VALENTINA
Getting started

Paradigma (www.paradigmasoft.com)

Acknowledgments:

**Andy Bachorski , Andy Fuchs, Bill Mounce, Brian Blood,
Craig A. Berry, David A. Bayly, Frank J. Schima, Guillermo Zaballa,
Hideaki Iimori, John Roberts, Lynn Fredricks, Paul Shaap, Robert Brenstein.**

Contents

Introduction

Welcome to Valentina, the scriptable database solution for Macintosh OS. How often have you said, «I wish I could collect all that information in a database?» Valentina is the answer. Using the system level scripting technology of the Macintosh OS, Apple Computer's AppleScript, you can easily build a custom solution to hook your critical, AppleScriptable applications to the fastest database solution on the MacOS. Hook your web server, spreadsheet, email software, accounting or any other scriptable application to Valentina and set your data free of built in storage and speed limitations.

Valentina also doesn't limit you to only working with text. You can store any multimedia data such as Quicktime movies, audio files, graphics files and more through its support of BLOB data.

Valentina is a kind of Relational Database Management System (RDBMS) which has an extremely fast and powerful database engine. When you create a new database using Valentina, it uses its own file system. These files are unique to Valentina and contain the information you want to store. You can quickly identify them on your Macintosh computer hard disk because they will have the Valentina icon. However, they also use the «dot» naming style used in other operating systems, which will also tell you what each file contains. These files are:

dbName.vdb	- Contains a description of the database.
dbName.dat	- Contains most data of the database (including tables and relations)
dbName.blb	- Contains BLOB-data (see below)
dbName.ind	- Contains temporary data used by your database (including indexes and any other temporary data)

Installation

To install Valentina, just drag it into a folder on your hard drive. If you have not registered your copy, a dialog box will prompt you for a serial number each time you open a database. You can get rid of this irritation by getting a serial number.

If you downloaded Valentina from the Internet, visit the Paradigma Software web site at: http://www.paradigmasoft.com. You can purchase Valentina and receive a serial number directly from Paradigma Software or its designated distribution partners.

If you received Valentina on a disk, you should also take the opportunity to visit the Paradigma Software web site at http://www.paradigmasoft.com to ensure that you have the newest support files, and download any free examples available.

Minimal requirements:

68K version:	68020 processor, 2.5 MB of RAM,
PowerPC version:	PowerPC processor, 2.5 MB of RAM
System Software:	MacOS 7.5 or higher, Applescript installed and enabled
Hard disk requirements:	1.2 MB (plus storage for your databases)

Working with Valentina

Now that Valentina is installed and you have serialized your software, you can get started making some databases. Lets look as some things you are likely to script with Valentina:

1. Create the Structure of a Database

If you are not working with databases you have already made, you can have Valentina create a new database structure.

• Creating a new database
• Adding a new table to a database
• Adding fields (columns) to a table

2. Adding and Changing Records

You can create AppleScripts which add or modify the information stored in your database using the built in AppleScript dictionary in Valentina. With a script editor, such as the script editor installed when you install AppleScript on your Macintosh or a third party solution, you can add or modify your database entirely from script.

• Adding new records to a table
• Updating existing records
• Deleting old records

3. Searching and Sorting Records

You can use AppleScript to search or sort information in your database. Because of the architecture of the Valentina database kernel, you can count on an extremely fast response, even with very large databases.
re
• Searching for records which meet some conditions you set
• Sorting selected records by one or more fields
• Looping through selected records

4. Search and sorting with SQL

Beginning with Valentina ver 1.5, you can use SQL to query a database from AppleScript. This new feature significantly simplify scripts and increase performance. By single command you can: select fields, define their order, specify search conditions and sort order for found records.

5. Modifying the Structure of a Database

After you create your database, you are not limited to the same fixed structure. You can use AppleScript to programmatically alter the structure of your database to suit your needs.

- Adding and removing fields from tables
- Changing attributes of fields (type, indexing, etc)
- Adding and removing tables

6. GUI of Valentina

Beginning with Valentina ver 1.5, Valentina gains a visual database browser that you can use to create, modify and work with Valentina databases.

But you are not limited in only working with the native visual interface of Valentina. Using AppleScript powered products such as GUI builders like FaceSpan or OneClick, you can create small, custom solutions that unlock the power of Valentina. These products can send and receive AppleEvents from Valentina. Also, since Apple first introduced AppleEvents, many standard Macintosh applications send and receive AppleEvents, allowing you to create solutions using many third party applications.

Using a GUI builder or the native ability of an AppleScript aware application, you can display your list of records. Two excellent examples are displaying records in a spreadsheet, or displaying records using the native ability of a web server to serve server-created web pages.

Create the Structure of a Database

Here is a simple sample script with creates a new database and adds a table with several fields. You can use this example in working with contact data, such as that found in products like Now Up-to-Date & Contact or Chronos Consultant.

In this example, it will create a database called «Customers» and add fields for each customer's first and last names, birthdates and telephone numbers. Since this database will be used primarily in Germany, it sets the language to German.

```
set DB to make new database with data file “Customers db”
set Customer to make new base object with properties {name:“Customer”} at end of DB

tell Customer
	make new field with properties
		{name: “FirstName”, type: tString, length: 20, language: “German” } at end
	make new field with properties
		{name: “LastName”, type: tString, length: 20, language: “German” } at end
	make new field with properties
		{name: “BornDate”, type: tDate } at end
	make new field with properties
		{name: “Phone”, type: tString, length: 10 } at end
end tell
```

Adding a New Record to the Table

Next, you will want to add new records to your table. Here you add information on customer «John Pierson» to the end of the your database after discarding («blank») any data in memory previously allocated to the table.

```
tell Customer
	make new record with data { “John”, “Pierson”, “03/05/65”, “2456897”} at end
end tell
```

Or, you could instead update an existing record with the same information.

Updating a Record in a Database

You can also update records in your database. Here you change the current record to the same customer information on «John Pierson».

```
tell Customer
	set current record to record 1 of SomeSelection
	set fields to { “John”, “Pierson”, “03/05/65”, “2456897”}
	update
end tell
```

Valentina gets record one, puts the «John Pierson» information into a memory buffer, then «updates» the fields in record 1 with the new information.

Deleting a Record in a Database

You can easily remove a record from your database with two lines of code, one to make the record the current record, and the second to delete it.

```
tell Customer
	set current record to record 1 of SomeSelection
	delete current record
end tell
```

or

```
tell Customer
	delete record 1 of SomeSelection
end tell
```

Using a loop, you can easily delete more than one record at a time based on specific criteria.

Selecting All or Some Records of a Table

You may want to work with all records in a table in order to search, sort or modify all of them.

```
set SomeSelection to select records of Customer
```

You can also use the «tell» AppleScript keyword to do it this way:

```
tell Customer
	set SomeSelection to select records
end tell
```

Or, you could select records which only meet specific criteria. In this example, it only selects records where the customer's first name is John AND who also has a birth date beginning in 1960 but no later than the end of 1969.

```
tell Customer
	set Selection2 to select records where
				{field "FirstName", "John",
				 field "BornDate", ">=1/1/1960 and <12/31/1969"}
end tell
```

After you select some records, you can sort, modify or delete them.

Sorting a Selection of Records

Once you have a selection of records, you can sort them according to one or more criteria. One example is sorting your customer list by first and last name:

```
set Selection3 to sort Selection2 by {field "FirstName", field "LastName" }
```

You can also sort by numerical or date values, such as sorting by birth date:

```
set Selection3 to sort Selection2 by {field "BornDate" }
```

Sorting a Selection of Records into a List

Using AppleScript, you can loop through your selected or sorted records so you can make them into a list or make calculations based on all the values of some fields. To do this, you put the first record (record 1) through the last record (record RecsCount) into the list using a repeat loop.

```
set RecsCount to get count of records in Selection3
repeat with i from 1 to RecsCount
		set current record of Customer to record i of Selection3
		set FieldsList to fields of the Customer
end repeat
```

Searching and sorting with SQL

Using command ‘SQL’ you can perform complex multi table querying to the database by one command. As a result of query you get a Cursor.

Work with a Cursor is very similar to the work with a BaseObject (Table):
- a Cursor has fields (columns)
- a Cursor has records (rows)
- a Cursor has current position (current record)

Actually a Cursor represents a resulted table of query.

Example:

```
tell DB
        set theCursor to SQL Select "select name, address, total from Customer, Invoice”
                                                “where name > 2  order by name desc"
end tell

set Recs to get count of records of theCursor
get records 1 thru Recs of theCursor

delete theCursor  -- free memory
```

Important:
you must delete cursor to free memory of Valentina from results of query.

Modifying the Structure of a Database

You can change the structure of a database runtime.

You can add/remove fields to the table:

```
tell base object “customer” of DB
        make new field with data { name:”phone”, tString, 15 } at end
        delete field "Address"
end tell
```

You can add a new table as shown in the part “Create the Structure of your Database” or remove a table from the database.

```
tell of DB
        delete base object "Customer"
end tell
```

GUI of Valentina

The environments of Valentina

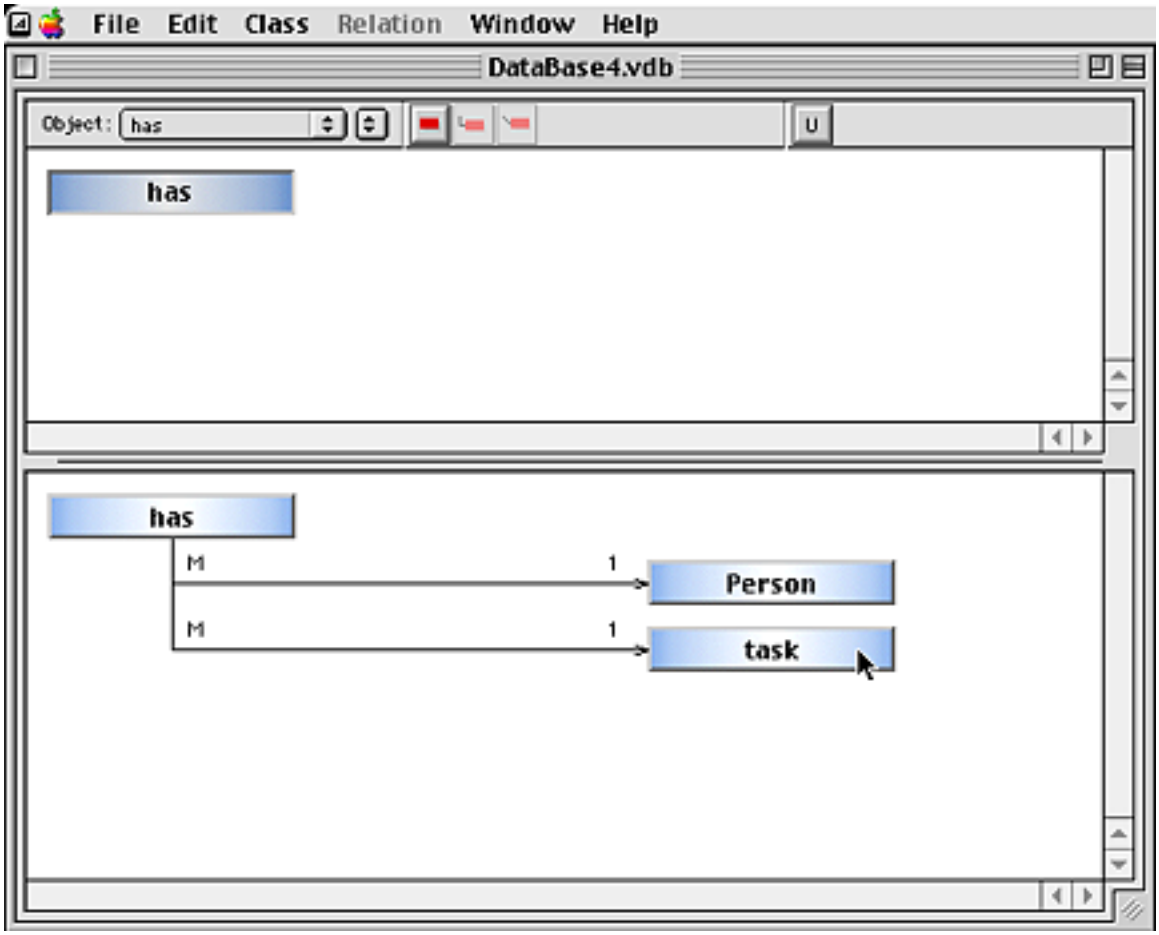
Developer environment – Here you can design the scheme of your database, i.e. create or remove Tables, add/remove fields of the Tables, and define parameters of the fields.

User environment – Is the environment in which you can browse and manage the contents of your database, i.e. you can:
• Add/remove/modify records to the Tables
• Search for specific records
• Sort records
• Import and export data

When you are working with a database you can be either in the developer window or in the user window. If you switch from one window to the other then the first is hidden, so you see only one environment at a time. To switch from one environment to the other, you can use the buttons “U” or “D”, or use the Windows menu item.

The Developer Window

The Developer window is split in 2 parts:
• **Inheritance diagram** (top)
• **Relations diagram** (bottom)



The Header Bar of the Developer Window

At the top of the Developer window is a header bar which contains the following controls (see the previous page for a picture):

• **BaseObject menu** – Popup menu which contains all BaseObjects (Tables) of the database. Choosing a BaseObject from the menu makes it current, i.e. it will be displayed in the Inheritance diagram.

• **New Root BaseObject button**
This button creates a new BaseObject (Table) in the database. Root BaseObject means a standard Table of your RDBMS. You can use the item New root class of the Class menu to get the same result.

A new BaseObject gets the default name “BaseObject N” and a red button appears on the diagram. The red color means that this BaseObject must be finished, i.e. you should create at least one field.

• **New Hard Child BaseObject button**
This button creates a new child BaseObject of the current BaseObject with Hard Inheritance. You can use the item New hard child of the Class menu to get the same result.

• **New Soft Child BaseObject button**
This button creates a new child BaseObject of the current BaseObject with Soft Inheritance. You can use the item New soft child of the Class menu to get the same result.

• **Switch to the User Environment button**
You can use the item Browser window of the Window menu to get the same result.

The inheritance diagram

Shows an inheritance tree of a root BaseObject and its children (in the current version, Valentina supports root BaseObjects only, this matches to a standard Table of a RDBMS).

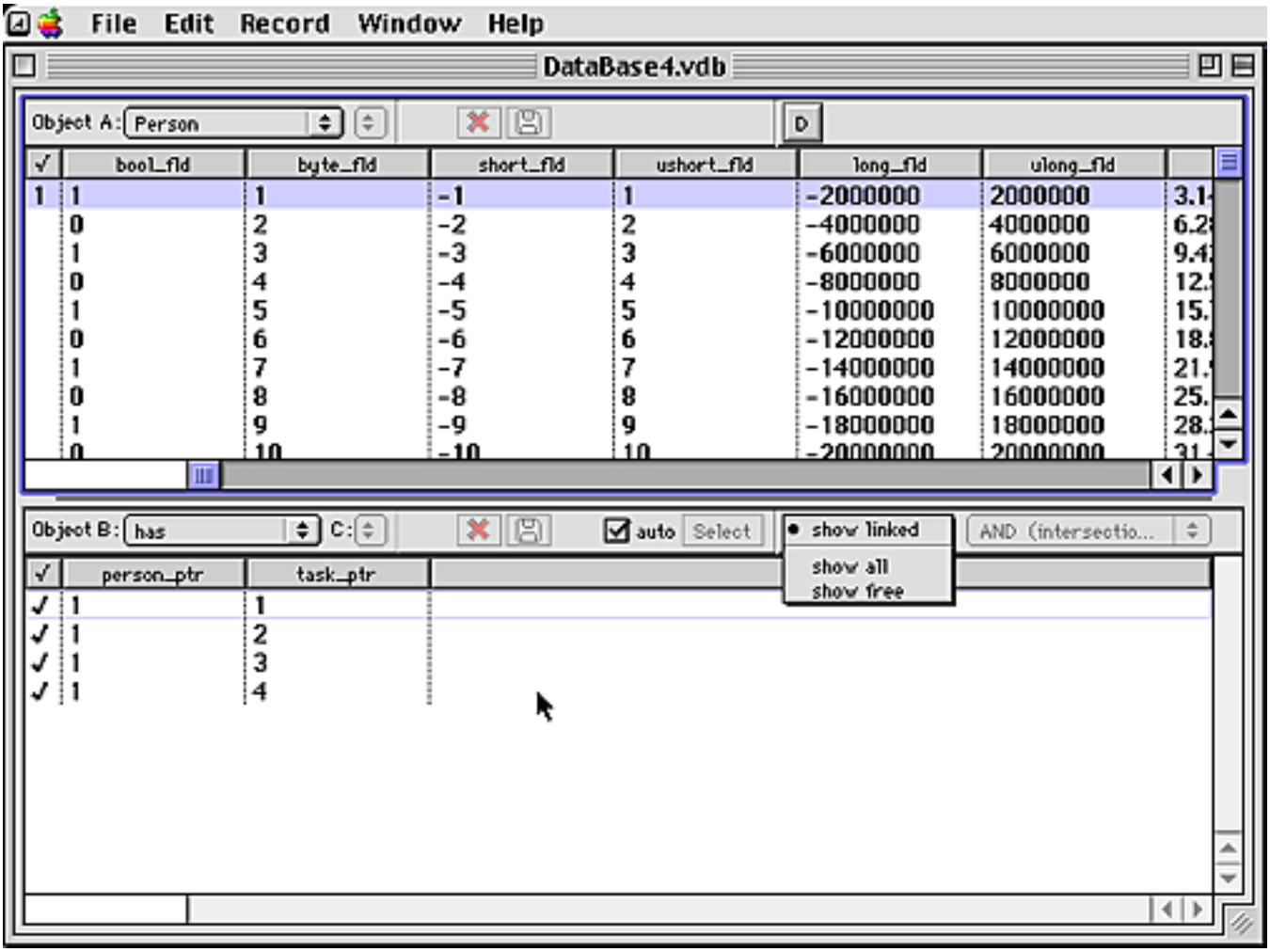
A BaseObject selected in the inheritance diagram (its icon looks like a pressed button) is called the current BaseObject. To select a BaseObject, choose it from popup menu or click its button on any diagram.

The relation diagram

Shows BaseObjects (Tables) related to the current BaseObject. Type and parameters of each relation are also shown on the diagram.

The User Window

The User Window is split into 2 parts:
• Main Browser **(top) – Displays records of the selected BaseObject.**
• Related Browser **(bottom) – Displays records of a BaseObject related to the Main BaseObject.**



pict. “User Window”

The Main and Related Browsers are intended either for displaying records in the Standard Output Layout or for entering/modification of records in the Standard Input Layout.

Standard Output Layout

Displays records in a table.

In the table you can select one and only one record, the Current Record, by mouse click. The Current Record is outlined by a color.

By performing a search you can build a selection of records which match to some search conditions. As a result of a search, the output layout will show say 2'000 records of 140'000 records of the Table. Records displayed in the output layout are called the "Current Selection".

Valentina adds an additional column, the Mark Column, at the beginning of the Output Layout. By a mouse click in that column you can mark record(s). The first marked record gets marked "1", the second gets marked "2", the rest are marked by the character "✓". In this way you can build a Marked Selection. There are menu commands which affect the Marked Selection, for example "Delete marked records". Also, the Related Browser displays records related to the marked record(s) of the Main Browser.

In summary:
• Output Layout can display the Current Selection built by a search or manually.
• Inside the current Selection you can build a Marked Selection.
• Only one record can be the Current Record.

By a mouse click on the header of a column, you can sort the Current Selection on that field. The column header will look like a pressed button. Clicking on a sorted column header will remove sorting of the Selection. To sort by several fields you can use the Sort Dialog (command-S). Valentina will use the current sort order for the all successive Selections until you change it.

To search, use the item Find of the menu Record (command-F).

To add a new record to the table, use the item New record (command-K) of the menu Record. The Input Layout will be displayed, where you can enter the data of the new record.

To modify an existing record, double click it in the Output Layout, then the Input Layout will be displayed, which allows you to modify a record.

Standard Input Layout

Used for adding new records, for modifying existing records, and for entering search conditions.

To save changes made in the Input Layout click the button OK in the header of the Browser or press Enter on the numeric keyboard. To cancel changes click the button Cancel in the header of the Browser or press Escape

The Header Bar of the Main Browser

At the top of the Main Browser is a header bar which contains the following controls (see the previous page for a picture):

• Object A menu
Popup menu which contains all BaseObjects (Tables) of the database. Choosing a BaseObject from the menu makes it the main BaseObject

• Cancel button
Cancel any changes made in the input or search layouts, returns back to the default output layout. You can press the Escape button on the keyboard to get the same result.

• OK button
Confirm changes made in the input layout or start search. You can press the Enter button on the numeric keyboard to get the same result

• Switch to the Developer Environment button
You can use the item Developer Window of the Window menu to get the same result

The Related Browser

To display records of a related table, Valentina uses the Related Browser which implements a new concept called Dynamic Layouts, which expands the concept of the Installed Layout used by FileMaker Pro and 4th Dimension.

For example, say you have 5 related tables in your database. With FileMaker or 4D you need to manually build 20 layouts to be able to navigate from any table to any related table. With Valentina you get everything automatically.

Also, the Related Browser can display related records in different modes and using different Set operations (see description on the next pages). Compare to FileMaker and 4D which can display records only in “show linked” mode in the Installed Layout and they can not use Set operations (4D would require programming to do that).

Using the “Object B” menu in the header of the Related Browser you can select any BaseObject (Table) related to the Main BaseObject.

Work of the Related Browser (RlBr) is based on the Marked Selection of the Main Browser. The RlBr finds and displays records of the “BaseObject B” which are related to the record(s) of the Marked Selection. For example, on the picture “User Window” we can see that record “1” of the Table “Person” is related to 4 records of the Table “Has”.

NOTES:
To establish a relation between 2 Tables, Valentina uses fields with type ObjectPtr, which play the role of a “Direct Pointer” (see “Relations between tables” in the Technical Reference).

How Valentina Relates Records

To relate records, a standard RDBMS uses a “Pointer by Value”. This means that a Table has a special field which is a foreign key of another Table. To relate a new record, the user must enter into that field a special code – an identifier of the parent record. An identifier can have some inherent value, for example social security number, or to be more effective, be some special number. To change the relation between records, the value of this field must be edited.

Valentina uses another technique called “Direct Pointer”. It stores internal numbers of the tables, so the user never edits them manually. Then how can a user establish a relation between records? For this Valentina offers a new feature of a DBMS GUI – Mark Column and relating via single mouse click.

1) Relating a new record
If a record is marked in the Main Browser, then any new records created in the Related Browser by the command “New record” will be automatically related with it.

2) Relating existing records
For this you should set the mode of Related Browser to “Show All”. By clicking in the Mark Column of a record in the Related Browser, you relate it with one (one to many relation) or several (many to many relation!!!) marked record(s) of the Main Browser.

3) Break relation between records
You can break a relation between records in the mode “Show Linked” as well as in the mode “Show All”. When related records are marked by “3” then you need to click the Mark Column of that record in the Related Browser.

If you use the mode “Show Linked”, then after breaking a link that record will be hidden in the Related Browser (because it is not linked anymore). Of course, the record itself still exists in the Table.

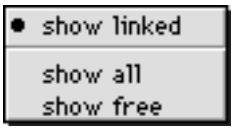
NOTES:
If the Type of relation between 2 tables does not allow you to relate the record which you are trying to relate, then the mouse cursor will become an “X” over the Mark Column of that record.

The Header Bar of the Related Browser

At the top of the Related Browser is the header bar which contains controls like in the main Browser, but it also has the following additional controls:

• Object B Menu
Popup menu which contains all BaseObjects related to the main BaseObject (if any).

• Auto Select Group
If this check box is selected, then when the user selects a record(s) in the main browser, Valentina automatically locates related records in the Related Browser. If the database is big or for some other reason, then the user may wish to disable Autoselect. In this case selecting records in the main browser does not affect the Related Browser until the user clicks the Select" button.



• Mode Menu
This menu defines the mode of how related records will be displayed. There are 3 options:

- show linked (default)

In the related browser, only record(s) related to the record(s) marked in the main browser are displayed

- show all

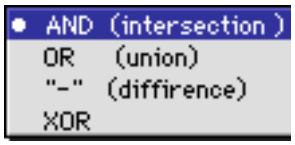
In the related browser, all records of the related BaseObject are displayed. Records related to the marked record(s) of the main browser are marked by "3".

This mode allow you to easily relate additional record(s) and break relations between records by clicking the check mark column in the Related Browser.

- show free

In the related browser, records which are not related with any record of the main BaseObject are displayed.

The displayed records are not related, and thus none of them will have a check mark. If you relate any of them by clicking the check mark column and Auto Select is ON, then the related records will be hidden (because they are no longer free).



• Set Operation Menu
This menu is enabled only if at least 2 records are marked in the main Browser. It defines a Set Operation to be applied to Sets of related records.

Lets consider a simple example.
In the Main Browser 2 records are marked: 1 and 3.
The main record 1 is related with records { 2, 3, 6, 9, 15} of BaseObject B.
The main record 3 is related with records { 3, 4, 5, 6, 8, 9, 12} of BaseObject B.

• AND (intersection)
Build intersection of Sets of the related records. In our example, in the Related Browser will be displayed records { 3, 6, 9 }

• OR (union)
Build union of Sets of the related records. In our example, in the Related Browser will be displayed records {2,3,4,5,6,8,9,12,15}

• "-" (difference)
Build difference of Sets of the related records, i.e. in the resulting set will be records which are in the first Set but not in the second. In our example, in the Related Browser will be displayed records {2,15}

• XOR (symmetric difference)
In the resulting set, records which are members of one Set only are included. In our example, in the Related Browser will be displayed records {2, 4, 5, 8, 12, 15}

NOTES:
• Operations AND and OR can be used for 2 and more records.
• Operations Difference and XOR can be used only if exactly 2 main records are marked. The First main record is marked as "1", the second as "2".

What is the use of these operations?

Example:
Let's say we have the Table Person related as Many to Many with the Table Task.

We want to find: "Which tasks must John and Bob do together?"
To do this, mark the records of John (1) and of Bob (2) in the Main Browser and set the operation to AND. The intersection of their tasks will be shown in the Related Browser.

The operation OR will display "tasks which can be performed by John or Bob".
The operation Difference finds "tasks which must performed by John without Bob's help".

Where to Go Next with Valentina

You should take the opportunity to work through the Valentina tutorial, also included with the Valentina software. The tutorial steps you through each of the example scripts included with Valentina.

Paradigma Software will add new examples and tips to the Paradigma Software web site. You should visit Paradigma at http://www.paradigmasoft.com to see if someone else has already posted some code you can use freely to help build your custom solution.