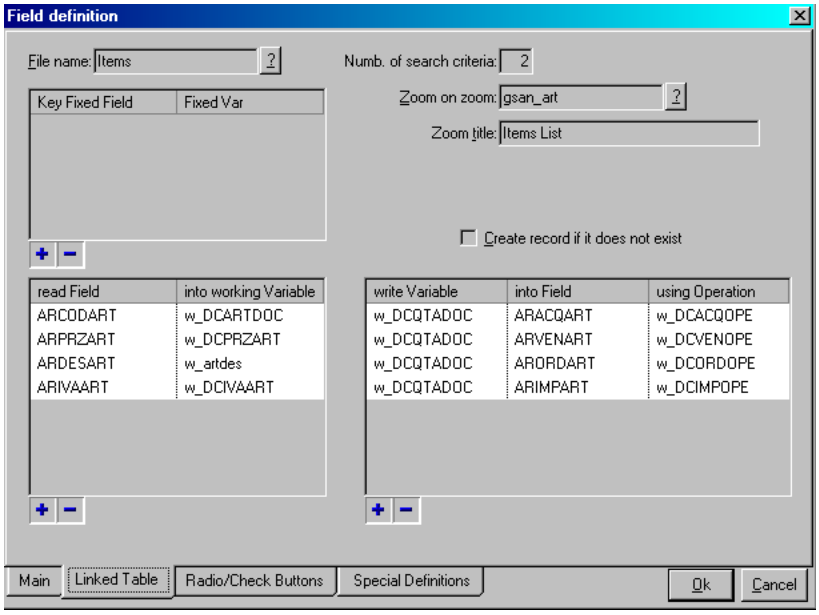
When a value containing an item code is input in the field 'DCARTDOC', the software application must validate it and default the VAT Code and the price from the chosen record.

Open the 'Pages' menu and select 'Body'. Select the field 'DCPRZART', press <Shift> and simultaneously select the other fields and their descriptions. Using the 'Arrows' keys move them to the right. Afterwards increase the window's widths.



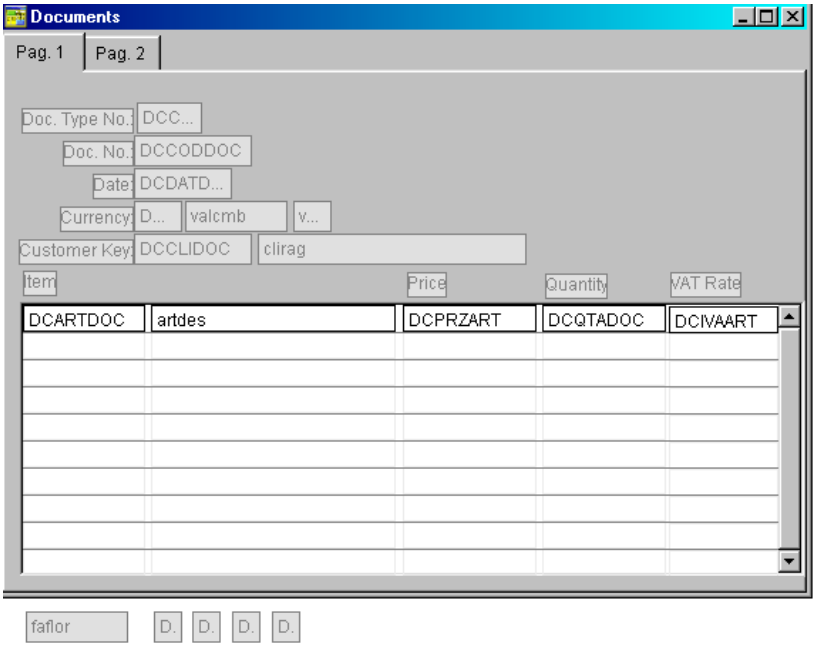
Picture 74 - Allocation Of Body Fields.

Edit the field 'DCARTDOC', go to the 'Linked Table' sheet, in the 'Read Field Into Working Variable' area select 'ARDESART' on the left and digit the decoding variable 'w_artdes' to the right. Add a new row and select 'ARIVART' on the left and select the decoding value 'DCIVART' on the right. Change 'No. of search criteria' to '2' so that the 'Zoom Functionality' also uses the secondary index, input the caption 'Items List' and confirm with 'OK'.



Picture 75 - Link Window

Add the variable containing the item description. Using the '?' button in the definition dialog window select 'artdes' next to 'Name', change the flag to 'Show' and confirm with 'OK'. Select the variable and move it next to 'DCARTDOC'. Rearrange the windows elements so that they all fit in it.

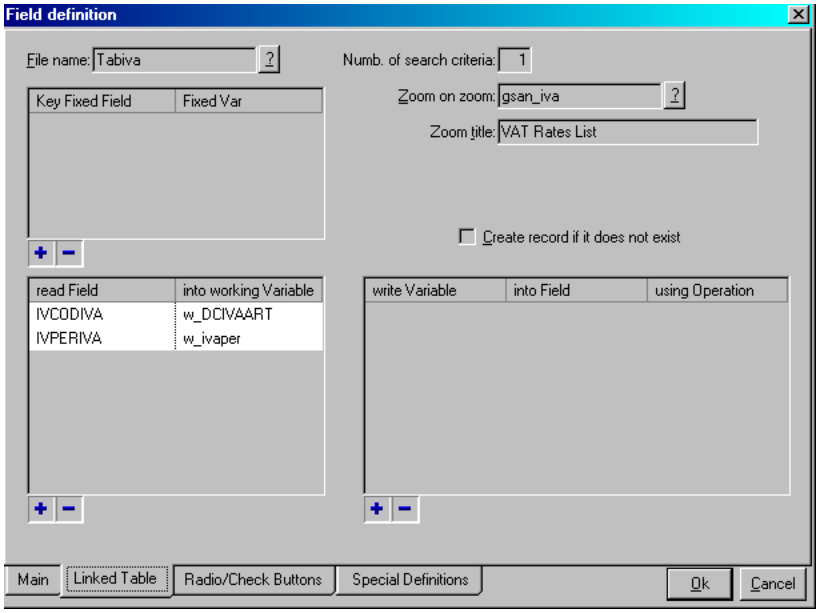


Picture 76 - Description Variable Added In The Window

Decoding The VAT Rate

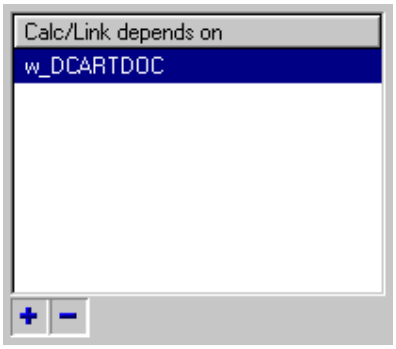
When a value containing a VAT Rate is input in the field 'DCIVAART', the software application must validate it and default the VAT Rate from the chosen record.

Select the fields 'DCPRZART' and 'DCQTADOC', move them to the right and then move the 'VAT' field next to the 'Item Description'. Edit the field 'DCIVAART', select 'Linked' in the 'Checked/LInked' area, select ' Standard' in the 'Zoom' area and then go to the 'Linked Table' sheet. Click the '?' button next to 'Table Name', deselect the check box 'Design Links' and select 'tabiva' from the pick list. Go to the 'Read Field Into Working Variable' area, add a new row using the '+' button, select 'IVCODIVA' on the left, and leave the defaulted value on the right. Add a further row, select 'IVPERIVA' on the left and digit 'w_ivaper' on the right. Write 'VAT Rates List' next to 'Zoom title'.



Picture 77 - Link Definition For VAT Decoding

Go to the 'Special Definitions' sheet, add a new row in the 'Calc/LInk Depends On' area, click the '?' button and select 'DCARTDOC'. When the 'Item Code' is changed the link execution is overridden.



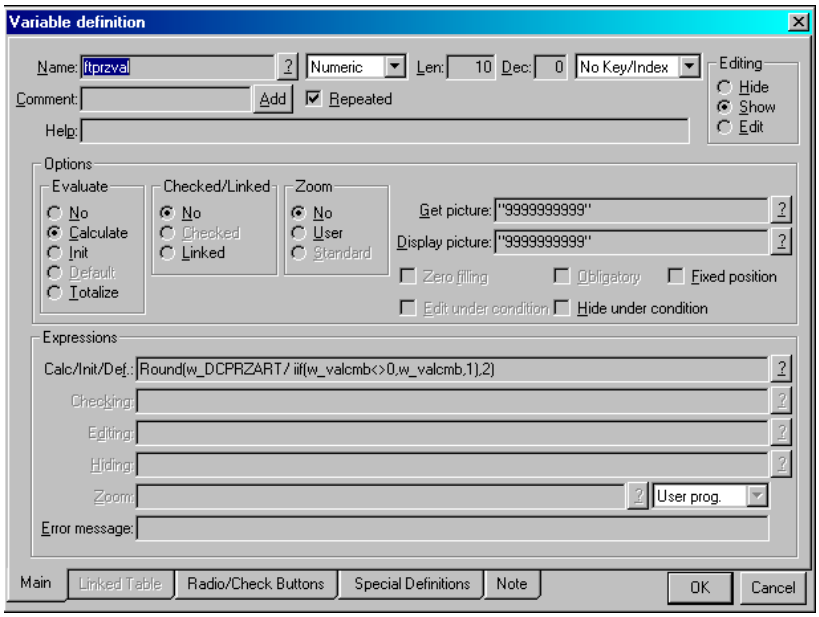
Picture 78 - 'Calc/LInk Depends On' Dialog Window

Confirm with 'OK'. Add a new variable next to 'DCIVAART' and using the '?' button select 'ivaper' next to 'Name'. Change the editing option to 'Show' and confirm with 'OK'.

Price Calculation Basing On Currency

To display the item's price in the selected currency, select 'DCPRZART' and move it downwards in order to leave enough space to add a new variable. Edit 'DCPRZART' and change the editing option to 'Hide'. Add a new variable on top of it, write 'ftprzval' next to 'Name', and change it to 'Numeric', set the length to '10' and decimals to '2'. In the 'Get Picture' and in the 'Display Picture' area click the '?' button to add the variable's picture. Set the 'Evaluate' vertical radio to 'Calculate', go to the 'Calc/Init/Def.' field and digit the following code:

```
Round(w_DCPRZART/iif(w_valcmb<>0,w_valcmb,1),2)
```



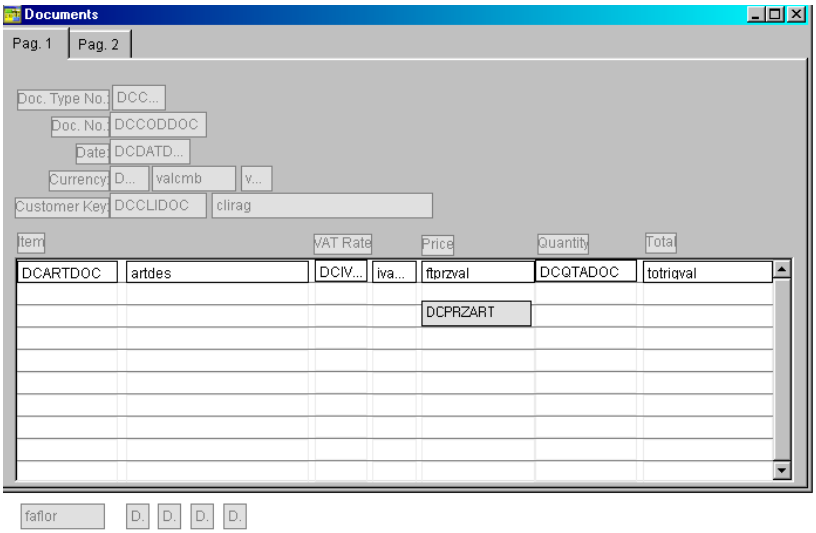
Picture 79 - FTPRZVAL Definition Dialog Window

Go to the 'Special Definition' sheet and set the 'Display Length to '13'.

If the exchange rate is different from '0' the variable 'ftprzval' is calculated dividing the 'Item Price' by the defined 'Exchange Rate'. Click 'OK' to confirm and align the added variable.

Adding Rows' Totals

You now need to add the total of each row (Price * Quantity) so that you can also have the Total of the document. To do so, add a new variable next to 'DCQTADOC', name it 'totrigval', digit 'Total' in 'Comment', and click 'add' to add the description. Change the type to 'Numeric', the length to '12' and decimals to '2'. Click the '?' button next to 'Get Picture' and 'Display Picture' to default values and change the editing option to 'Show'. Set the 'Evaluate' vertical radio to 'Calculate' go to the 'Calc/Init/Def.' field and digit the following code: 'w_ftprzval * w_DCQTADOC'. Go to the 'Special Definition' sheet and digit '15' next to 'Display Length'. Confirm with 'OK'. The new variable goes over the right window frame. Adjust the window's size and align the fields.

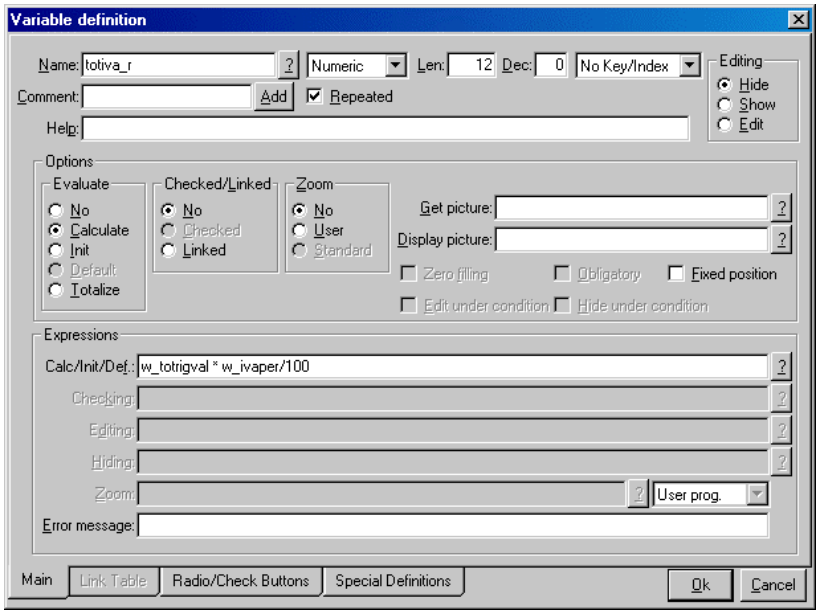


Picture 80 - Variable ('totrigval') Adjustment

Total VAT Amount Per Row

You now need to add the VAT amount on each row so that you can also have the total VAT amount fot the whole document. To do so, add a new variable under 'totrigval', name it 'totiva_r', set the type to 'Numeric', the length to '12', and decimals to '2'. Change the editing option to 'Hide'. Set the 'Evaluate' vertical radio to 'Calculate', go to the 'Calc/Init/Def.' field and digit the following code: 'w_totrigval * w_ivaper/100'.

Picture 81 - Defining The Variable 'totiva_r'



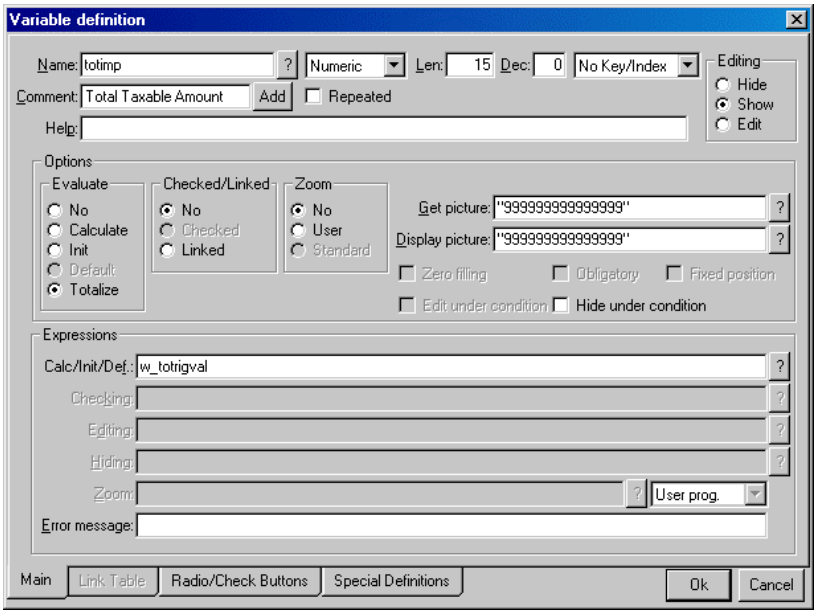
Confirm with 'OK'.

'Total Taxable Amount', 'Total VAT Amount' and 'Total'

You now need to add the Total, the Taxable Amount, the Total VAT Amount and the Total for the whole document. You need to define these totals as variables, which work on row's totals ('w_totrigval' for taxable amount and 'w_totiva_r' for VAT amount).

Open the 'Pages' menu and select 'Footer'. Increase the window's size so that three new variables fit one under the other.

Add a new variable (Total Taxable Amount) under the 'Totals' column, name it 'tottax', write the description in 'Comment' and click 'Add'. Set the field type to 'numeric' the length to '15', the 'Editing' option to 'Show' and click the '?' button next to both 'Get Picture' and 'Display Picture' to default values. Go to the 'Evaluate' area, select 'Totalize' (display only), digit 'w_totrigval' next to 'Calc/Init/Def.' and confirm with 'OK'. Go to the 'Special Definition' sheet and define the 'Display Length' to '18'.



Picture 82 - Defining The Variable 'totimp'

Add the 'VAT Total Amount' variable ('totvat') following the same steps. In 'Calc/Init/Def.' write 'w_totiva_r'. Confirm with 'OK' and align it under 'tottax'.

Add the 'Total Amount' ('totamt') under 'totvat'. In the 'Calc' area select 'Calculate' and next to 'Calc/Init/Def.' digit the following formula: 'w_tottax + w_totvat'. Confirm with 'OK'.

Align the three variables and their descriptions on the right of the window.

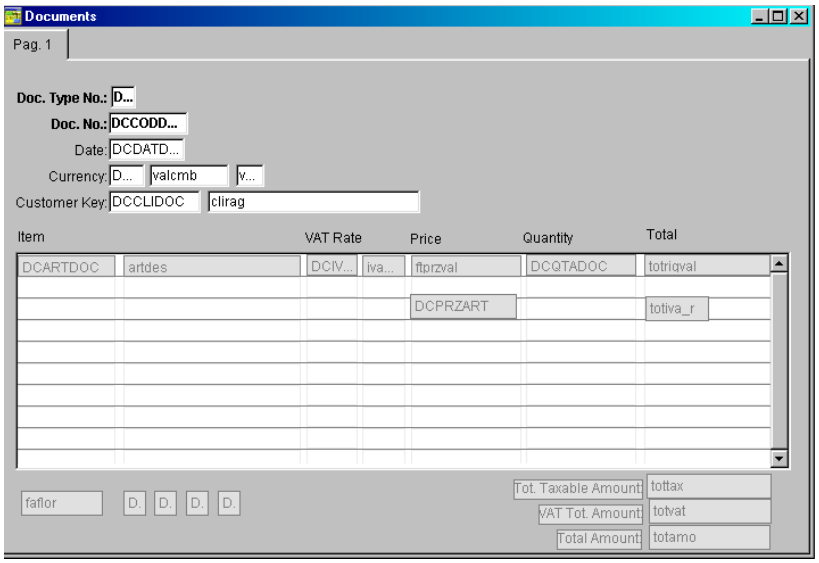
Improve The Window's Layout

- Highlight key fields and its descriptions using 'Arial', 'Bold', sized '9'.

Edit the string 'Document Type No.', deselect the 'Global Font' flag, and click 'Change Font'. Select 'Arial', 'Bold', sized '9' and confirm with 'OK'. Do the same for the field 'DCCODTIP'. Go to the 'Special Definition' and define the 'Display Length' to '18'.

Follow the same steps for 'Document Number' (both, string and field 'DCCODDOC'). Move all header fields to the right so that the descriptions fit in the window.

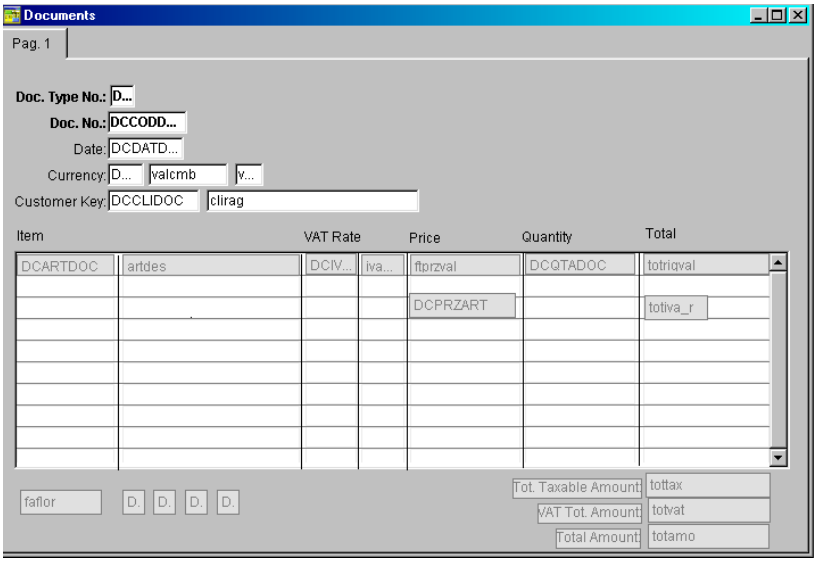
Picture 83 - Changing The Header



- Using dividing columns.

Open the 'Pages' menu, select 'Footer', and add a 'Box' object between the body fields 'artdes' and 'DCIVAART". Reduce the box's size to a line, which goes along the body, then repeat the same between the following fields: 'ivaper' and 'ftprzval'; 'ftprzval' and 'DCQTADOC'; 'DCQTADOC' and 'totrigval'.

Change the description of the fields and its arrangement in the window.



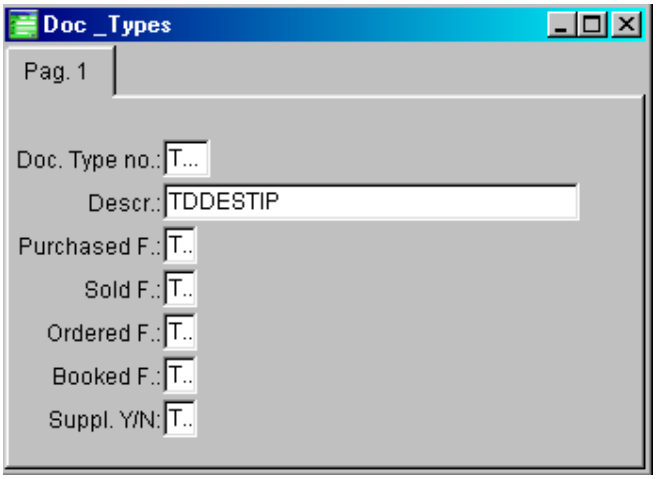
Picture 84 - Changes To The Body

Open the 'File' menu and select 'Save' to save changes. Open the 'File' menu and select 'Exit' to close the selected Painter.

5.2 Changing Other Entities

You can now change the layout of the remaining Master Files following the same standards. Pay attention to the flags in 'Document Types'. To optimize layout and functionality you can manage flags using ComboBoxes. Open the 'Master' Painter and open 'gsan_tip'.

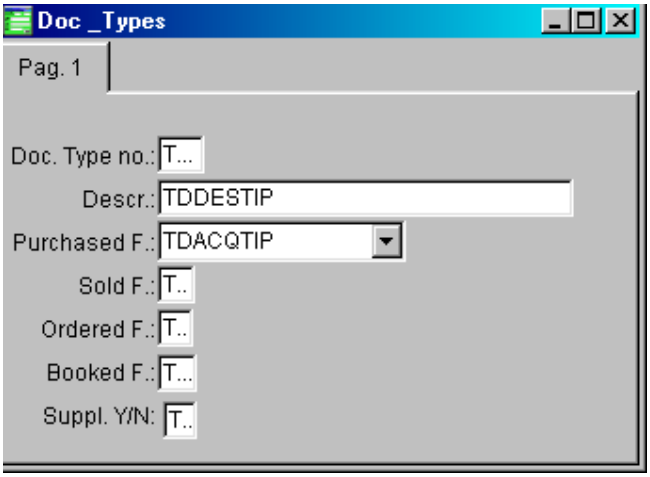
Picture 85 - 'Document Types' Window



Double click the field 'TDACQTIP', go to the 'Radio/Check Buttons' and select 'ComboBox'. Click the '+' button to add a new row and define the fields 'Value' and 'Label' as defined in the following table:

Value	Label
""	No Transaction
"+"	Add
"-"	Minus
"="	Replace

Go to the 'Special Definitions' sheet and set the 'Display length' to '18'. In the 'Evaluate' area select 'Init', go to the 'Calc/Init/Def.' field, add ' ' as field initialization, and confirm with 'OK'.



Picture 86 - Windows Displaying ComboBoxes

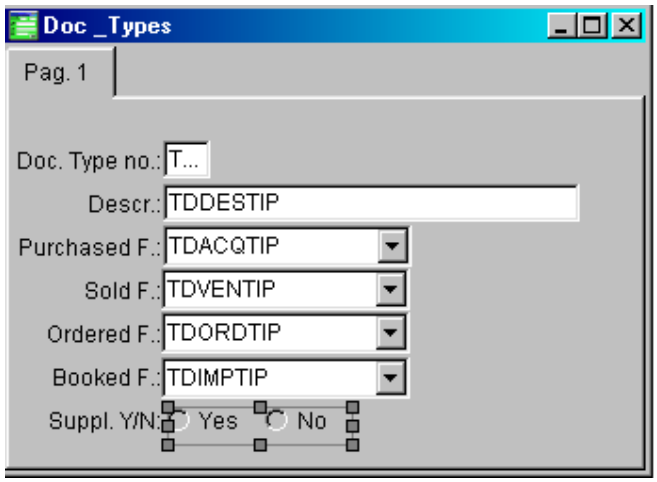
Do the same with the fields 'TDVENTIP', 'TDORDTIP', and 'TDIMPTIP'. Then adjust the window's layout.

Edit the field 'TDFORTIP', go to the 'Radio/Check Buttons' sheet and select 'Radio Horiz.'. Using the '+' button add a new row, and define the fields 'Value' and 'Label as shown in the following table:

Value	Label
"Y"	YES
"N"	NO

Go back to the first sheet, select 'Init' in the 'Evaluate' area, digit 'N' in the 'Calc/Init/Def.' field and confirm with 'OK'. Improve the window's layout aligning fields and descriptions.

Picture 87 - Completed Window



Save the changes and go back to the 'Front-End'.

Open the 'Generation' menu, select 'Codify', to regenerate the code for the various program files.

The Dialog Window defaults the flags of changed entities; simply click 'OK'. Once the regeneration is completed check that the changes you made are the changes you meant to make.

5.3 Documentation And Application Online Help

CODEPAINTER REVOLUTION not only makes software application development easy, but also enables you to quickly produce thorough documentation, allowing to write notes during the various development phases. Notes can be subdivided into two major groups: **Design** Notes and **Codify** Notes. Codify Notes are subdivided in **User** Notes, which are also used to produce the application's on-line help and **Technical Notes.**

Design Notes are technical descriptions used to document the project's logic and are typically used by software developers.

Technical Notes are detailed Design Notes that explain functional controls used at interface level. Technical notes could be the explanation of a calculation and of the implemented formula, or the explanation of why a field is 'Obligatory', etc.

User Notes generally contain descriptions writeten during the Codify phase. These descriptions must be detailed and accurate as they will form the user reference guides and the application's on-line help explaining the application's logic and functionality to the end user.

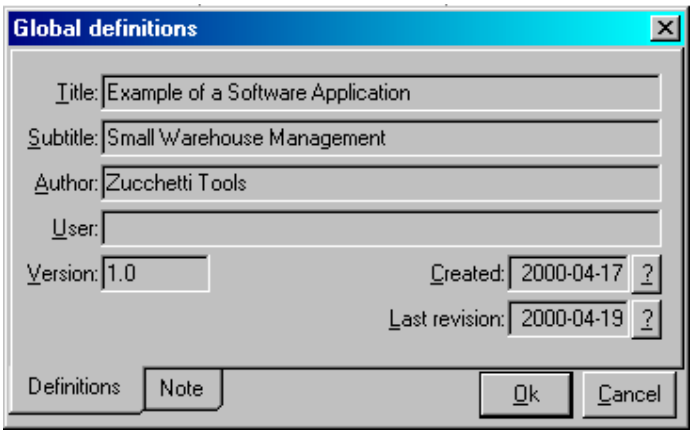
During the Documentation Generation phase, CodePainter structures the written notes homogenously and produces XML files. Using the Documentation Painter these files can be organized differently, giving the desidered structure to the finished user reference guides.

5.3.1 Design Notes And Documentation

Let us now add some notes at Design Level in order to produce the Design documentation.

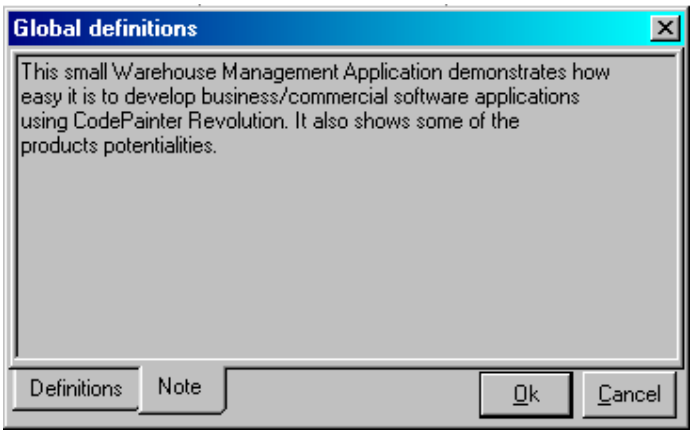
First of all you may wish to write general notes on the application. Open your design plan, open the 'Global' menu, select 'Definitions' and fill in the fields following the following picture.

Picture 88 - Software Application Title



Go to the 'Notes' sheet and write a brief description of the application.

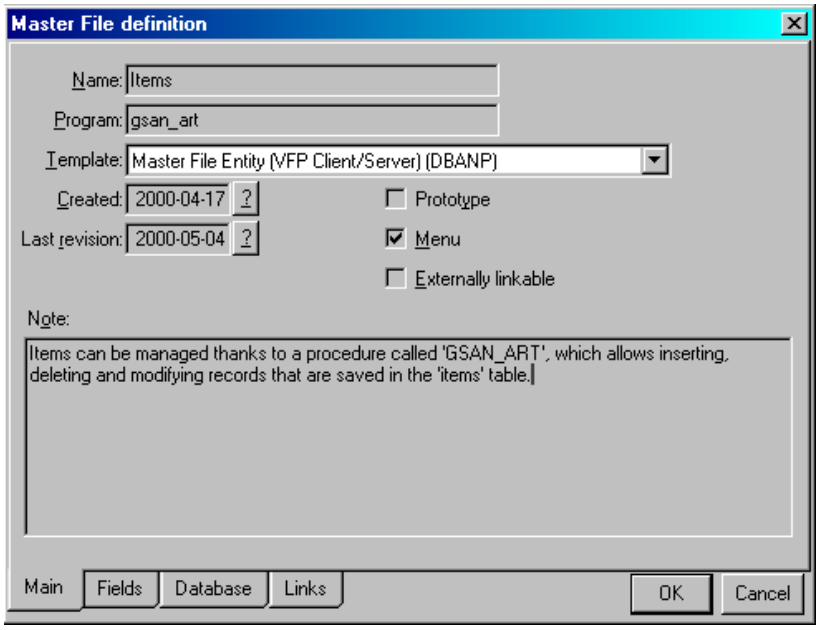
Picture 89 - A Brief Description



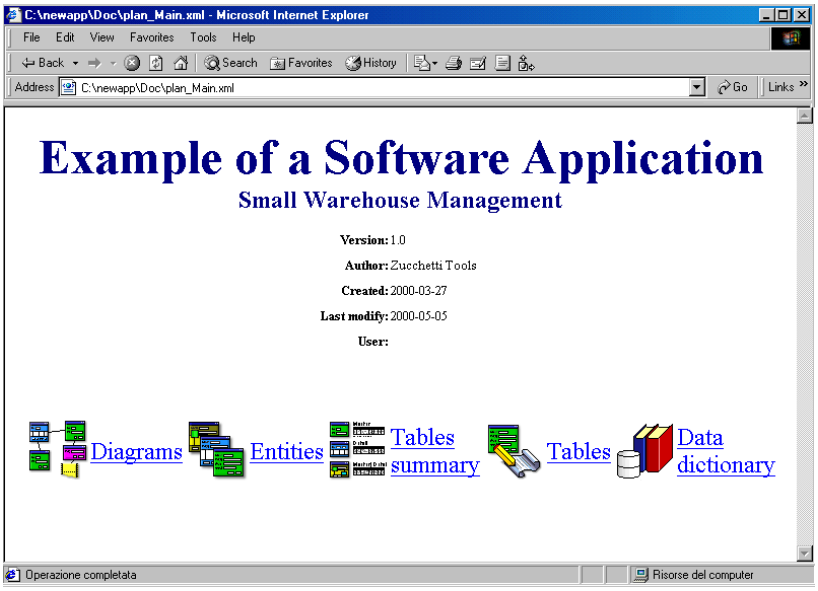
Confirm with 'OK' and then edit the 'Items' entity.

In the 'Notes' section of the 'Main' sheet add notes following the next picture.

Picture 90 - Notes on 'Items'



Following the same principle you can now add notes against single fields and on defined relationships. This can be done for all entities in your plan. Once you have entered all requires notes, save your plan and generate the 'Design' documentation opening the 'Generation' menu, selecting 'Documentation...' and activating the 'Design' flag only. To view the documentation open the 'Project' menu, select 'View Documentation' and 'Design'. Your result should look like the following picture.

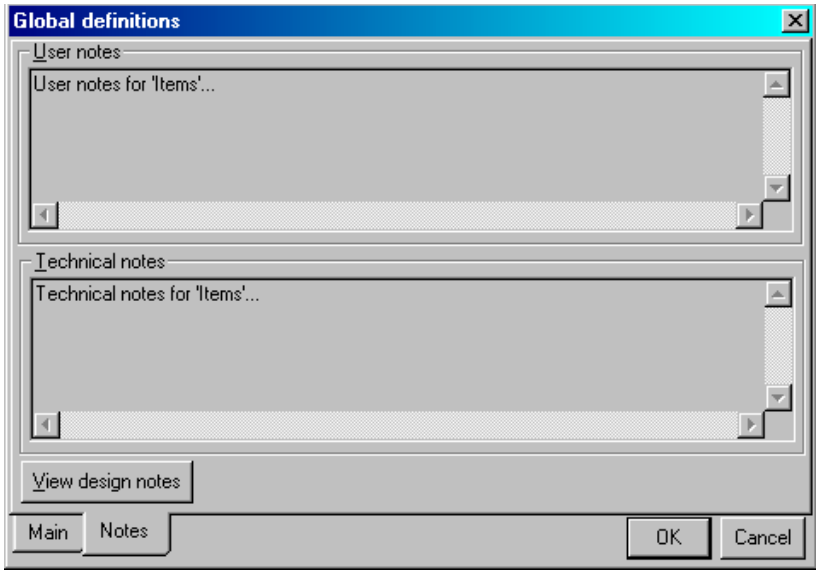


Picture 91 - Example of Design Documentation

5.3.2 Codify Notes And Documentation

Let us now add some notes at Codify Level in order to produce User and Technical documentation.

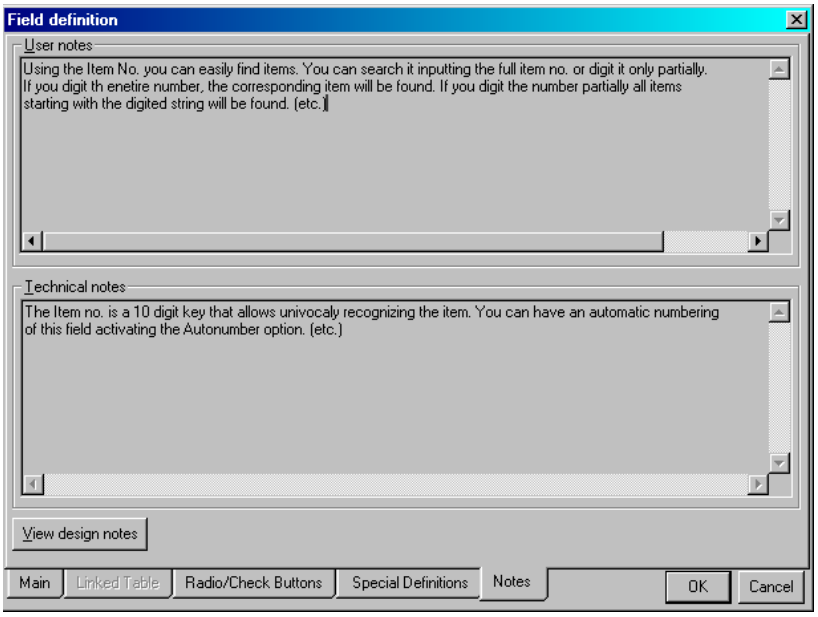
Using the Master File Painter open 'Items'. Open the 'Globals' menu and select 'Global...'. The 'Global Definitions' window opens up. Go to the 'Notes' sheet. You can write both, 'User Notes' and 'Technical Notes'. Clicking the 'View Design Notes' button you can also view the notes you have entered at Design level. You can copy these latter notes in the 'Technical Notes' section and integrate the text.



Picture 92 - User And Technical Notes

Confirm your notes with 'OK'.

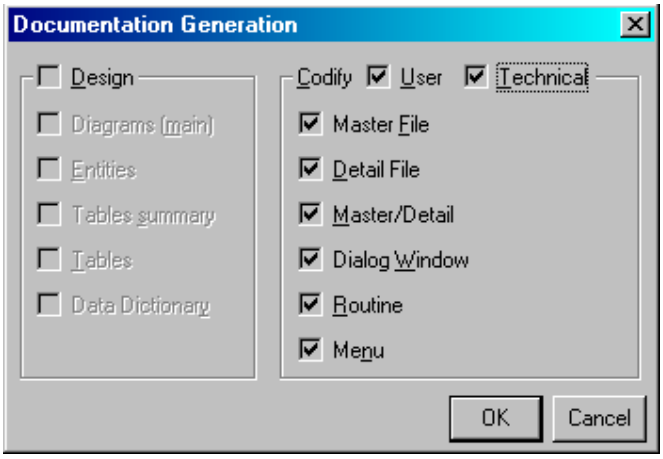
Edit the notes section for the field 'ARCODART". Enter 'User' and "Technical Notes' following the example given in the next picture, then confirm with 'OK'.



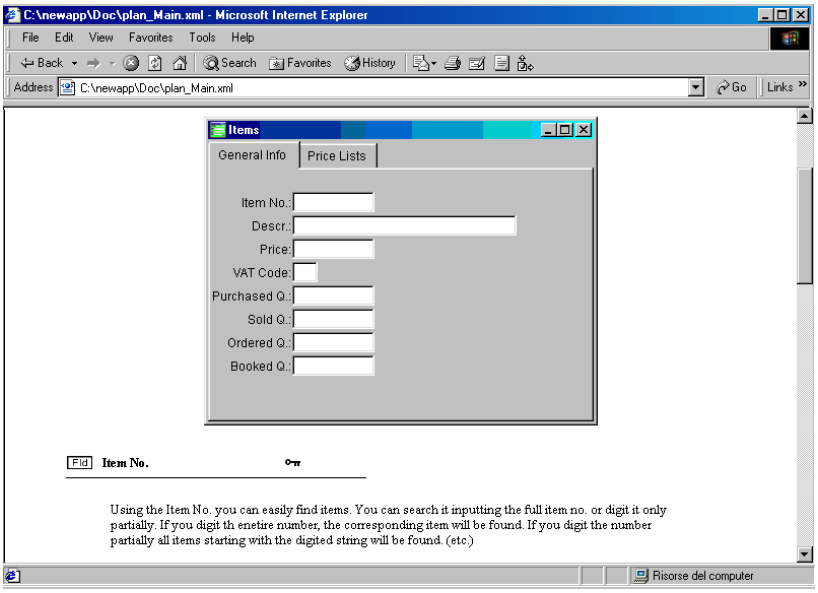
Picture 93 - User And Technical Notes For 'Item No.'

Following the same principle you can now add notes against single fields and variables. This can be done for all entities in your plan. Don't forget to save the changes. Once you have entered all required notes go to the Front End, open the 'Generation' menu, select 'Documentation', activate the flags on 'User' and 'Technical', and confirm with 'OK'.

Picture 94 - 'Documentation Generation' For 'User' and 'Technical Notes'



Open the 'Project' menu, select 'View Documentation' and 'User'. The end result should look like the following picture.



Picture 95 - Example of 'User Documentation

Capitolo 6

User Interface

6.1 User Interface using MS Visual FoxPro

Let us now analyze user interfaces developed using MS Visual FoxPro.

6.1.1 The Menu Bar

The menu bar details the names of the software application programs and the default heading 'Utility' containing basic functionalities, which will be analyzed later on.

To start a procedure click the corresponding menu heading, or press <Alt> and digit the letter underlined in the heading.

The procedure is always opened in the 'Query' mode. You can turn to the 'Load' or 'Change' mode clicking the matching icon or using functional keys.

6.1.2 The Toolbar

The default toolbar is displayed under the menu bar. It can be reduced and moved anywhere on the screen. If you position the mouse on any icon, a tooltip describes the functionality of that particular icon. The same functionalities can be accessed using functional keys.

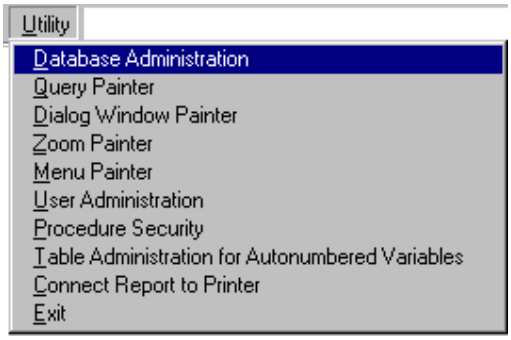
Functional Keys

The following table shows the meaning of icons on the toolbar and depicts the corresponding functional key.

Icon	Function Key	Functionality
Load	F4	To add a new record
Search	Initial status	To read files
Change	F3	To change the selected record
Filter	F12	To search on all table's editable fields
Delete	F5	To delete the selected record
Zoom	F9	To zoom on fields
Next Record	F8	To go to the next record
Previous Record	F7	To go to the previous record
Page Down(PgDn)	PgDn	To go to the next page
Page Up (PgUp)	PgUp	To go to the previous page
Exit (Esc)	Esc	To Exit the program
Delete Row (Detail File Entity)	F6	To delete a row of a body or in a Detail File Entity
Save	F10	To save a record
Display Menu Bar	<CTRL>+T	To treeview the menu bar

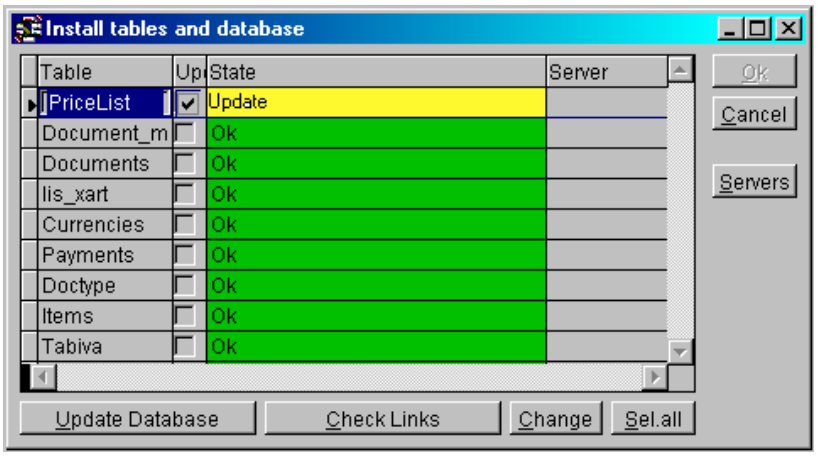
6.1.3 The Utility Menu

The Utility menu is generated automatically with any software application. The next picture shows Utility options.



Database Administration

The 'Database Administration' option audits and aligns when required the database by comparing the database structure and the database design. When the software is run the database is compared with design specifications and if there are discrepancies this function is automatically launched.



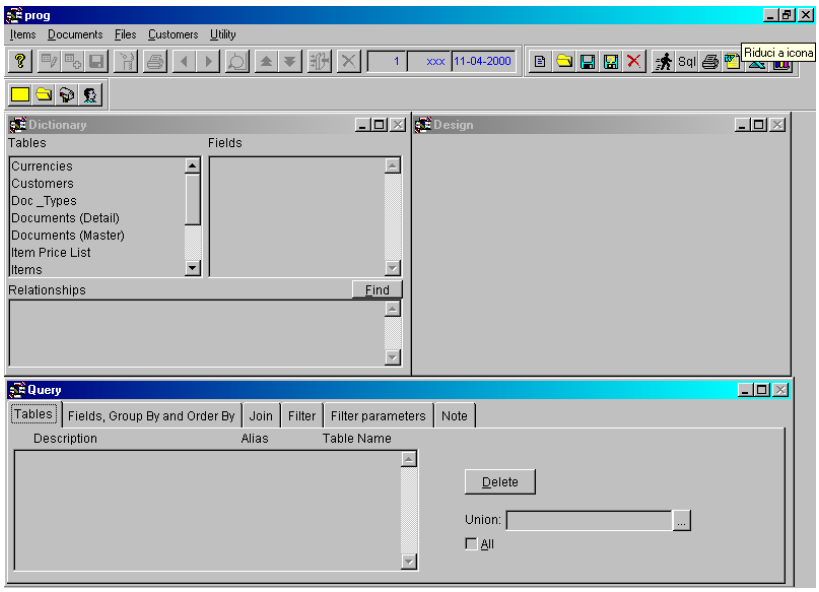
Clicking the 'Update Database' button all tables in the list flagged with 'to update' are audited and aligned. The SQL commands of the running update can be followed in a small dialog window.

Clicking the 'Check Links' button a test procedure for any user defined 'ODBC' link is launched.

Using the 'Change' button the user can manually configure options on single tables. 'Select All' selects all items in the active window.

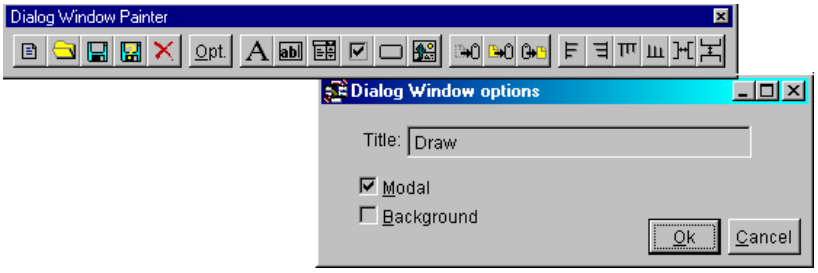
Query Painter

Using Visual Query you can create queries and reports using the SQL Language efficiently. You can define multifile queries, which can be recalled and used by batch procedures, Visual Zooms or reports.



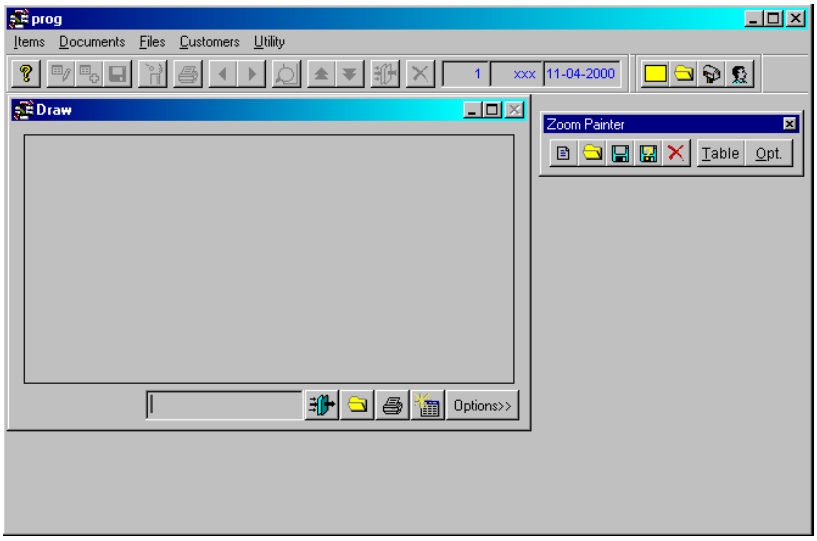
Dialog Window Painter

The 'Dialog Window Painter' integrates the Visual Query functionality, exploiting the object technology, thus allowing to create interpreted dialog windows. Objects created using this tool can be integrated in a query or in a batch in order to define search criteria or to support procedure launches.



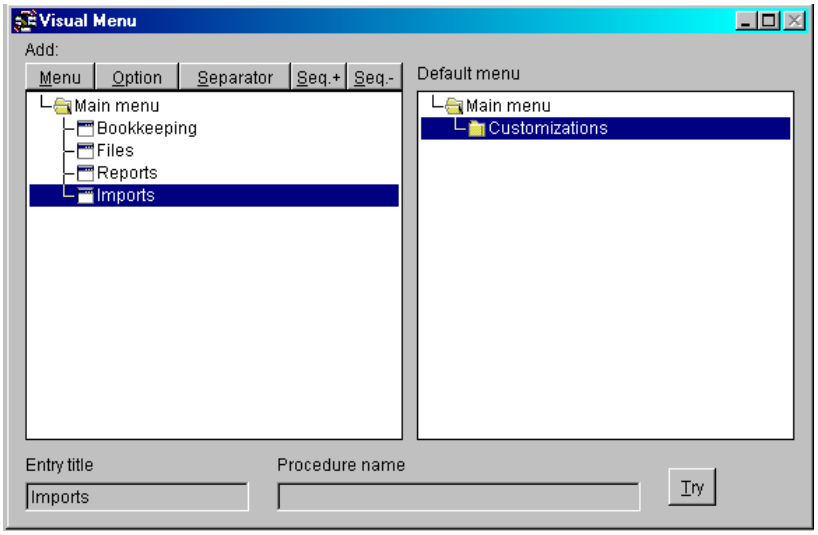
Zoom Painter

Visual Zoom integrates the Visual Query tool allowing to create interpreted zooms on multifile queries. Visual Zooms can be build in a CodePainter application and can be launched from an object button or from the menu.



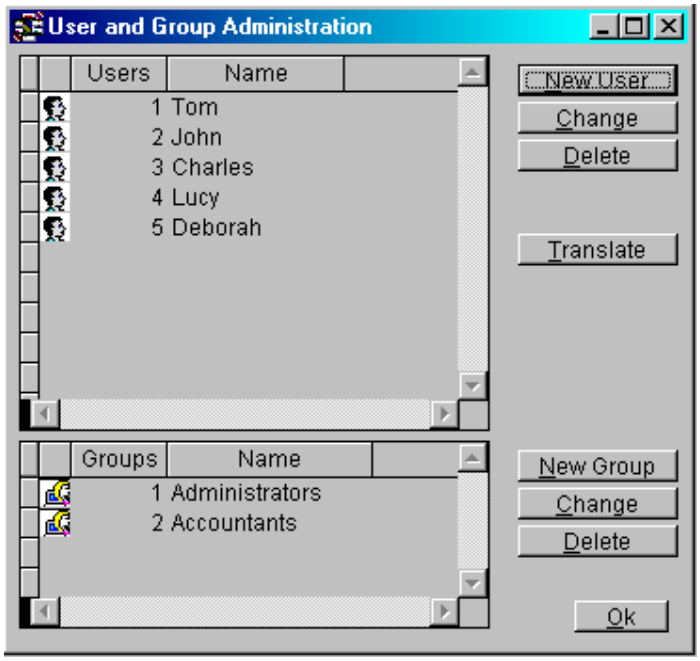
Menu Painter

In the Client Server environment, user menus can be customized using the Visual Menu. The menu structure can be 'painted', the default menu defined, the order of menu items can be changed, or access can be limited to certain users or user groups only.



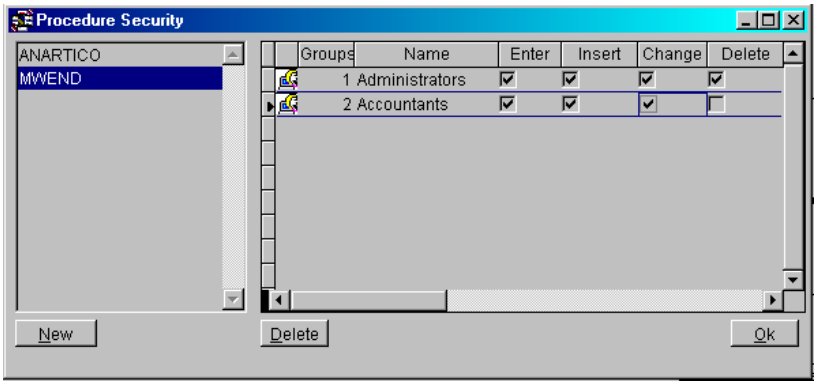
User Administration

The User Administration functionality is at the basis of access control policy, managing the users database, defining user groups to which single users can be assigned, and managing messages between users.



Procedure Security

Using the Procedure Security management, the System Administrator can define which users or user groups can access a single procedure and also access levels (Enter, Insert, Change, Delete).



This functionality is protected by password (default password being 'CodePainter') and can be activated pressing <Alt> and <F12> within the active window. In an autozoom window security levels can be defined for advanced zoom options.

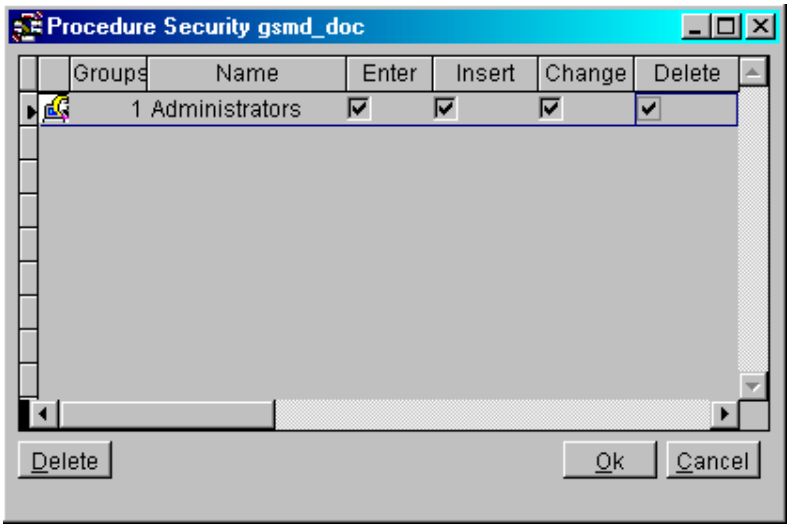
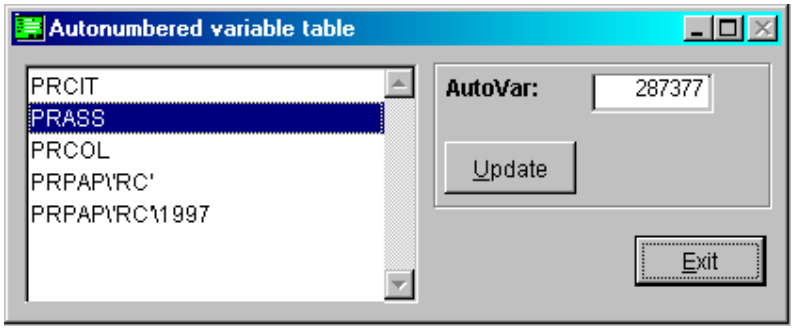


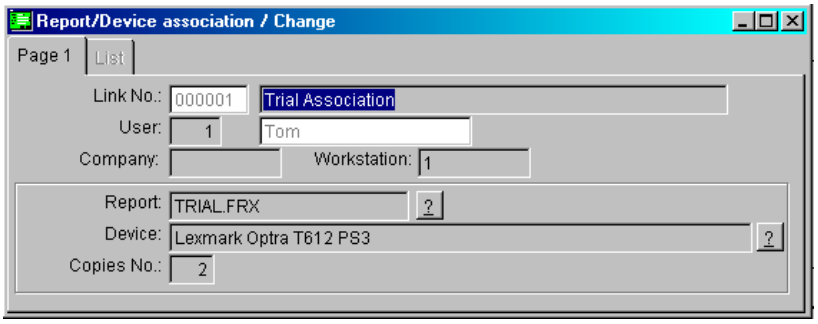
Table Administration for Autonumbered Variables

The 'Progressive Table' option in the 'Utility' menu allows managing accrued progressive numbers. On the left of the dialog window the list of progressive fields defined during development is displayed; on the right the accrued values can be changed.



Connect Report to Printer

With this option you can maintain the file containing connected programs and printers. Adding a record in this file report you can connect a report to a user, to a workstation, to a company or a combination of different options.



6.1.4 The Application Bar and Post-IN

Next to the toolbar, there is the so-called 'Application Bar', which has different user functionalities including Post-IN and application messages. The following table lists the Application Bar functionalities:

Icon	Name	Functionality
 Post-INFCreate a new Post-INigura 3		
 Post-IN FileFOpen user's Post-IN fileigura 4		
 Post BoxFRead in-coming Post-IN messagesigura 5		
 Users/GroupsFManage users and groups and send Post-IN messagesigura 6		

Post-IN Container

The 'Post-IN' functionality is available in the Client/Server version and can be used to take notes, to temporary extend information on records, and to send messages and/or programs to other users. Post-INs can have:

- Text
- Programs, i.e. references to a specific record
- Attachments, i.e. references to an external file

Using Post-IN

Within the application, Post-INs can be used for:

- Notes
- Integrated Warnings
- Non Integrated Warnings
- Messages

Post-IN 'Notes'

Used to take notes, Post-INs can be 'glued' on the desktop or saved in a file.

Post-IN 'Warnings'

Post-INs can be used to add information to a specific record, which is displayed every time the record is opened. To attach a 'Non Integrated Warning' Post-IN simply drag and drop it on the record being in the desired module (e.g. item management). The message 'Warning Added' confirms the action.

Post-IN Attachments

To integrate a Post-IN Warning in a record, press <F3> to define the required properties, or click the 'modify' button on the toolbar. Activating the 'Integrate In Form (warning only)' flag, the Post-IN is added to the record as a new window sheet.

Post-IN Messages

Post-INs can be also used for sending internal messages: on the Application Bar select the 'Users' option and drag and drop the Post-IN on the selected user name. Clicking OK on the new dialog window, the message will be sent to the addressee, who receives it immediately. This does not mean that the addressee can read the received message straight away; indeed the application checks incoming messages every ten minutes. You can otherwise force the checking by clicking the 'Check Mail' button. When there are new messages, the 'Check Mail' button displays a little flag.

Post-IN Folder

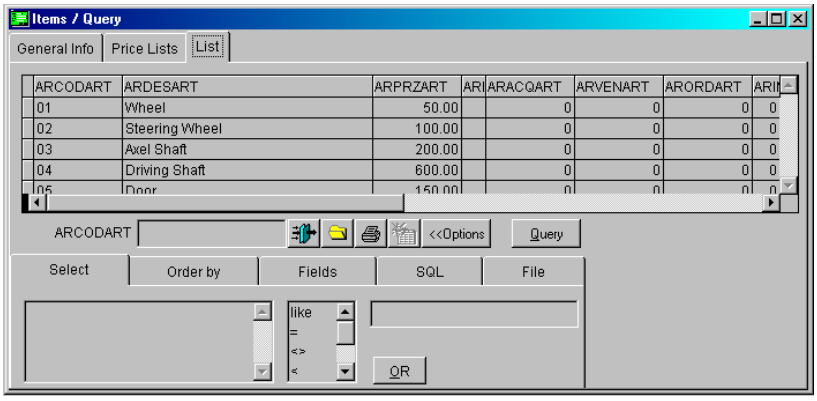
To save a Post-IN you can simply close it clicking the 'X' on the window. Saved Post-INs are in the Post-IN folder. To read or delete Post-INs enter the Post-In Folder: read it double clicking it or delete it clicking the 'Delete' button in the folder. More messages can be selected in one go.

Check Mail

Using the 'Check Mail' button you can access in-coming mail. When a new message arrives, a message pops-up telling you 'Incoming mail ...' and the Check Mail bitmap changes. To read new messages simply click the 'Check Mail' button.

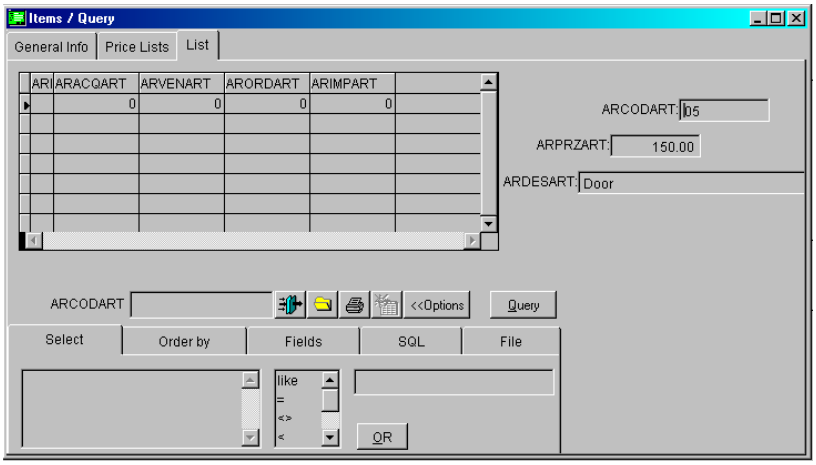
6.1.5 Visual Zoom Configuration

The Visual Zoom is used as autozoom as well as connection zoom. This tool is particularly fast and configurable, allowing to define an infinite number of queries for one file only.



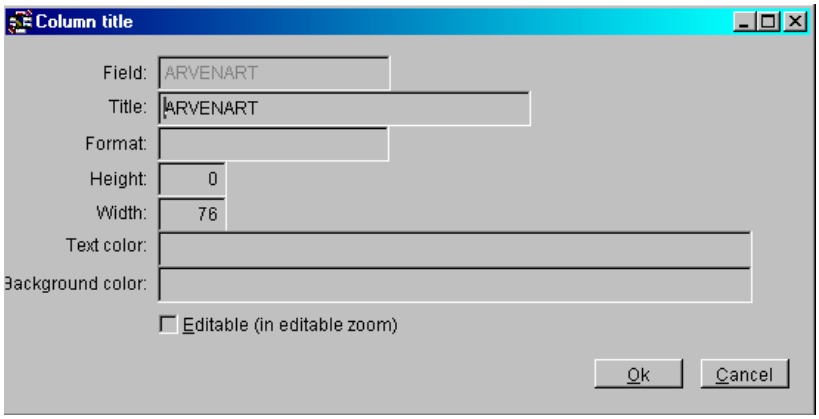
Visual Area

This area is highly configurable and customizable and contains records, which are usually displayed in headed columns. The size of this area can be reset as to leave some space for scrolling fields or for fixed fields (see 'Option Button')



Column Options

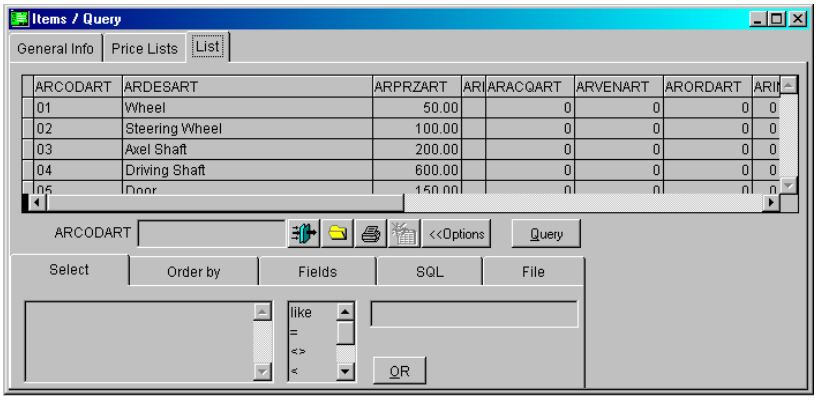
Right clicking the column title, the 'Column Title' window is opened allowing to define column properties for the zoom. The 'Reference Field Name' is not modifiable. The 'Title' field contains the description string, which appears on top of the column or next to the field in case of fixed fields.



In the 'Format' sheet you can introduce the variable's picture using the source code language. The fields 'Height' and 'Width' define the pixel dimension of the field. The fields 'Text Color' and 'Background Color' allow changing the colors in the columns, using the MS Visual FoxPro functionality *RGB()*. It is further possible to add conditions or logical expressions that are accepted by the MS Visual FoxPro language.

Configuration Area

The configuration area is activated selecting the 'Options' button and is used to define all zoom parameters.

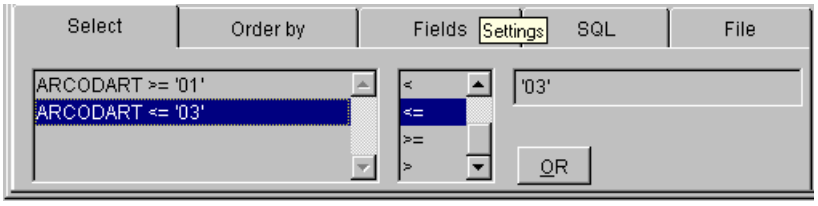


The configuration area is divided in various sheets. The first three ('Selection', 'Order by', and 'Fields') define the SQL sentence characteristics. In the 'SQL' sheet you can display the SQL sentence based on the defined parameters, or select a visual query to be associated to the Zoom. In the 'File' sheet you can save specified parameters in the zoom or create and modify a report.

Changes made to the zoom can be checked anytime using the query mode. Clicking the 'Query' button the query is re-executed basing on set parameters.

Selection Sheet

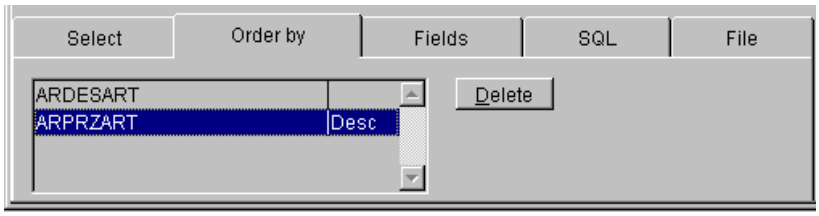
In this area you can define search criteria and/or further selection parameters on records extracted using the Visual Query. You can create complex multi parameter search expressions, thus exploiting comparison operators on the second column and the boolean operator 'OR'.



To select the fields drag the column value in the selection area. The comparison operator is selected double clicking it in the second column. The 'OR' operator is selected from the 'OR' button.

Order By Sheet

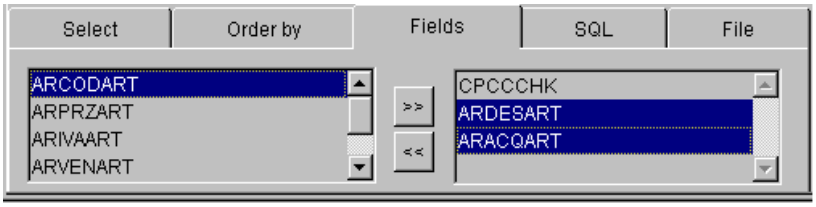
You can define different order types dragging the fields in the 'Order By' sheet. Double clicking these fields you can define ASCENDING (Default) or DESCENDING order.



Selected fields can be removed from the order list clicking the 'Delete' button.

Fields Area

In the 'Fields Area' you define the fields that must be added to the Display Area. On the left all displayable fields are listed. On the right all table fields that are in the zoom are listed. Using the two buttons you can move one or more fields from one side to the other.



SQL Sheet

This area contains the SQL sentence that will be used to extract the data that must be displayed. If you associate a Visual Query to a Zoom you will notice that the term 'WHERE' is replaced by the query name.

With the 'Query' button you can activate the Visual Query selection window (*.VQR). Activating the 'Change Sentence' flag you can type the changes to the SQL sentence obviously using the SQL language.

File Sheet

This area is typically used during configuration in order to save specified parameters in the zoom and to create or change a report.

Setting the 'Default' flag the actual configuration is used by default. Setting the 'Ask For Parameters' flag you need to define selection parameters whenever the Zoom is opened. Setting the 'Keep Dimensions' flag the window size is saved in the configuration file.

Giving a name to the configuration and saving it using the 'Save' button, the configuration file is created using the following syntax:

```
<ConfigurationName><TableName>_VZM
```

N. B.

A Zoom configuration can be saved as default for specific users or groups by saving the zoom file as 'DEFAULT_X' or 'DEFAULT_GX'. The parameter 'X' stands for user and 'GX' for groups.

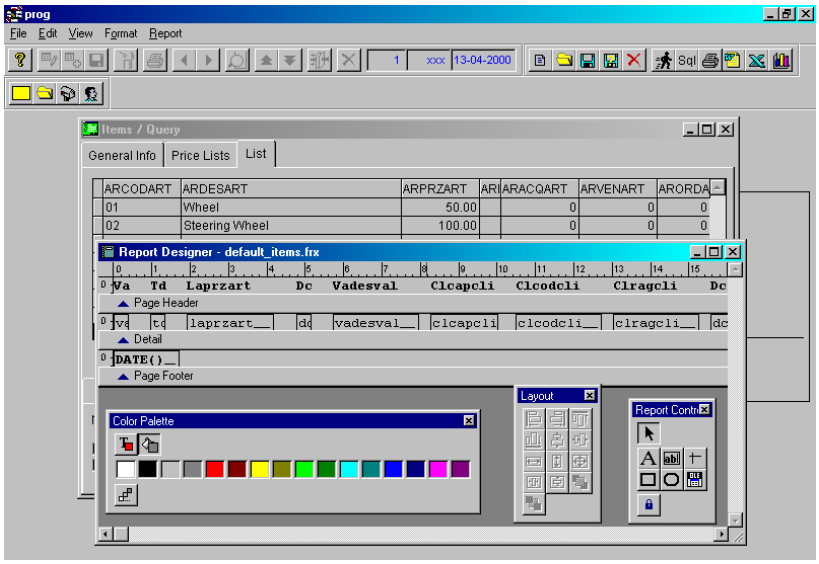
N. B.

The user can create a subdirectory 'Default' under the application' s main directory to save default zooms (in large applications the main directory is 'EXE'). The system will automatically search for any default configuration in the main directory first and in the sub-directory afterwards. When the Zoom is launched by a program you need to set the correct path; e.g. if you want to launch the zoom 'Prova' from the subdirectory 'Zoom' you need to set the path as 'Zoom/prova.VZM' so that the system can find the file.

The 'Modify/Create Report' button launches the MS Visual FoxPro 5.x Report Tool associating a report to the zoom. The report name is created according to the following syntax:

```
<ConfigurationName><TableName>.FRX
```

The associated report can be recalled clicking the 'Execute Report' button in the 'User Area' (see 'User Area').



The 'Modify/ Create MS Word' button launches MSWord creating a model, namely <ConfigurationName><TableName>.DOC, in which you can merge extracted data. The page set-up of the model can be changed to meet your requirements (e.g. you can print labels, envelopes, pre formatted documents, etc.). The support database for mail merge (__WORD__.DBF and FPT) is created in Windows' temporary directory (usually TEMP=C:\WINDOWS\TEMP).

User Area

The 'User Area' is made of a set of buttons to interact with the zoom.

Search (Ask) Parameters

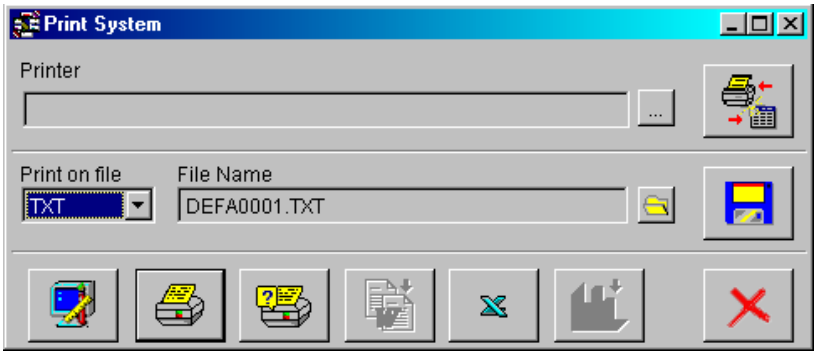
This button opens the 'Search Parameters' window created basing on the specifications defined in the 'Selection' area.

Configuration

Allows selecting configurations.

Execute Report

This button opens the Device Selection window on which the report is executed.

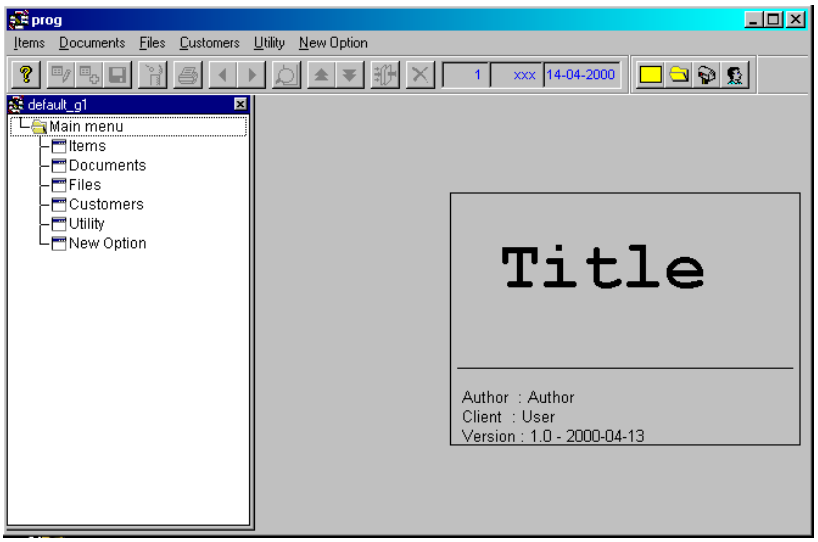


Re-execute Query

Left clicking this button the last query is re-executed. Right clicking it clears the parameters set in the 'Configuration Area'.

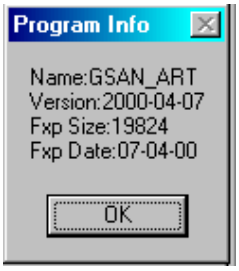
6.1.6 Using the Treeview for The Menu

The menu can be also be treeviewed pressing <CTRL> and <T> simultaneously.



6.1.7 Program Name, Version, Dimension and Date

Information on the active program can be displayed pressing Alt and F11 simultaneously.

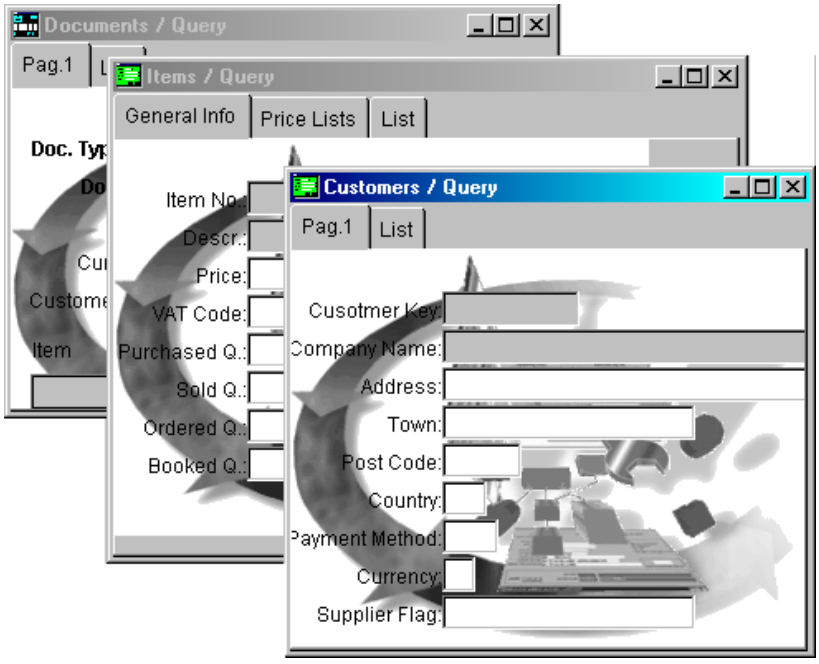


6.1.8 Change The Application Background Bitmap

You can change the picture on the application background creating or copying a new bitmap named 'PROG.BMP'into the main application directory (or in the EXE directory for large applications) .

6.1.9 Dialog Window Background

Associating a background to objects is almost automatic: all you need is to create a BMP directory under your application directory (or under EXE for large applications) and copy all the files that must be used as background. To have all dialog windows with the same background you must save the image in the sub-directory using the name 'DEFAULT.BMP'.



To have dialog windows with different images, you need to save the image in the subdirectory as <ProcedureName>.BMP.

Example

To have the Item Dialog Window (GEST_ART.PRG) with the background file MIA_IMM.BMP you need to create the subdirectory BMP and save the file MIA_IMM.BMP as **GEST_ART.BMP**

Capitolo 7

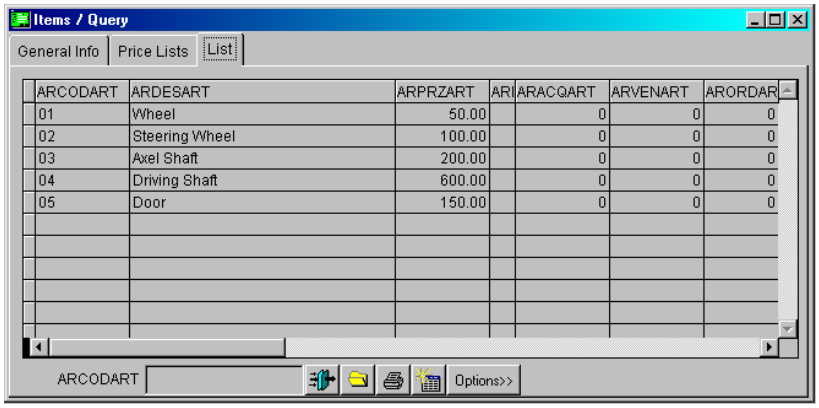
Visual Zoom

Your project is now terminated and you can define Visual Zooms. Open MS Visual FoxPro 5.0 and start your application (CP3START.PRG).

7.1 Visual Zoom Configuration using MS Visual FoxPro

Click 'Items' to open the dialog window. Go to the 'List' sheet to display the default zoom.

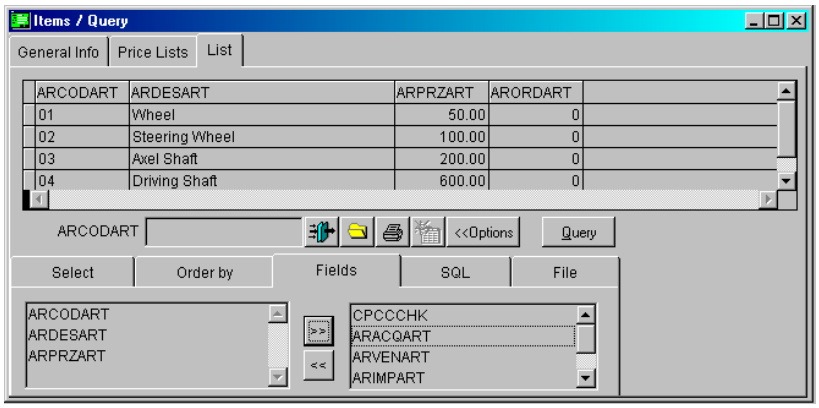
Picture 96 - Default Visual Zoom



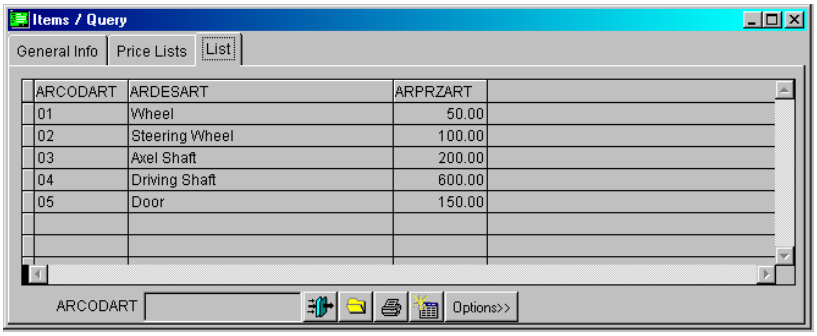
Let us change the default zoom so as to display only the fields 'Code' (ARCODART), 'Description' (ARDESART) and 'Price' (ARPRZART).

Click the 'Option' button and go to the 'Fields' sheet. Select the field 'CPCCCHK' in the left column and move it to the right clicking the button with the double arrow pointing to the right. Do the same with the following fields: 'ARACQART', 'ARVENART', 'ARORDART', 'ARIMPART', and 'ARIVAART'.

Picture 97 - Field Columns

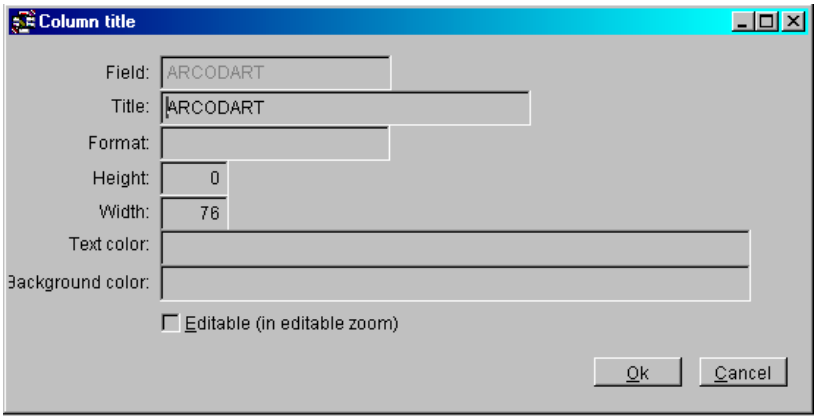


Click the 'Query' button first and the 'Options' button afterwards and check your query.



Picture 98 - First Changes

To change the column title right click the title 'ARCODART". A new dialog window is opened in which you can set column properties.



Picture 99 - Column Properties Dialog Window

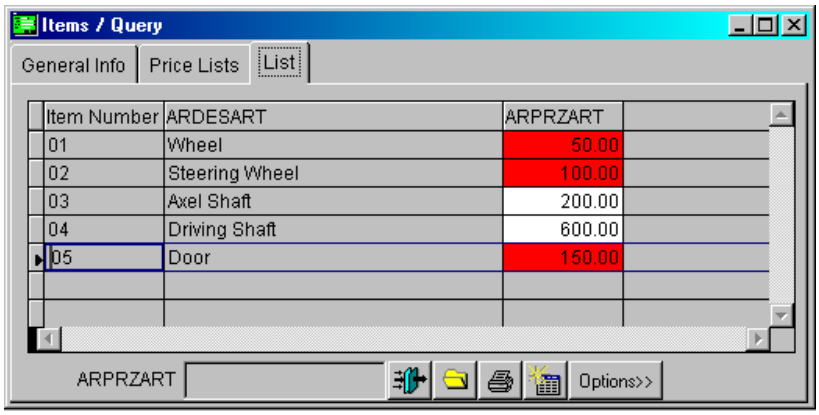
You can now change field name, column title, picture display format, column height and width, text and background colors.

Example

To highlight the 'Item Number' column with red, right click the column title ('ARCODART') and write 'RGB(255,0,0) next to 'Text color'. Click OK to confirm.

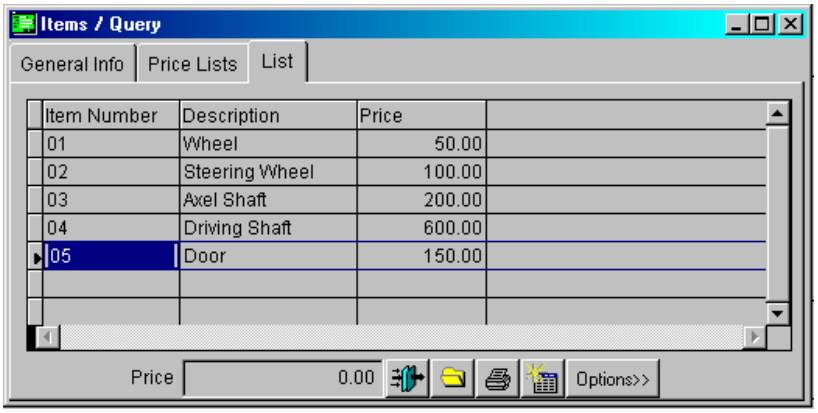
To change the column color for prices under 200, right click the 'ARPRZART' column title and write the expression 'iif(ARPRZART<200000,rgb(255,0,0),rgb(255,255,255))' next to the field 'Background Color'. Confirming with 'OK' your 'Item' dialog widow should look like the following picture.

Picture 100 - Changed Visual Zoom



To set default colors back simply delete the expressions written into the dialog window fields and click 'OK'.

Define the 'Title' field as 'Item Number' and confirm with 'OK'. Do the same for the columns 'ARDESART' and 'ARPRZART' defining them as 'Description' and 'Price'. Resize the column width manually pointing the mouse on the line that divides two columns, clicking and dragging it until the new titles fit properly. Your window should now look like the following picture.



Picture 101 - More Changes

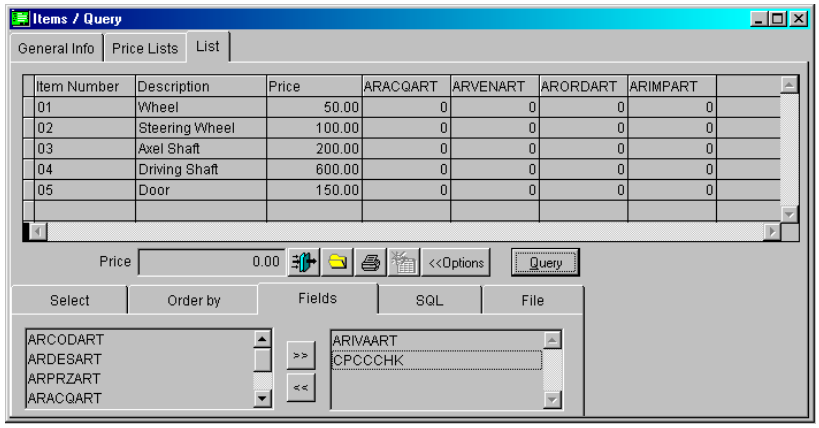
To save the changes click the 'Options' button, go to the 'File' sheet, flag 'Keep Dimensions' and 'Default' and click 'Save'. Go back to the previous screen clicking the 'Options' button.

N. B.

You can save and default a Visual Zoom configuration for particular users or groups saving the zoom file as 'DEFAULT_X' or 'DEFAULT_GX' where 'X' stands for 'user' and 'GX' stands for 'group'.

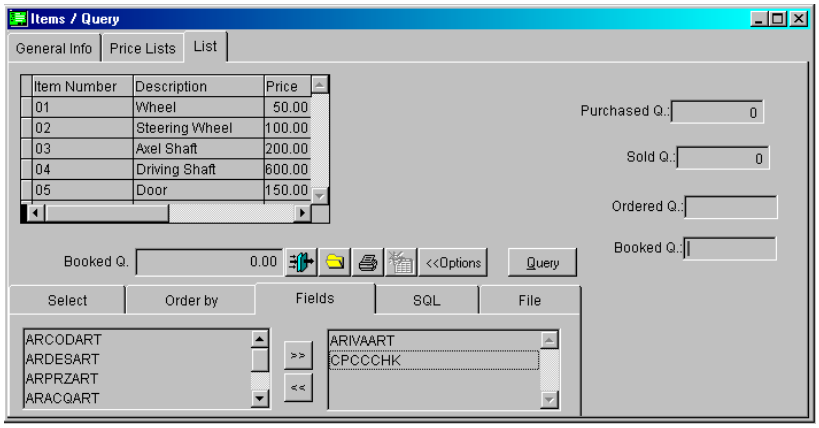
To create another configuration for totals, you need to display the same fields plus the ones containing quantities, namely: purchased (ARACQART), sold (ARVENART), ordered (ARORDART) and booked (ARIMPART) quantity. Click the 'Options' button and go to the 'Fields' sheet. Go to the right window and move the fields 'ARACQART', 'ARVENART', 'ARORDART', and 'ARIMPART' to the left window. Click the 'Query' button to obtain the following window:

Picture 102 - Added Fields



Right click the columns and change the column titles, i.e. replace 'ARACQART" with 'Purchased', 'ARVENART" with 'Sold', 'ARORDART" with 'Ordered' and 'ARIMPART" with 'Booked'. Resize the window left clicking the grey area where scroll bars cross and drag it to the left. Drag & drop the four fields in the area you just created. The fields will be 'fixed' in the area as shown in the following picture.

Picture 103 - Moved Fields



Save the new Zoom going to the 'File' sheet, deselecting the 'default' flag (the 'Keep Dimension' field must be flagged) and naming it 'Totals'. Click 'Save' and then 'Options' to go back to the main screen. There are two configurations for the 'Item Zoom'.

To change configuration parameters recall the zoom clicking the 'Folder' button in the Zoom dialog window. You can configure other zooms within the application following the same steps.

You can now create a report containing selected data. Click the 'Options' button, go to the 'File' sheet and click the 'Modify/Create report' button to obtain a report prototype, which you can change, using both, CodePainter and Visual FoxPro tools.

Capitolo 8

Visual Query, Report And Visual Zoom

8.1 Introduction

Applications generated with MS Visual FoxPro use an application- integrated tool (Visual Query), in order to create queries for extracting data and pass it on to the MS Visual FoxPro reporting tool. The Visual Query tool by exploiting the SQL language is a powerful multifile query and reporting tool. In Client/Server environments it fully replaces traditional tools, thus offering versatility and independence from database servers.

Using this tool you can define multifile queries, which can be recalled and used within applications in order to obtain custom data extractions (batch procedures), data displays (Visual Zoom) and/ or other advanced reporting functionalities. The system does not produce code that must be compiled or interpreted, but complex SQL sentences, which can be changed or implemented. Applications so become highly flexible and maintainable.

The Visual Query tool is strictly linked with the application's Data Dictionary to guide the user through complex queries, without needing a deep understanding of the software. This correlation makes this powerful system easy-to-use.

8.2 Creating Queries, Reports and Zooms basing on the MS Visual FoxPro Query

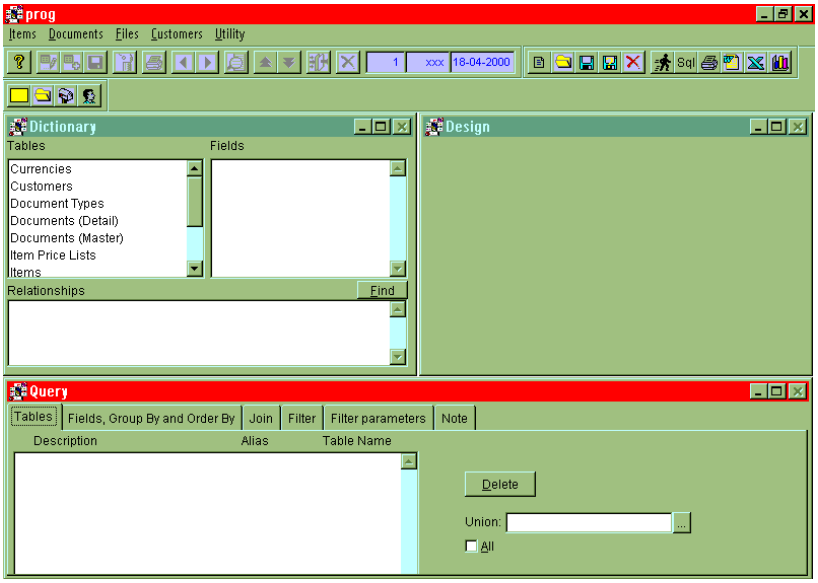
Let us now analyze the various steps requied to create queries, reports and zooms.

8.2.1 Creating a Report

First of all you need to create reports of single documents, which allow printing the current document directly from the 'Documents' window.

The Query

Open the 'Utility' menu and select 'Query Painter'.

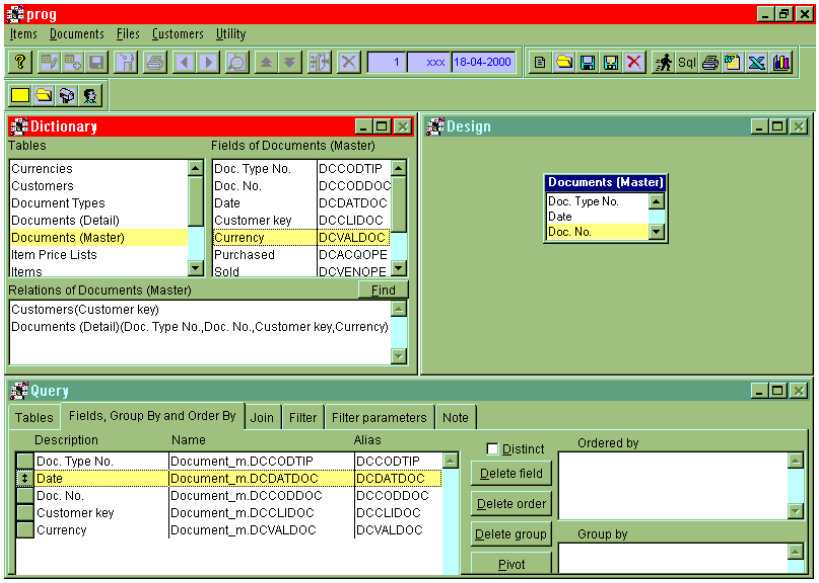


Picture 104 - Query Definition Dialog Window

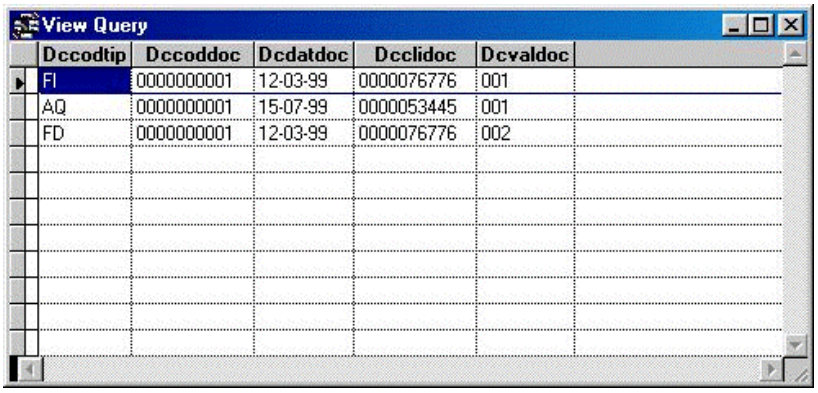
You need to select the files containing the data you want to print. Select the 'Documents Master' file, which contains header and footer of the Master/Detail file 'Documents' and drag it into the 'Design' section.

You can now see the Documents Master fields in the 'Fields' section. Drag the field 'DCCODTIP' into 'Design' so that it is displayed in the 'Documents Master' window. Do the same with the fields 'DCCODDOC', 'DCDATCOC', 'DCCLIDOC' and 'DCVALDOC'. Your window should look like the following picture:

Picture 105 - 'Documents Master' File



Click the 'SQL' button to display the created SQL sentence. Confirm with 'OK' and select the 'Execute Query' button. Using this option you can check anytime whether the query extracts file data correctly or not.



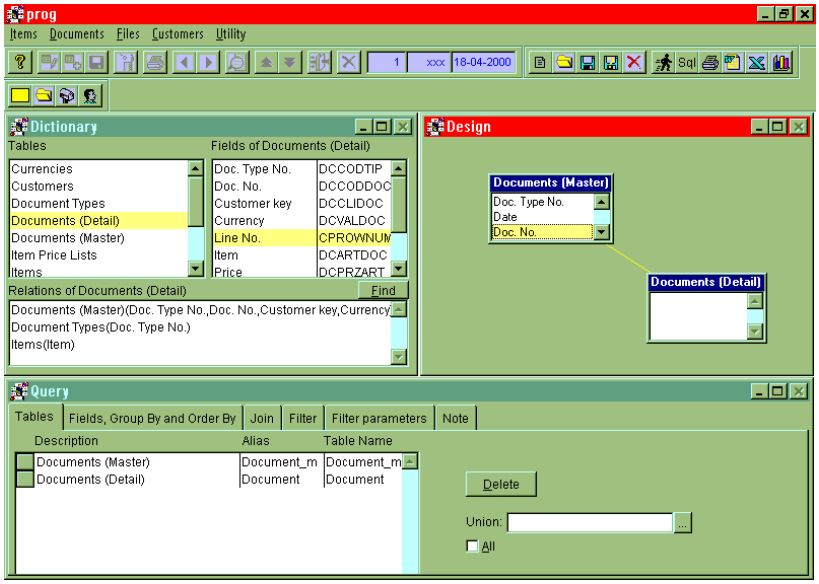
Picture 106 - Query Result

Press 'Enter' to close the window and add the 'Documents Detail' in the 'Design' section.

N. B.

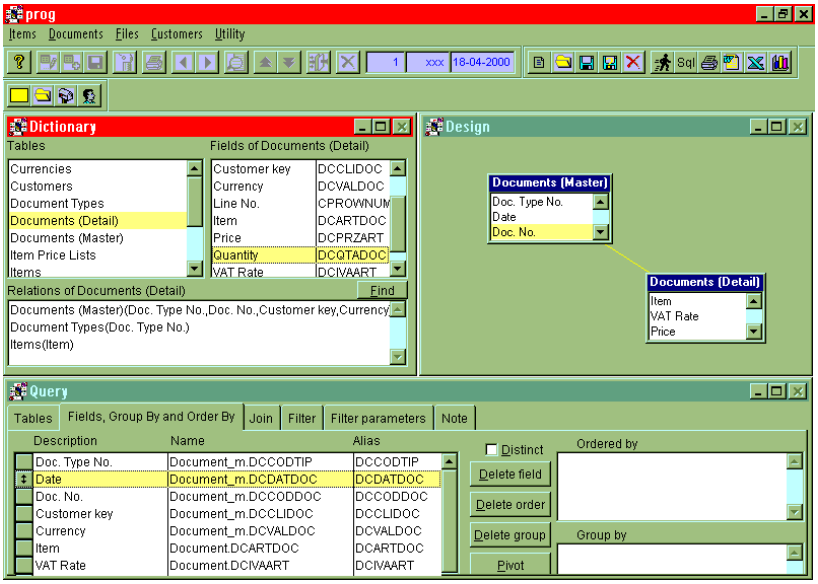
Master/Detail entities defined at 'Design' level are physically made of two files: one containing header and footer fields and the other containing the body fields and the fields building the primary key.

The 'Documents Detail' and the 'Documents Mater' files are related and therefore both must be moved from the 'Relation of Documents Master' section to the 'Design' section. The two files are graphically linked by a yellow line. If you go to the 'Join' sheet of the 'Query' section you can see the 'Join' expression, which defines the relationship between 'Documents Master' and 'Documents Detail'.



Picture 107 - Link between 'Documents Master ' and 'Documents Detail'

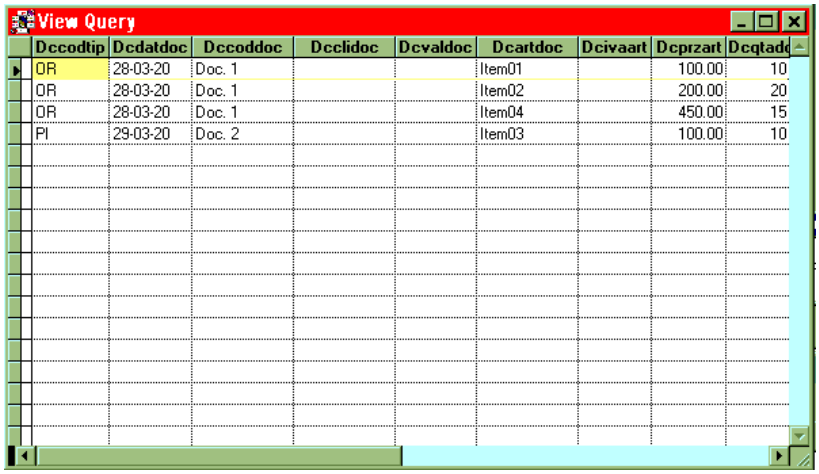
In the 'Fields' section you can now see the Documents Detail fields. Drag the field 'DCARTDOC' in the 'Design' section to display it in the 'Documents Detail' window. Do the same with the fields 'DCIVAART', 'DCPRZART', and 'DCQTADOC'. Your window should look like the following picture:



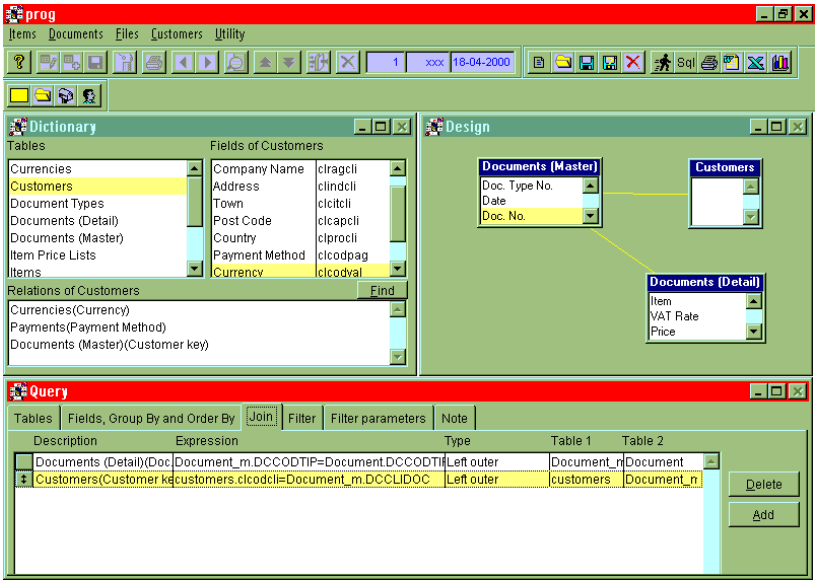
Picture 108 - Query After Adding the 'Documents Detail' File

You can check the result of the query clicking the 'Execute Query' button.

Picture 109 - Query Result

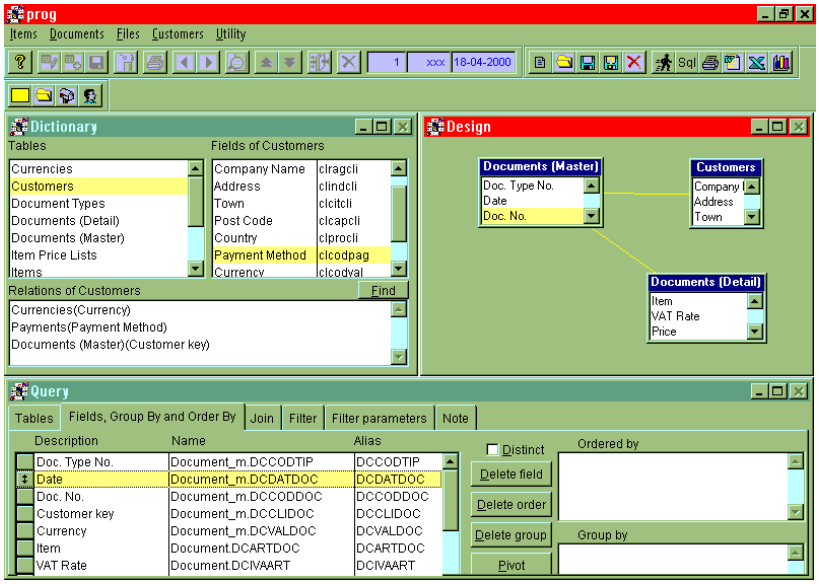


The Query displays the 'Documents' file fields that you want to print. Some decoding values are missing and you need to take them from the corresponding files. Open the 'Documents Master' file and select the 'Design' section. The 'Relations of Documents Master' section displays the list of relationships, which you defined for this file during the Design phase. Move the 'Customers' File from the 'Relations of Documents Master' section to the 'Design' section. The 'Customers' File is linked with the 'Documents Master' by a yellow line. If you go to the 'Join' sheet of the 'Query' section you can see the 'Join' expression, which defines the relationship between the two files.



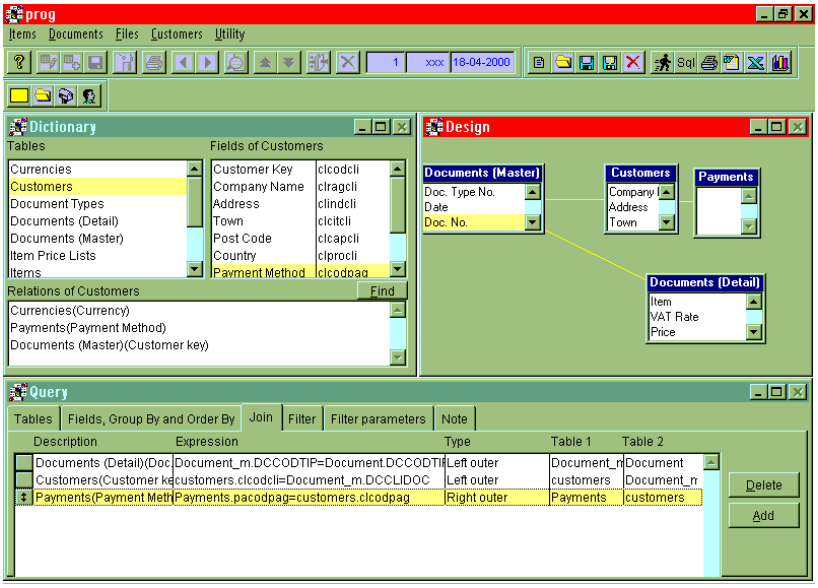
Picture 110 - Link between 'Documents Master ' and 'Customers'

In the 'Fields' section you can now see the 'Customers' fields. Drag the field 'CLRAGCLI' in the 'Design' section to display it in the 'Customers' window. Do the same with the fields 'CLINDCLI', 'CLCITCLI', 'CLCAPCLI', 'CLPROVCLI' and 'CLCODPAG'. Your window should look like the following picture:



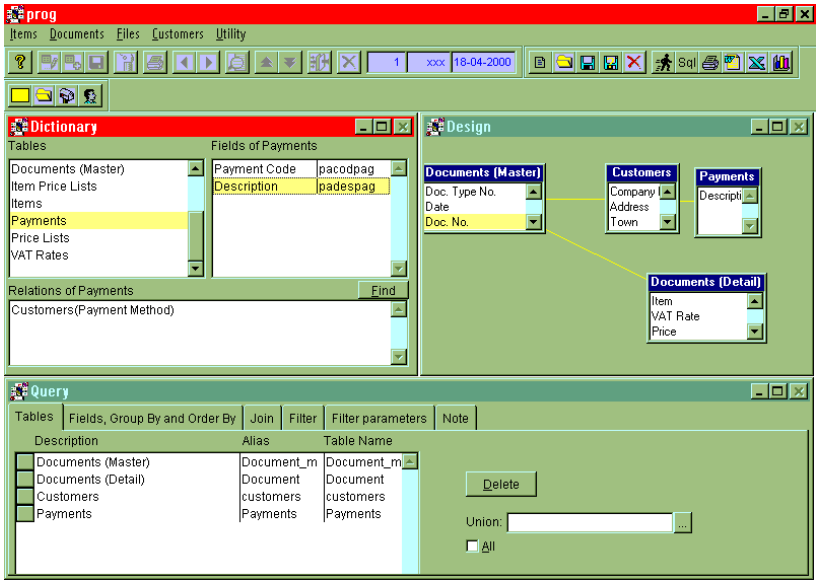
Picture 111 - The Query After Adding the 'Customers' File

You now need to take the 'Payment Method Description' from the 'Payments' file, which is linked with the 'Items' file by the 'Item Number'. In the 'Design' section select the 'Customers' file. In the 'Relations of Customers' section you can now see the list of relationships defined during the Design phase. Move the 'Payments' file from the 'Relations of Customers' section to the 'Design' section.



Picture 112 - Link between 'Customers' and 'Payments'

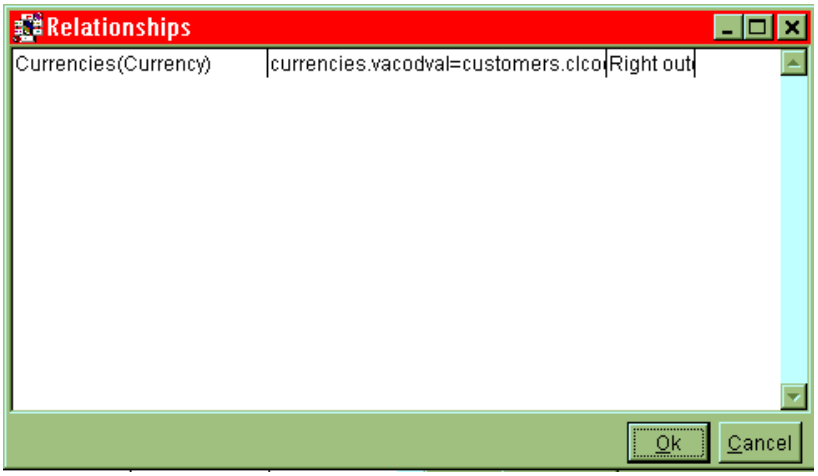
In the 'Fields' section you can now see the 'Payments' fields. Drag the field 'PADESPAG' in the 'Design' section to display it in the 'Customers' window.



Picture 113 - The Query After Adding the 'Payments' File

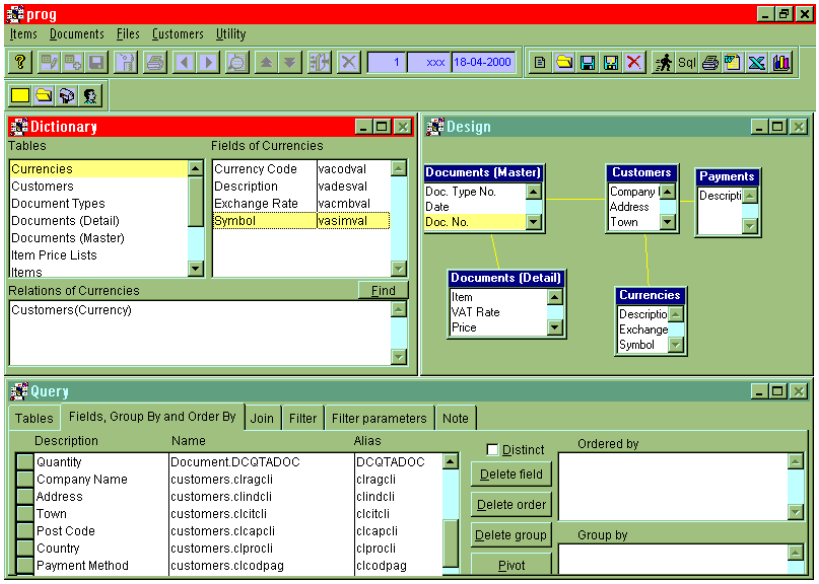
You now need to take currency data from the 'Documents Master' depending on the code defined in the field 'DCVALDOC'. This relationship has not been defined during the Design Phase and you therefore need to manually define the link between the two files.

Move the 'Currencies' file from the 'Dictionary - Tables' section to the 'Design' section. The relationship window between 'Customers' and 'Currencies' opens up. Click 'Cancel' because in this case you need to select the currency basing on the currency code defined in the 'Documents' file (and not in the 'Customers' file).



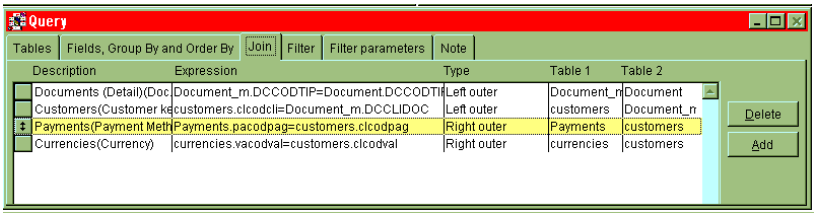
Picture 114 - 'Relations' Window

In the 'Fields' section you can now see the 'Currencies' fields. Drag the field 'VADESVAL' in the 'Design' section to display it in the 'Currencies' window. Do the same with the fields 'VACMBVAL' and 'VASIMVAL'. Your window should look like the following picture:



Picture 115 - The Query After Adding the 'Currency' File

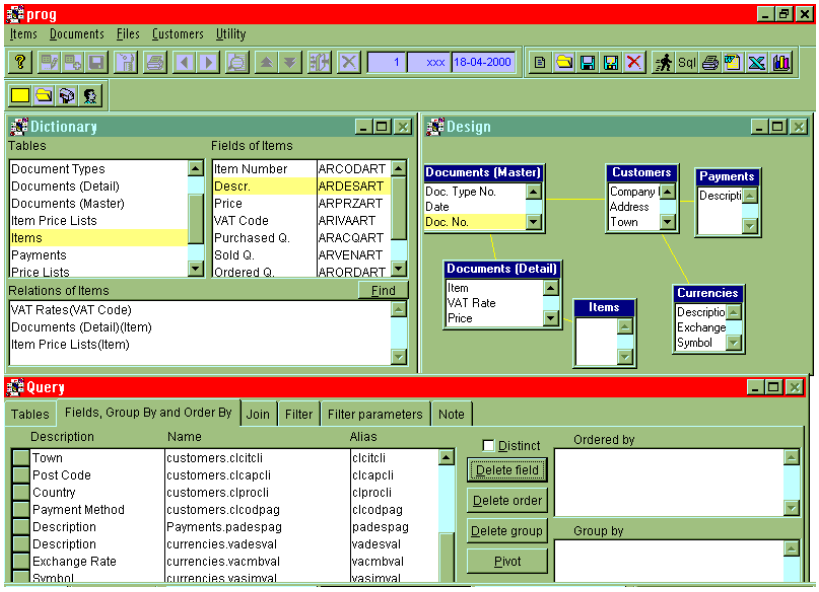
You now need to define the join expression that establishes the relationship between 'Documents Master' and 'Currencies'. In the 'Query' section click the 'Join' sheet. Click the 'Add' button in order to add a new row to the join list. Go to the 'Description' column in the new row and write the string 'Documents Master(Currencies)'. Go to the 'Expression' column and write the string 'docume_m.DCVALDOC = currencies.VACODVAL'. From the combo box in the 'Type' column select 'Left Outer'. From the combo box in the 'Table1' column select 'Docume_m' and from the combo box in the 'Table2' column select 'Currencies'.



Picture 116 - Join definition between 'Documents Master' and 'Currencies'

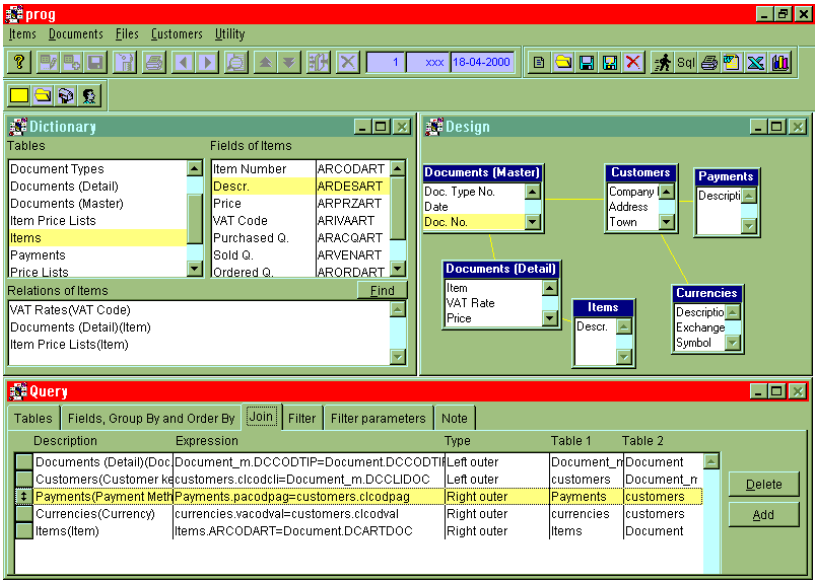
You have defined the relationship between 'Documents Master' and 'Currencies', but in the 'Design' section the two files are not yet linked by a yellow line. To display the yellow line you need to save, close and reopen the query.

You now need to take the 'Item Description' basing on the 'Item Number' defined in the 'Documents Detail' field 'DCARTDOC'. In the 'Design' section select the 'Documents Detail' file. In the 'Relations of Documents Detail' section you can see the relationship list defined during the Design phase. Move the 'Items' file from the 'Relations of Documents Detail' section to the 'Design' section.



Picture 117 - Link between 'Documents Detail' and 'Items'

In the 'Fields' section you can now see the 'Items' fields. Drag the field 'ARDESART' in the 'Design' section to display it in the 'Documents Detail' window.

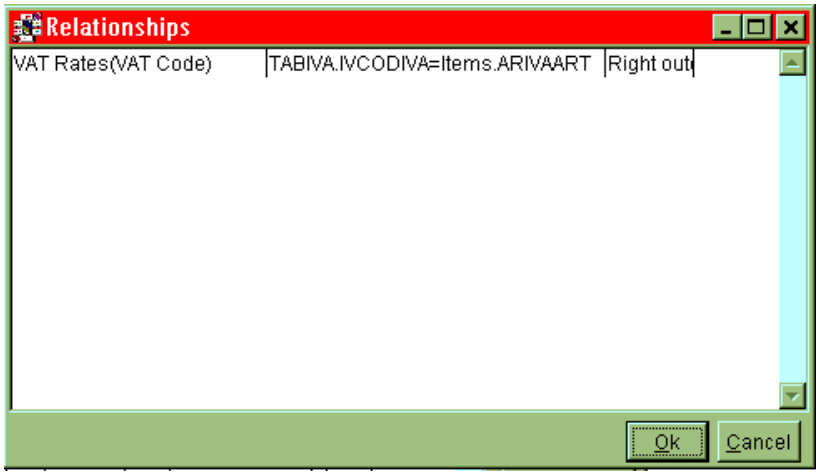


Picture 118 - Query After Adding the 'Items' File

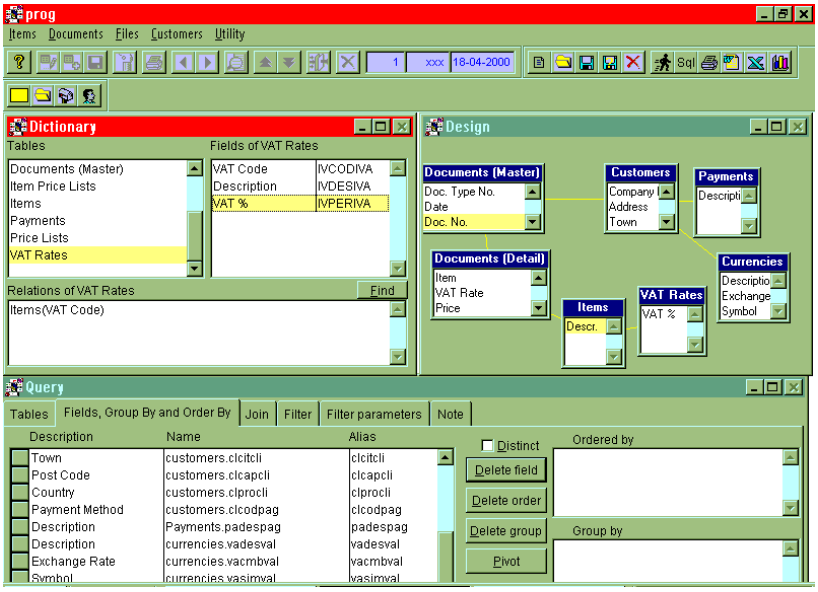
You now need to take the 'VAT Rate' basing on the 'VAT Code' defined in the 'Documents Detail' field 'DCIVART'. This relationship has not been defined during the Design phase and you therefore need to manually define the join expression between the two files.

Move the 'VAT Rates' file from the 'Dictionary - Tables' section to the 'Design' section. The relationship window between 'Items' and 'VAT Rates' opens up. Click 'Cancel' because in this case you need to select the 'VAT Rate' basing on the 'VAT Rate' defined in the 'Documents' file (and not in the 'Items' file).

Picture 119 - 'Relations' Dialog Window

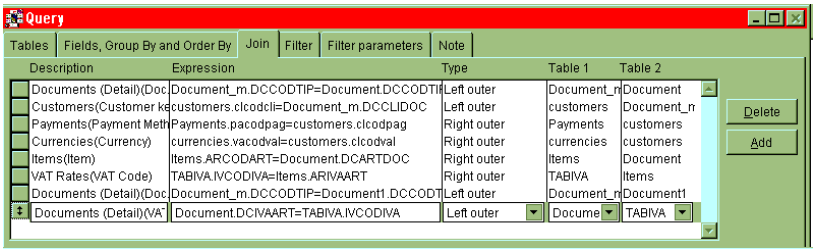


In the 'Fields' section you can see the 'VAT Rates' field. Drag the field 'IVPERIVA' in the 'Design' section to display it in the 'VAT Rates' window.



Picture 120 - The Query After Adding the 'VAT Rates' file

You now need to define the join expression that defines the relationship between 'Documents Detail' and 'VAT Rates'. In the 'Query' section click the 'Join' sheet. In the new page click the 'Add' button to add a new row to the join list. Go to 'Description' column of the new row and write the string 'Documents Details(Rates)'. Go to the 'Expression' column and write the string 'document.DCIVAART = tabiva.IVCODIVA'. From the combo box in the "Type' column select 'Left Outer'. From the combo box in the "Table1' column select 'Document' and from the combo box in the "Table2' column select "Tabiva'.

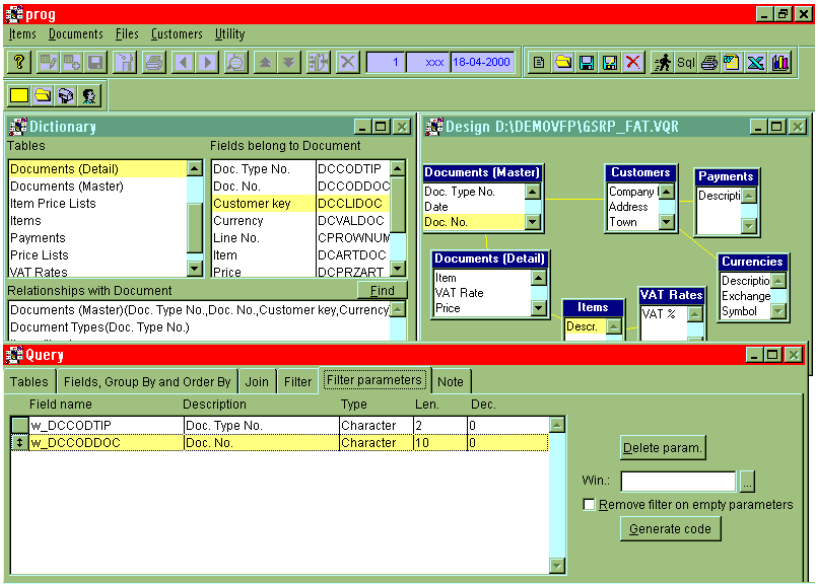


Picture 121 - Join Definition between 'Documents Detail' and 'VAT Rates'

You have defined the relationship between 'Documents Detail' and 'VAT Rates', but in the 'Design' section the two files are not yet linked by a yellow line. To display the yellow line you need to save, close and reopen the query.

You now need to define parameters for the query filter. You need to create the documents report and hence your filter parameters are the two fields building the 'Documents' primary key, namely 'DCCODTIP' and 'DCCODDOC'.

Go to the 'Query' section and select the 'Filter Parameters' sheet. Go to the 'Design' section, select the 'Documents Master' file and move the fields 'DCCODTIP' and 'DCCODDOC' from the 'Fields of Documents Master' section to the 'Filter Parameters' section. Go to 'Filter Parameters' section, change the variable's name 'pDCCODTIP' in 'w_DCCODTIP' and change 'pDCCODDOC' in 'w_DCCODDOC'.



Picture 122 - Defining Filter Parameters

When you launch the query from the 'Documents' window, only the records of that current document will be selected.

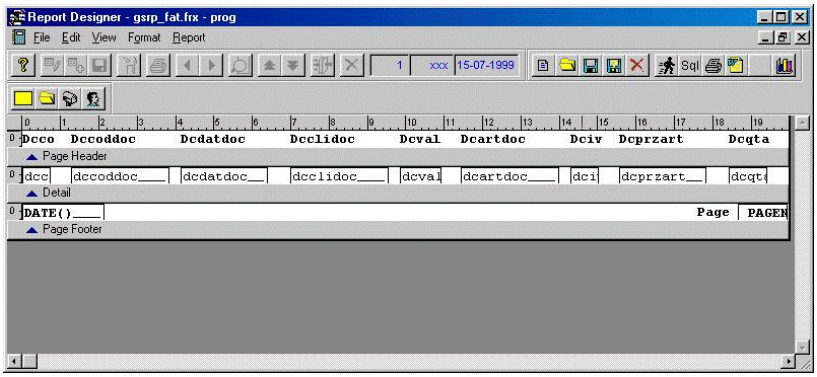
You can now save the query naming it after the file, namely 'GSRP_FAT'.

Creating The Report

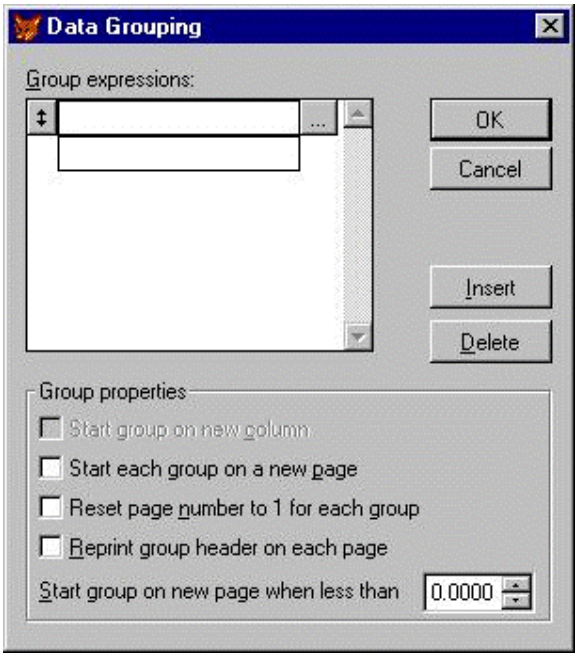
You now have all necessary data to create the report.

Click the 'Create/Modify Report' button. The window for parameters definition is opened. Close it clicking 'Cancel'. The 'Report Designer' is opened, showing a first draft of the report. The report is divided into three sections, namely the 'Page header', the 'Detail', and the 'Footer'.

Picture 123 - The Report Designer



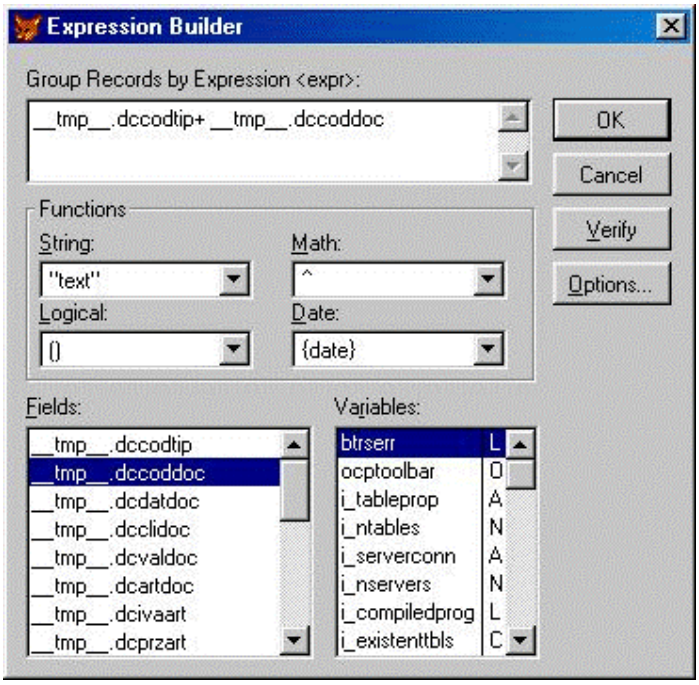
To print documents, in which the body is made of a variable number of elements, you need to group data on the Master/Detail primary key. Open the 'Report' menu and select 'Data Grouping'. Click the '...' button to choose the field on which to create the group.



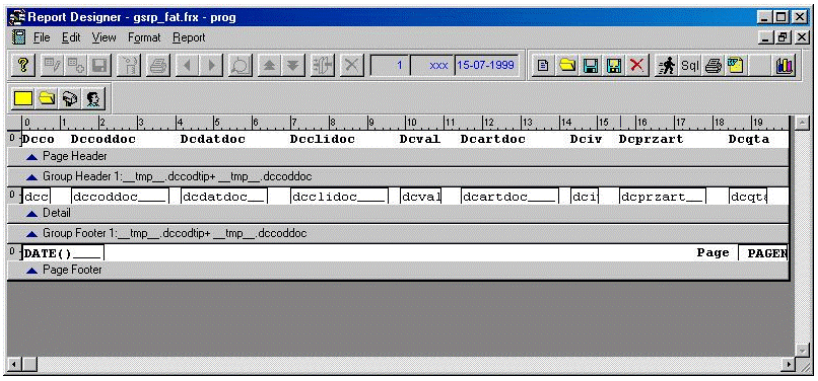
Picture 124 - The 'Data Grouping' Window

Double click '__tmp__.dccodtip' to add the variable in the 'Group Records by Expression <expr>' section. Manually add a '+' after the variable's name and then select '__tmp__.dccoddoc' from the variable list.

Picture 125 - Expression Builder Window

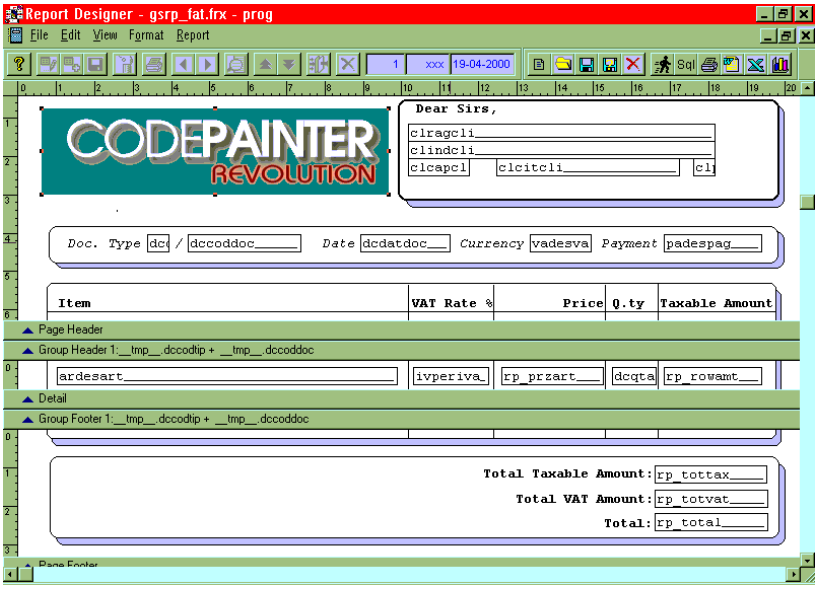


Confirm the 'Expression Builder' window and the following one to obtain the following situation:



Picture 126 - Report Window

You can now improve the report's layout moving fields until you obtain something like what is shown in the following picture.

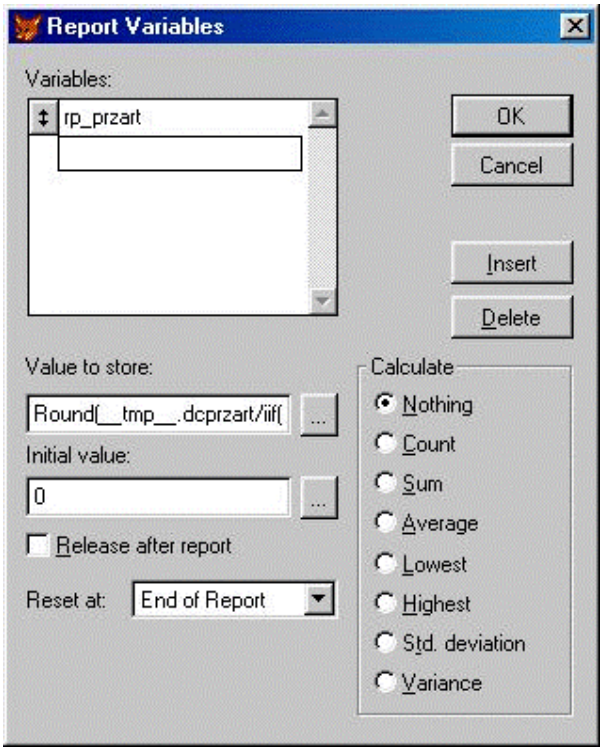


Picture 127 - Completed Report

In the sample report you can notice that the body has two variables containing the row's 'Total Taxable Amount' and the 'Total VAT Amount'. The footer has three variables containing the document totals. The body has also a variable containing the item price calculated basing on the currency. Let us now analyze how a calculated variable can be created.

Open the 'Report' menu and select 'Variables'. Define the variable's name, e.g. 'rp_przart'. In the 'Value to Store' section write the following calculation rule:

'Round(__tmp__.dcprzart/iif(__tmp__.vacmbval<>0,__tmp__.vacmbval,1),2)'. This is the same rule that has been used in the 'Documents' window.



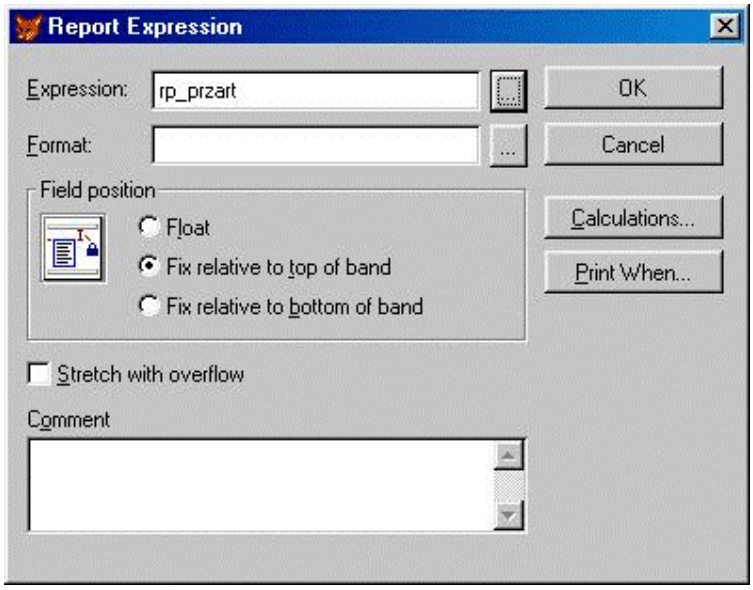
Picture 128 - Report Variables Dialog Window

Confirm with 'OK'. Open the 'Report' menu and select the 'Report Controls' toolbar to add or delete objects from the report.

Picture 129 - Report Controls Toolbar

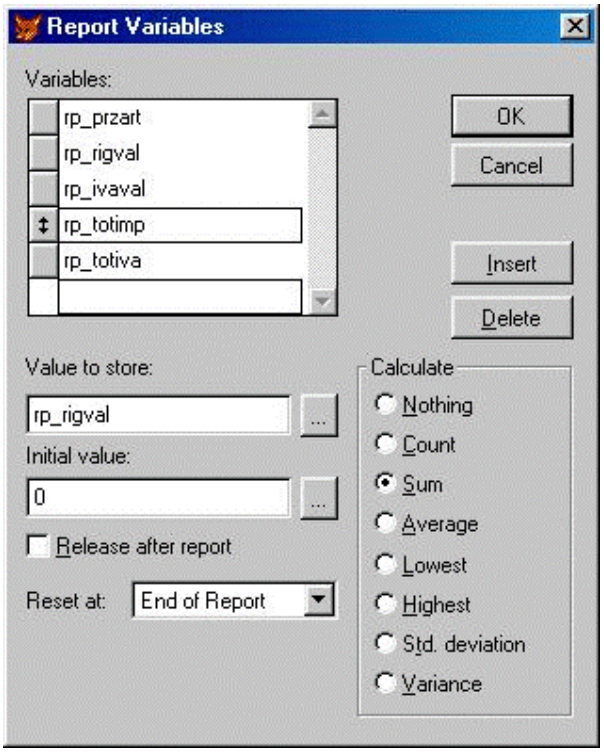
Click the 'Field' button to open the 'Report Expression' window. Click the '...' button next to 'Expression' and select the defined variable from the list.

Picture 130 - 'Report Expression' Dialog Window



Confirm with 'OK' and drag the variable on the report using the mouse.

The 'Totals' variables on the footer are very similar. The only differences lie in the calculation formula, which can be derived from the ones defined in the 'Documents' window, and in the 'Calculate' flag in the 'Report Variables' window that must be set to 'Sum'.



Picture 131 - 'Report Variables' Dialog Window

Concerning the use of MS Visual FoxPro report tools please refer to MS Visual FoxPro user reference guides.

To display your finalized report open the 'View' menu and select 'Preview'. To close it click the 'Close Preview' button on the 'Print Preview' toolbar. Go back to the main menu. Save the query and the report clicking the 'Save' button.

Click the 'Cross' button to exit.

For further information on 'Visual Query' and 'Visual Report' utilities, please refer to the 'Visual Tool Guide', chapter 2 'Visual Query'.

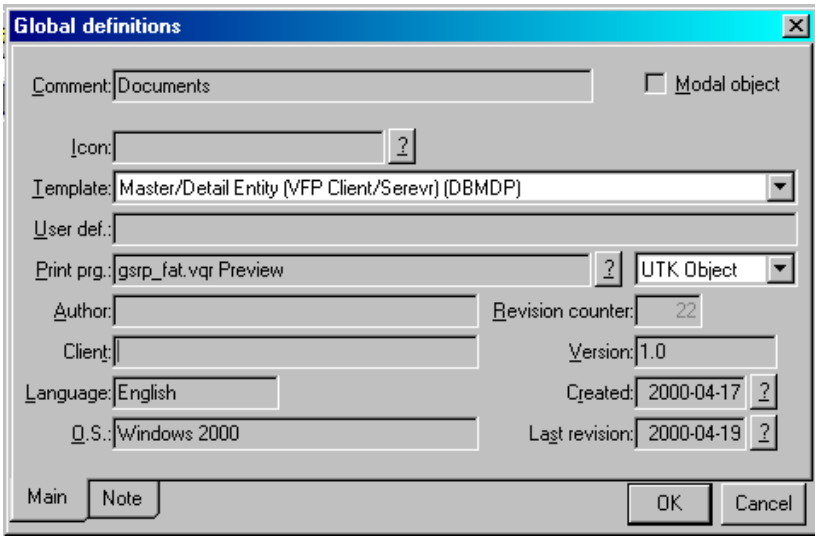
8.2.2 Integrating the report in the application

Let us now integrate the report in the application so that it can be launched from the 'Documents' window pressing 'F2' or clicking the 'Printer' button.

Exit the application and go back to **CODEPAINTER** REVOLUTION. Open the 'Painters' menu and select 'Master/Detail'. Open the 'File' menu and select 'Open'. Open 'gsmd_doc', the only Master/Detail in your plan.

Open the 'Globals' menu and select 'Global'. From the combo box next to the field 'Print Prg.' select 'UTK Object'. Go to the 'Print Prg.' field and click the '?' button. Select the created Visual Query 'GSRP_FAT.vqr'.

Edit the 'Print Prg.' field clicking it, digit 'blank' after the query name and then type 'Preview'.



Picture 132 - 'Generation Options' Dialog Window

Confirm with 'OK'.

Click the 'Save & Generate' button to save and directly generate the changes made to the 'Master/Detail' entity. Go back to the Front-End. Run the application to check the changes made.

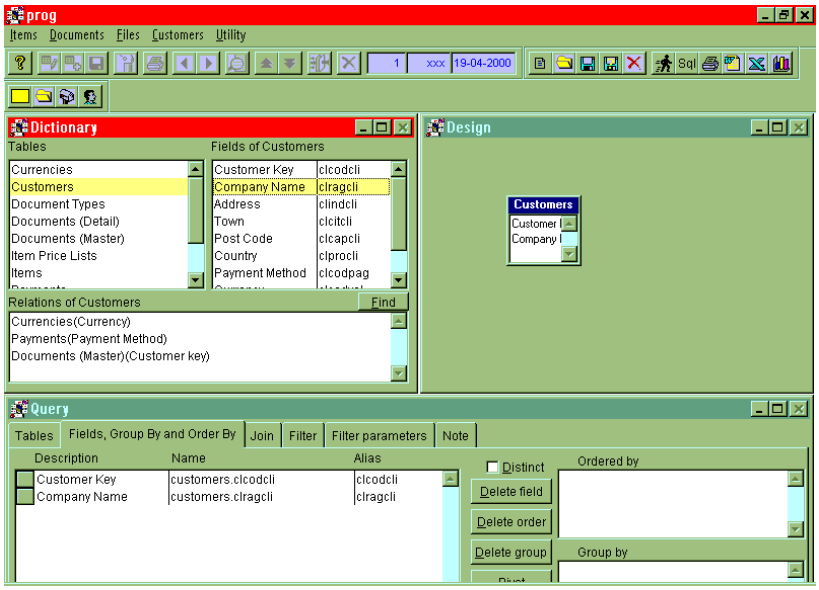
8.2.3 Creating A Query Configuration For A Zoom

Let us now create a query, which is called from the 'Customer Key' zoom in 'Documents' and which displays customers or suppliers basing on the document type.

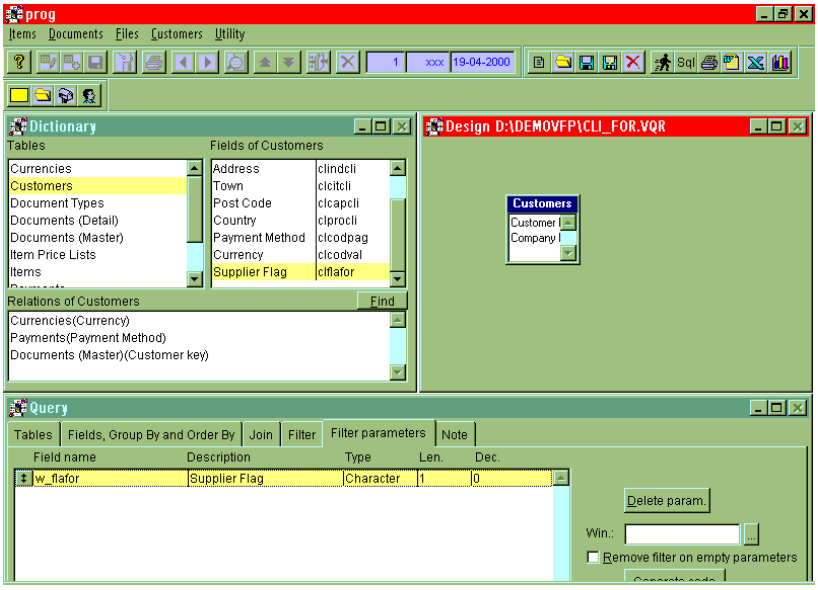
Please note that when you created the 'Documents' dialog window, you defined that the 'TDFORTIP' flag is downloaded from 'Document Types' and input in the 'w_flaflor' variable. This enables you to select customers having the 'CLFLAFOR' flag set to 'N' or suppliers having the 'CLFLAFOR' set to 'Y', simply reading the 'w_flafor' variable.

Run the application, open the 'Utility' menu, select the 'Query Painter' and open the 'Visual Query' tool. Drag the 'Customers' File into the 'Design' section. Move the fields 'CLCODCLI' and 'CLRAGCLI from the 'Fields of Customers' section to the 'Design' section.

Picture 133 - Query on 'Customers' File



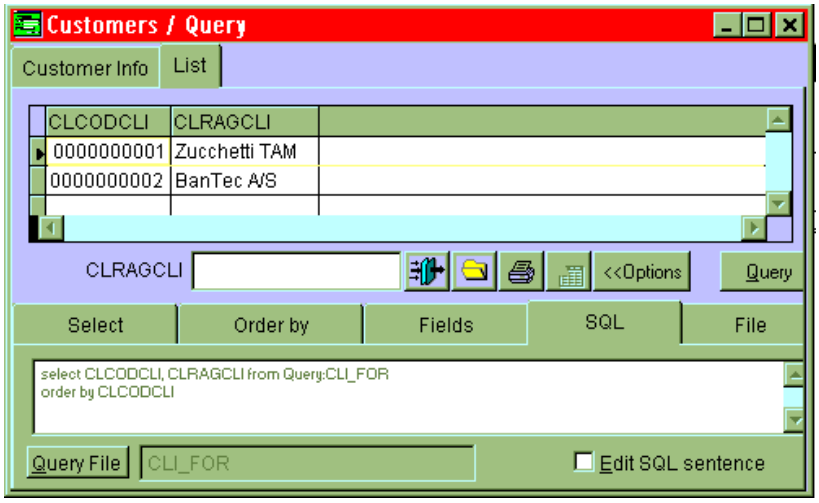
Move the field 'CLFLFOR' from the 'Fields of Customers' section to the 'Filter Parameters' sheet in the 'Query' section. Still being in the 'Filter Parameters' folder go to the 'Field Name' column and change the variable's name into 'w_flafor' (same name as the variable in the Master/Detail 'Documents').



Picture 134 - Defining The Filter

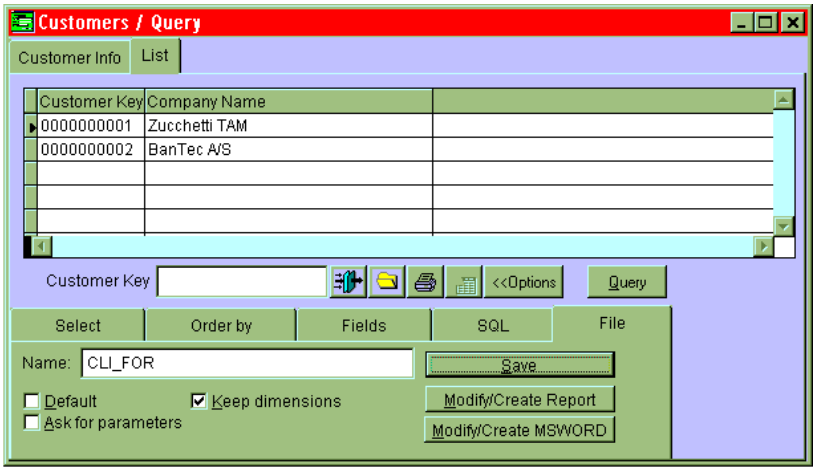
Save the query naming it 'CLI_FOR.vqr' and exit the 'Visual Query' tool.

Open the 'Customers' menu and press 'F9' to open the 'Zoom' window. Click the 'Options' button and then the 'Query' button. Select the created query 'CLI_FOR.vqr'. In the 'Parameters' window click 'Cancel'. Your window should now look like the following picture.



Picture 135 - 'Customers' File Zoom

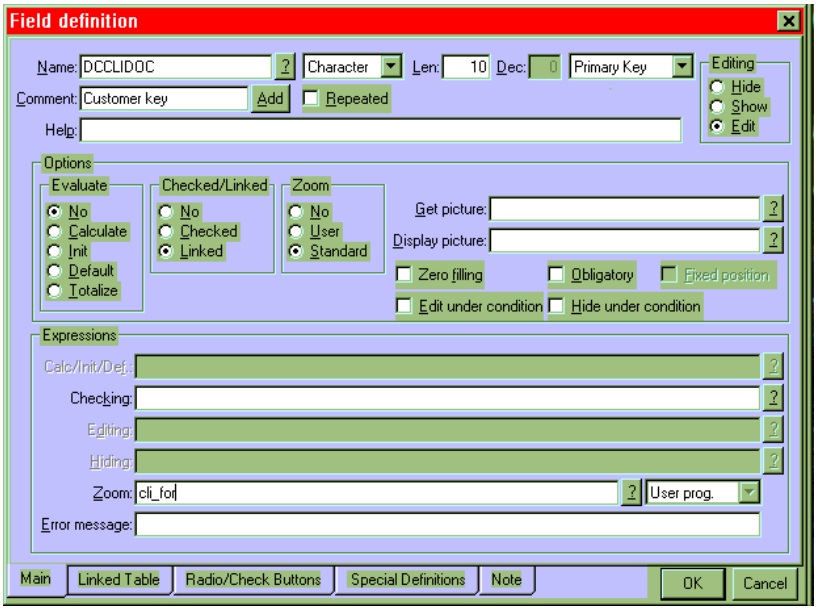
Rearrange the column size and descriptions. Go to the 'File' sheet and digit the name for the new Zoom configuration, e.g. CLI_FOR.vzm. The 'Default' flag must be deselected. Save clicking the 'Save' button and close the 'Zoom' window.



Picture 136 - Changed 'Customers' File Zoom

Exit the application, go back to CodePainter Front-End, and open the Painter window for 'Documents'. Double click the field 'DCCLIDOC'. In the 'Definition' window go to the 'Zoom' field and digit the configuration name for the 'Customers' zoom ('CLI_FOR').

Picture 137 - 'Definition Field' Window For The Variable 'DCCLIDOC'



The customized zoom will be displayed when the zoom on the 'Customers' file is called from 'Documents'. Confirm the dialog window with 'OK', save and exit the Master/Detail Painter to go back to the CodePainter Front-End.

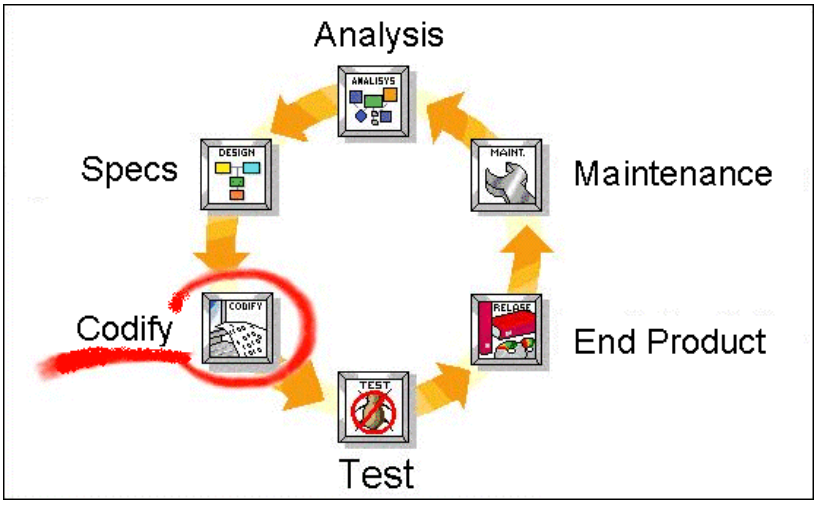
You now need to regenerate the Master/Detail code. In the 'Recent Files' section right click the 'Documents' icon ('GSMD_DOC') and select the 'Generate Code' option. CodePainter does not regenerate the code for the entire application, but for 'GSMD_DOC' only. Run the application and check the changes.

Capitolo 9

Changing The Database

9.1 MS Visual FoxPro Databases

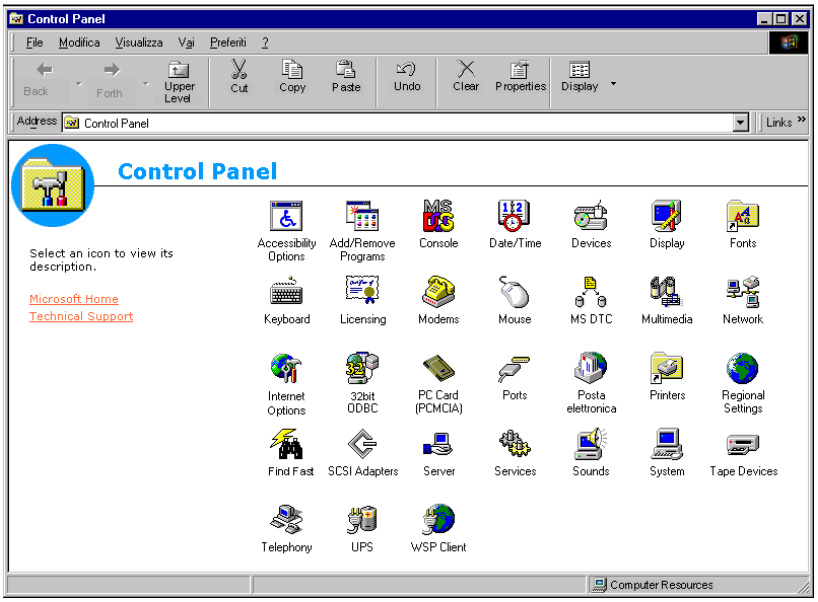
One of the fundamental features of developing Client/Server applications with **CODEPAINTER** REVOLUTION is that they are fully independent from the database used. This means that in whichever development stage you are, or even once the application has been installed, you can always change the database. This makes CodePainter very flexible and makes you save money and time.



This chapter will show you how you can change your application's databases (Visual FoxPro 5.x) into Jet (Acces/.mdb) or into SQL Server 6.5.

9.1.1 From Visual Fox Pro to Jet

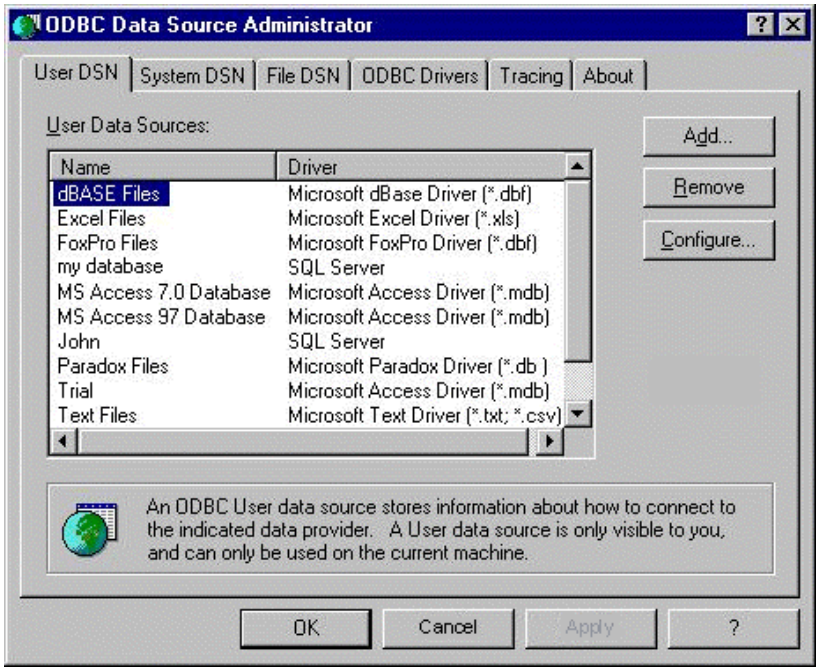
To convert from Visual FoxPro to Jet (the engine used by MS Access) you need to create a Windows 95/98 ODBC connection. Open the 'Start' menu, select 'Set-up' and open the 'Control Panel' containing the system icons.



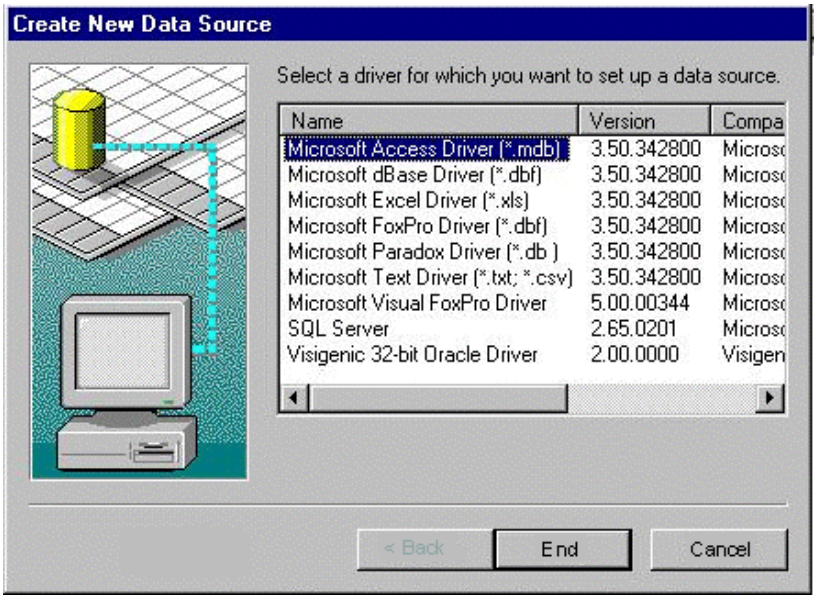
Picture 138 - The Control Panel

Double click 'ODBC' or '32bit ODBC' to open the 'ODBC Modules' window. In the first sheet click the 'Add' button to add a new module.

Picture 139 - 'ODBC Modules' Dialog Window



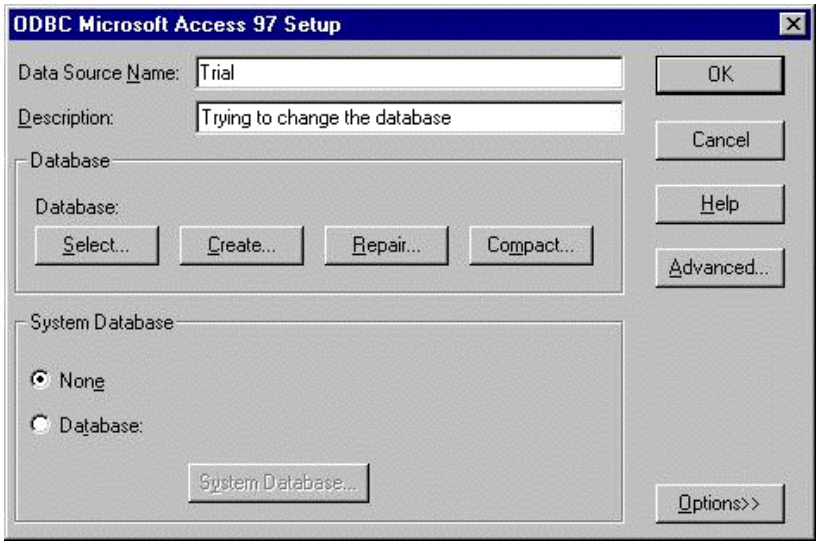
You now need to define the driver for the new connection. Select 'Microsoft Access Driver' and click 'End'.



Picture 140 - Selecting the Driver

You are now required to define the connection name and to give a brief description. Name the connection 'Trial' and input the description 'Trying to change the database'.

Picture 141 - Defining The New Module

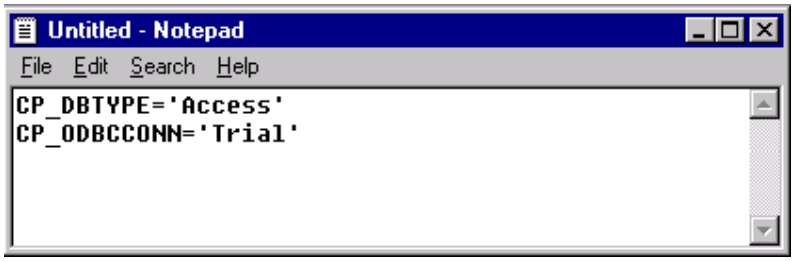


Click 'Create' to create a new database. You are required to name the file and to select the path in which to save it. Browse to find the directory in which you saved your project plan (C:\Newapp) and save the file naming it 'Plan'. You have just configured a new ODBC connection, which allows interfacing the Jet database (MS Access).

The new connection must be communicated to the application, by creating a new file named 'PLAN.CNF' containing the information on the database and the 'ODBC' connection address. Open the MS Windows 'Note Pad' and digit the following instructions:

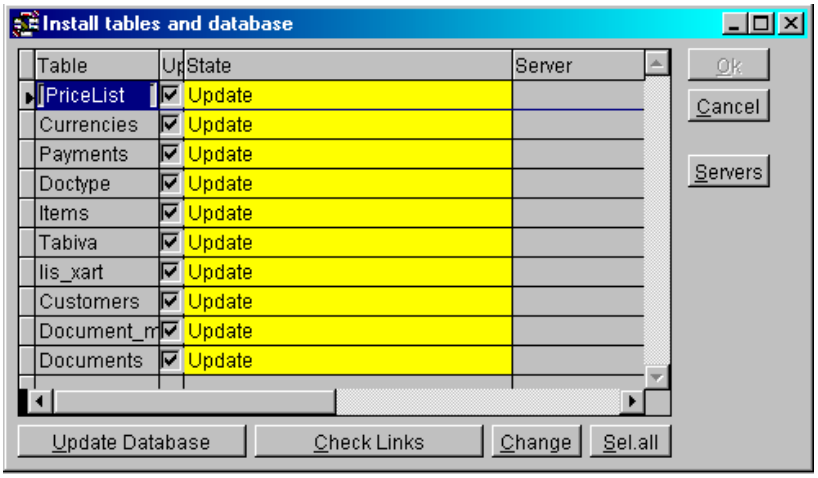
CP_DBTYPE='Access'

CP_ODBCCONN='trial'



Save the file in the application's directory ('Newapp') naming it 'PLAN.CNF'.

Open MS Visual FoxPro and run the application as usual. You can notice that the database update window is opened. This happens because tables do not exist in the new database. Click the 'Update Database' button and confirm the windows that follow.



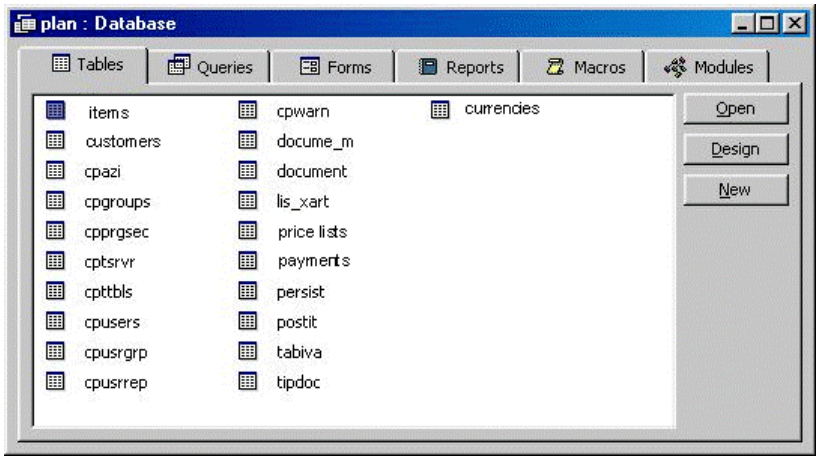
Picture 142 - The Database Maintenance' Window

All databases used by the application are created again. This happens because the system has noticed the existence of a new 'PLAN.CNF', i.e. the name of your project. The file has been read and the new databases created linking via ODBC to the Jet database. Your application manages the Jet database via ODBC and not using Visual FoxPro. Enter some data in the files to verify it.

The application's functionalities have not changed. The only thing that has changed is the database management. Indeed the system deals with this aspect using Jet and the ODBC connection.

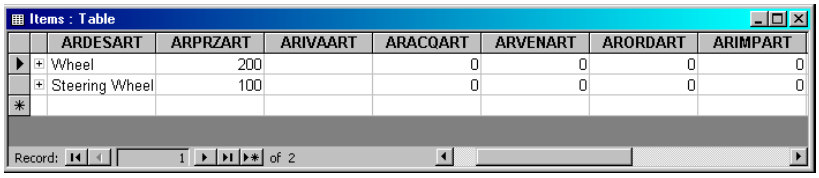
Using MS Access you can now verify whether the created tables are managed correctly. Open the 'File' menu, select 'Open', go to the application directory (C:\Newapp) and open the only file available (PLAN.MDB), i.e. the file you created via ODBC. A new window opens up showing how the database is build. You can notice that in the 'Tables' sheet there are all application's tables.

Picture 143 - Application Tables viewed with MS Access



Click 'Items' and then 'Open'. The list of records input in the application is displayed.

Picture 144 - Corresponding List of Records



Check the other database tables to verify that the change of database has fully succeeded and that the two programs are now compatible.

9.1.2 From Visual Fox Pro to SQL Server

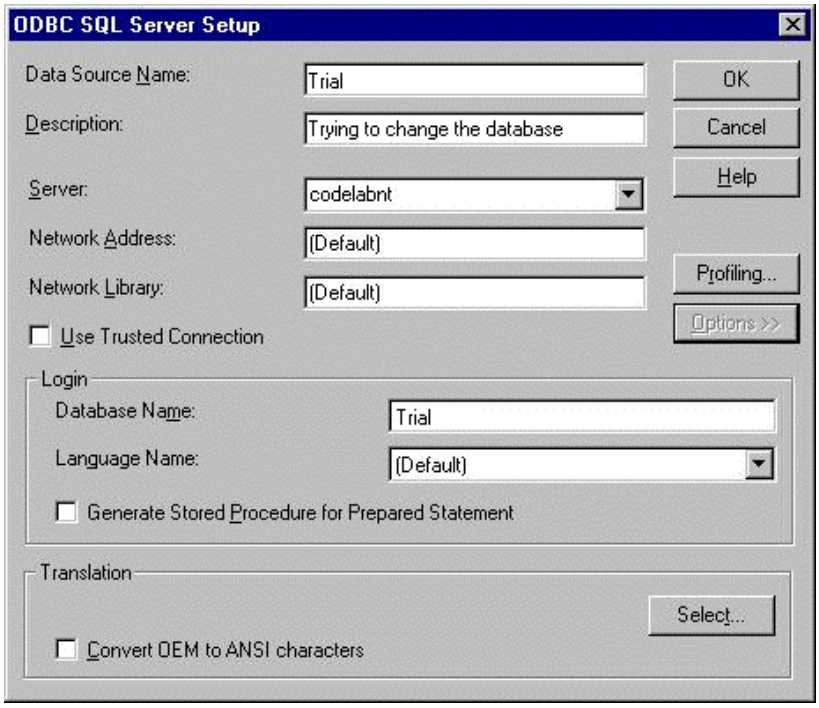
To change the database from Visual FoxPro to SQL Server you need a Server NT, an installed SQL Server and an existing database.

Again, you need to create an ODBC connection. Open the 'Start' menu, select 'Set-up' and open the 'Control Panel' containing the system icons.

Double click 'ODBC' or '32bit ODBC' to open the 'ODBC Modules' window. In the first window click the 'Add' button to add a new module.

You now need to define the driver for the new connection. Select 'SQL Server' and click 'End'.

You are now required to define the connection name and to give a brief description. Name the connection 'Trial' and input the description 'Trying to change the database'. On the 'Server' row digit the name of the server you are going to use. Click 'Options' and on the 'Database Name' row digit the name of the database on the Server NT (e.g. 'Trials').



Picture 145 - Defining the ODBC Module for SQL Server

You have configured a new ODBC connection that allows you to interface the SQL Server Database.

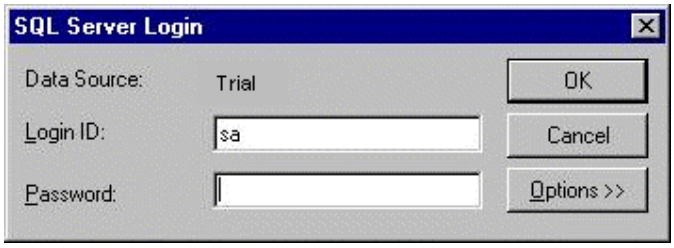
You now need to communicate the new connection to your application, creating a new file and name it 'PLAN.CNF' containing information on the database and the 'ODBC' connection address. Open the MS Windows 'Note Pad' and digit the following instructions:

CP_DBTYPE='SQLserver'

CP_ODBCCONN='trial'

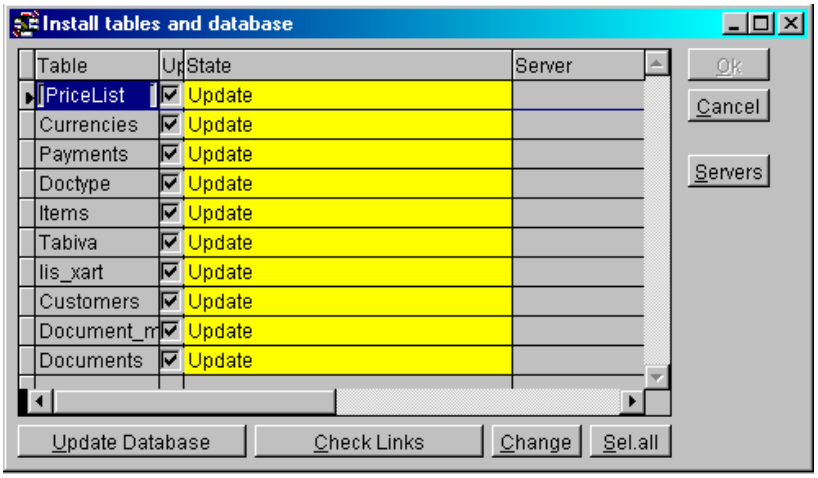
Save the file in the application directory ('Newapp') and name it 'PLAN.CNF'.

Open MS Visual FoxPro and run the application as usual. You are asked to input a login and a password. Input 'SA' (System Administrator) as login and click 'OK'.



Picture 146 - SQL Server Log-On Requirements

You can notice that the database update window is opened. This happens because tables do not exist in the new database. Click the 'Update Database' button and confirm the windows that follow.

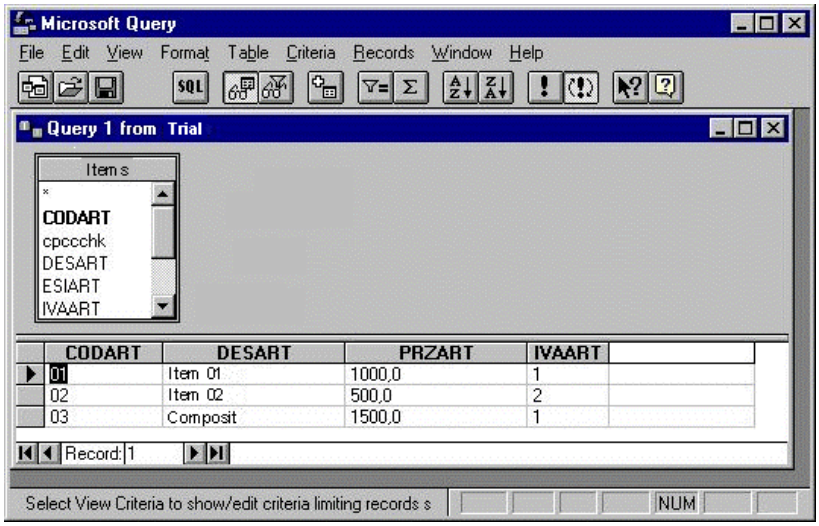


Picture 147 - Database Maintenance Window

All databases used by the application are created again. This happens because the system has noticed the existence of a new 'PLAN.CNF', i.e. the name of your project. The file has been read and the new databases created linking via ODBC to the SQL Server. Your application manages the SQL Server database via ODBC and not using Visual FoxPro. Enter some data in the files to verify it.

The application's functionalities have not changed. The only thing that has changed is the database management. Indeed the system deals with this aspect using the SQL Server and the ODBC connection.

Picture 148 - Example Of A SQL Server Query



Using MS Query (MSQUERY32.exe) you can now verify whether the created tables are managed correctly. You can open a database and execute a query on the files. For example you can execute a query on the 'Items' file and verify the success of the database change.