User’s manual

Table of contents:

About the manual

Manual introduce you new **Diamond-Draw** program and intended to users having basic knoledge of pic language . however reading the manual one can learn more than using a dry interface only but capabialities of pic language itself.

In overview we’ll talk about goals we tryed to achieve, consider what points of previuos interface we want to improve and make some assumptions on language we’ll use and constrains we’ll try to preserve.

First chapter intoduce new system ,new terms we will use and explains how to build first example.

In chapter 2 we consider the Edit toolbar - a tool which makes use of standart program options more convinent and simple.

Chapter 3 and 4 intended to direct mouse manipulations - the power of new interface. In chapter 4 we concentrate how to manipulate with group of elements and element insertions.

Chapter 5 returns us to the diagram creation typing text but using an internal text window with its advantages such as history of commands, autoappending commands and the most important the real time picture update as earlier (using direct manipulations)

Next chapters 6,7 and 8 introduce advanced program features: relative size settings, font settings, grid changing defaualts. Also in chapter 8 we learn how to use standart macros and make them youself.

The last chapter talks about use of accelerators and getting Help

Glossary at end of the manual have a full dry descriptions of all functionality of new interface.

Overview of product

For many reasons in computer sciences we need simple diagrams containing boxes , circles and other explaining documents we working with.

For creation such documents (in UNIX environment) we have a specially designed tool - **Pic** , language which support mentioned diagrams and allows to build sophisticated things . However working with it we are forced to batch drawing. It has its own advantages and disadvantages. It makes it possible to insert diagrams in a document, but it not answer to a modern design approach.

Of course there are few programs that present **WYSIWYG**(what you see is what you get) approach . Someone is compatible to Pic compiler (xfig WD-pic) but they are not as flexible in drawing as Pic language allows and have a room for improvement.

As main lacks of the previous solutions we may mention :

- They are not flexible and not allow to do everything that the Pic does
- There are too much dialogs in these products
- Working with mouse is not supported
- Current object position is very hard to be altered

Our purpose is to design a tool which will be able to present more features , be more flexible reliable and simple to use.

Our goals during the design of this program are:

- Preserve that internal representation looks like human-written Pic description
- Implement multilevel undo and redo mechanism (changes will not be so harmful)
- Avoid dialog boxes and confirmation clicks as much as possible
- Both attributes and arguments are changeable
- As indicative names for buttons as possible

As been said in **WD-Pic** (one of the previous WYSIWYG approaches) specification it allows to create simple schemes on the fly but any unusual element requires too complex actions and it may be done in text editor with less efforts. It’s clear that to perform simple and powerful interface is a very difficult stuff. Anyway we try to perform more direct mouse actions , mouse menu which will allow to edit current element ,more features for editing element and group of elements. Besides of it , we do a special text window and user will be able to make the same operation as in text editor ,but see the updated picture . Mix of direct manipulation and textual changes will allow to create things using all Pic language power.

So using our product you will(as we hope) achieve both the regular Pic power and comfortable interface.

Now let s talk about the language we' ll use for describing our product.

There are two basic ways to create a picture and its representation in the product:

- using the WYSIWYG graphic interface to draw the picture using one from follows:
  - direct command choosing from different menus
  - menu accelerators
  - direct mouse manipulation
- writing a textual description of the picture in the Pic language

From the user s point of view these two ways are equivalent in the final result.

By those two ways you can:

1. Insert in your document any graphical element that specified in the Pic language:
   - box , arrow , ellipse , circle, line, scale, spline , arc , text
2. Specify directions , that influence the rest of the picture:
   - up , down , left , right
3. Define element parameters:
   - size (width , height , radius)
   - line style (solid , dotted , dashed , invisible)
   - places (at , with ... at)
   - adjustments (above , below , right , left)
   - arc directions (clockwise)
4. Use default parameters
   - scale , boxwid , boxht , linewid , lineht , circlerad , arcrad , ellipsewid, ellipseht,
   - movewid , moveht
5. If you want to insert in your document the same picture more than once or if you want draw different geometric objects depend on some condition there are defined the next operators:
   - if ... then ... else ...
   - for ... to ... by ...
   - loop

`We assume the next thing` :*not every Pic function will be described in this user manual. A user who want deal with some too sophisticated stuffs may receive appropriate information from the existing Pic manuals.*

What really will be discussed :

Operations that you can perform using the product functionality which is specified by WYSIWYG approach (by the way we cover almost everything that defined in the Pic so we assume it will be enough)

Those functions for working with files

- New
- Open
- Save
- Save as
- Print
- Print preview
- InsertFile(add some file at given position)
- AppendFile(add some file at the end of the editing)
- Quit

Those functions for document editing

- Clear
- Select All
- Move (Manipulation with group of elements)
- Cut
- Copy
- Paste
- Delete
- Undo
- Redo
- Properties
- Attach label

Those functions for geometric elements drawing:

- Box
- Circle
- Line
- Spline
- Arrow
- Ellipse
- Arc
- Text
- Same
- Custom
- End
- Move (pic element)

Those built in functions:

- if...then...else
- for...to...by...do
- Loop (drawing element desired number of times)

It is possible to define a label that simplifies the working with object locations.
Also user can insert in object some text (associated text) or write something he want to add to edited

document (standalone text)

Those allow to view a wanted document:

- View source (open separate window with a source text of the current picture)
- Text editor (open text editor with a source of the current picture)

Specific Options:

- Scale
- Set defaults
- Font settings
- Grid settings

  All objects sizes are given in terms of grid size so grid size changing forces elements changing in terms of new grid size. This grid is just internal program structure which forces user not to change element size in couple of pixels (which will change inch representation by 0.00.. inch)but in terms of 0.05 inch.
- Relative size and position settings

  We can define height and width of an element in terms of other element sizes. For this purpose we have in Pic language:

  box width = B.width ...

  User should set relative sizes by choosing some element. Program will define relative height and width and one can use these settings for next elements.Use of this option will be close to regular parameter settings.We define the next syntax:

  width 3x - element width is (3 * relative width)

  height 2x - element height is (2 * relative height) and so on.
- Label view on/off (if user may see labels)
- Tips on/off (if tips occur near each button)
- Hide/Show *Edit Toolbar*

And Help availiable in next forms:

- About the program
- Help Index
- Tutorial

The program defines different cursors for different tasks:

- You working with base cursor if no special mode mentioned below has been activated
- Cursor for **At,To** position operations is ’+’.By it user locate desired point on the screen for further operations.
- Cursor for getting help on graphic interface componenets: ’?’

All above has standard meaning but in unusual cases needed comments are added.Besides all this there is a set of features added for more comfortable working with the product:

- hide/show toolbar.
- help when mouse points on some button-tips.
- help on every function and button, sorted by topics.
- text window always available.
- accelerators for wide used buttons or options

- some functions on right button of mouse
- changing fonts for text
- choosing current object or group of objects by clicking on it
- changing size of object by dragging it
- changing position of object using direct mouse manipulation
- deleting number of objects using direct mouse manipulation
- loop copying element or group of elements using direct mouse manipulation
- for current object always available all functions in right button
- indicating attributes like at , from , to by clicking on appropriate point
- change all objects parameters by changing it attributes directly
- button for changing line style and font settings
- button for fill bounded elements (circles,boxes and ellipses)
- attaching label to object by clicking on mouse
- indication insertion point by clicking on mouse
- changing font and font size for particular text in element by direct mouse manipulation
- changing line style and palette for particular element by direct mouse manipulation

We should mention that some advanced features are added in the product they will be discussed in the next chapters.

`In all chapters we try to keep next constrains to facilitate your efforts understanding the manual`

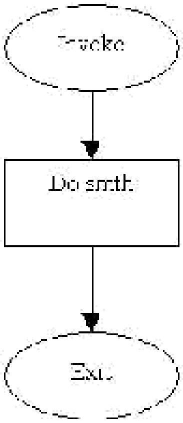
- Text related to the diagrams signed by < > tags ,for example box with text <box>
- All definitions we make typed using *Italic* font style.
- All new terms signed using **Bold** font style.

Back

1.From the beginning

All pic language diagrams are based on basic pic elements. There are: **box**, **ellipse**, **circle**, **line**, **arrow**, **arc**, **text** and **move**. All elements have their own attributes like position, line style or associated text. Pic compiler uses those attributes to perform desired diagrams. Some of these attributes are necessary other are optional.
Consider we want to create our first diagram. It may mirror the work of any program:

Diagram1



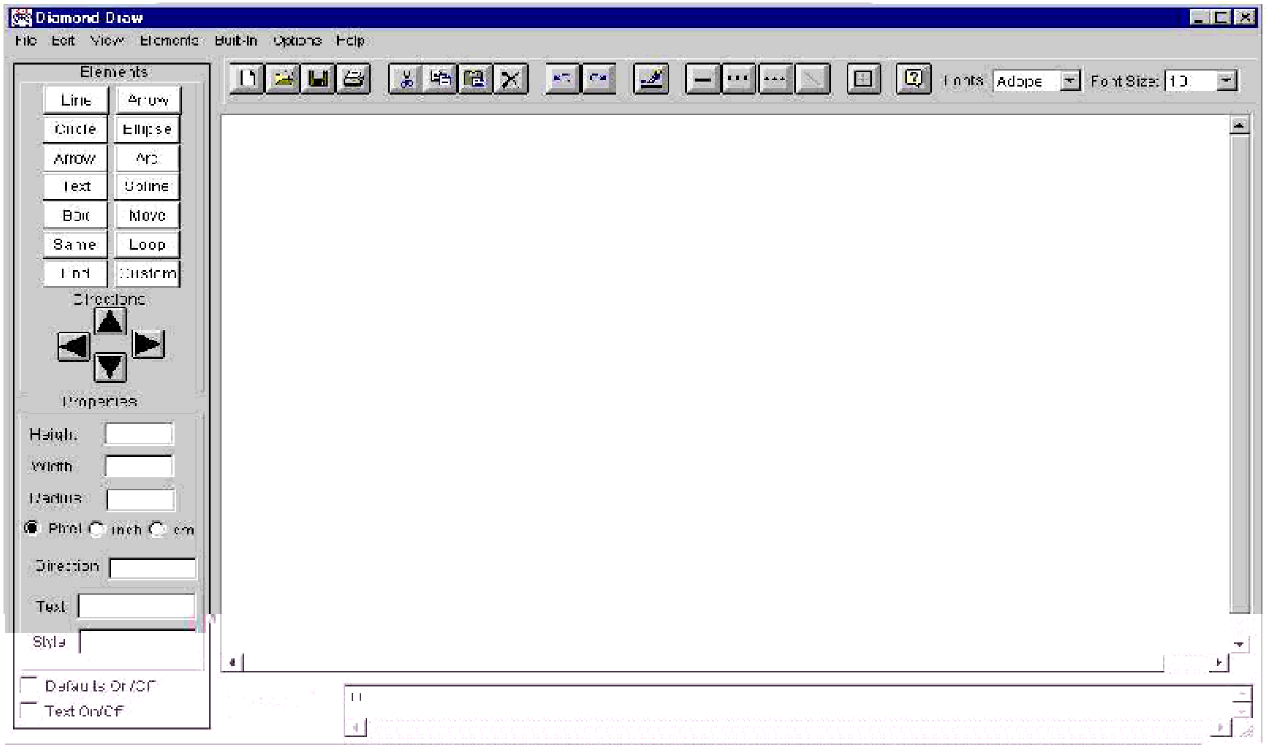
The text of this program can be as follows:

```
.PS
down
ellipse "Invoke"
arrow
box "Do smth"
arrow
ellipse "Quit"
.PE
```

To create this piece of code you can use your favorite text editor after that compile it and see if your program produce a desired diagram.

This so called BATCH approach .Of course it seems not so convenient to do changes and only after recompiling see the change. Another option is to open Diamond-Draw program and do it there. What we see once opening Diamond-Draw?

Main window looks as follows:



● Pixel ○ inch ○ cm

Fonts Adobe Font Size: 10

On the left panel there is **Pic token bar directions buttons ,properties bar** and some checkboxes . In our first example we will use first two to perform **Diagram 1** So the first keyword is down. This is one of the four basic directions in which element can be drawn (related to last element).
We can set this direction (down) by clicking *Down* direction button.
After that we want to draw ellipse with <*Invoke*> text inside. You paid attention that our box has internal text (further we called it associated text) .For bounded elements such as box, circle or ellipse you can set if program asks you for associated text automatically or not.

This option can be set using *Text On/Off* text checkbox which can be found at left panel.We can use it and as we will see later this will economize some time.Nevertheless Text Off is default.
Assume we choose this option. Now clicking *Ellipse* button cause dialog for **associated text** to appear. There we will type <*Invoke*> and after clicking *OK* or pressing *Enter* (in all dialogs pressing *Enter* is equivalent to clicking *OK* button) ellipse with text Invoke will appear on the screen. Element is signed by color border which means **Current element** . More about it you can read in the next chapter.
Now we can continue with arrow just clicking the *Arrow* button. It will be drawn down related to the ellipse since direction we set entirely takes effect until other direction chosen. Since the rest actions to

complete the diagram are same to already done we leave to reader complete this diagram.
After diagram is done we can compare text generated by program to source code above.To do that open **View** menu and choose**View source** option. Separate window with already familiar text will be opened.
During the work you of course paid attention that for every drawn element **Properties window** on pic token toolbar has description of current element attributes. There are so called **Default attributes** . In future we will see how they can be modifyed or added.
The other thing that you of course noticed is that mouse location on the screen always can be found in small window near the left bar. Further we will use name **Mouse location window** for that.
May be you want to save your first example ? You can do it clicking **File** menu and then choosing **Save As** option. Save dialog (and all file dialogs are the standard ones).

In the next chapters we will see how to build more complicated pictures so the next example may be helpful.

Back

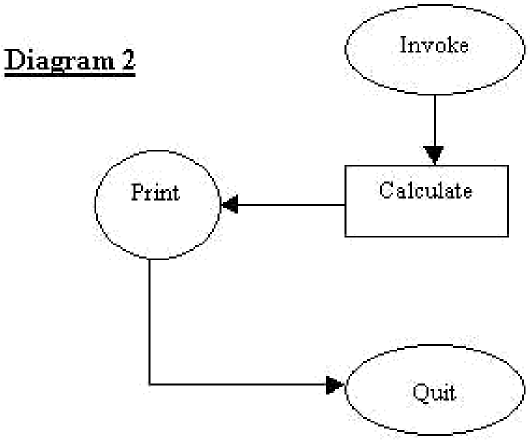
2.Introducing edit toolbar and element properties

The next example will introduce how to use the Edit toolbar and learn how to modify element properties.

Edit toolbar has all standard file and edit options performed by clicking appropriate icon.

- File options available: **New, Open, Save,** and **Print**.
- Edit options available **Cut, Copy , Paste , Delete , Undo** and **Redo**
- Special options

Let s modify a little our first diagram:



First of all we delete second arrow (from middle box to last ellipse).

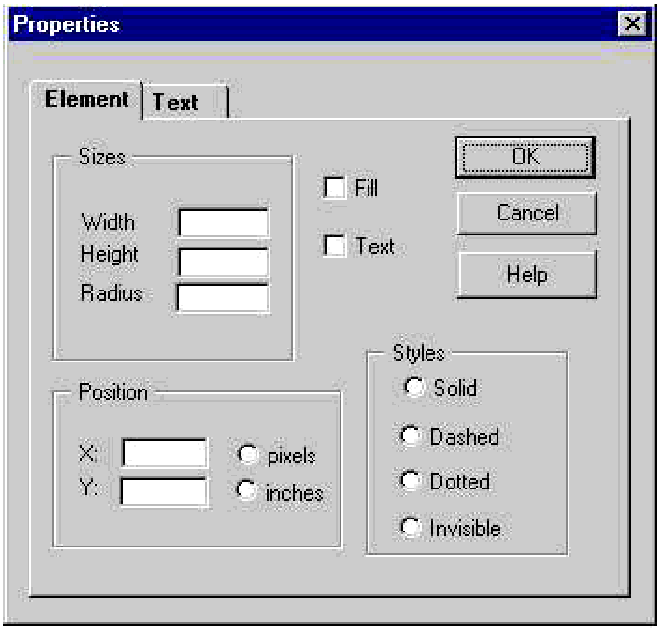
Click on arrow to make it current. It s time to clarify what current element is. You should keep in mind that program always defines current element (even when you open new file current element is default insertion position of first token).

By definition: *Current is the element inserted or modified last*.

Insertion of new element will be done after **Current** in specified direction as you already saw. All element manipulation via **Edit toolbar**, **Edit menu** or mouse are related to current element if not specified otherwise .

Now we can delete an arrow clicking *Delete* button at **Edit toolbar**. What element will be current now? Yes, previously modified (in our case previously drawn) a Box.

Then let s modify text in the box . Clicking the right mouse button causes menu to appear. Further we call this menu **Current element menu**. Current element menu has all basic editing functions such as **cut , copy , paste , delete** .Also some specific options Properties (we use it now) ,**Attach label** ,and **To, At** drawing option. So to modify text we choose Properties option. A properties dialog will appear:



`Remark:` Inches and pixels radoibuttons are non-relavant ...

To modify a text we use text page in it . Text page contains all information about text associated to the element. For our box dialog specifies text <do smth> . Other text options will be explained later. Now remove old text and insert desired one. After *Enter* or *OK* button is clicked entire picture is changed. Further we do some already familiar operations:

Specify left direction at left panel
Clicking arrow button
Clicking circle button (when with/without checkbox is checked)
Insert text Print in a dialog
Changing direction to down

Clicking line button
Changing direction to right
Clicking arrow button

What we’ ve got? Of course, not a desired diagram. What happened? All pic tokens has as mentioned some default attributes some of them are height and width. A default line **width is 0.5** inch while we need it **0.75** inch to reach the de sired point on the ellipse big diameter. The same argument keeps for arrow also. To change specified element sizes we use already mentioned current element menu and **Properties dialog**.

Don’ t forget to select the line before modifying its attributes. Choose*Element* page in the dialog. To modify appropriate fields choose it by mouse or pressing the Tab key. Replace line height to **0.75.** Doing that we get desired line. F or arrow we need to change its size and location. It can be done using the same dialog for the arrow.

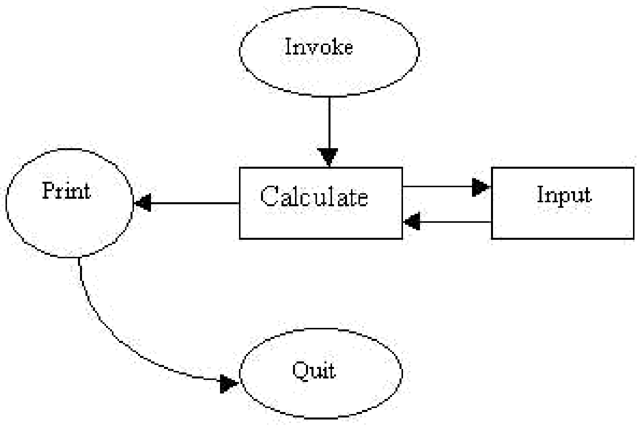
Location changes in dialog location field. Choose inch representation and add **0.25** inch to Y location. Finally we’ ve got diagram 2.

Back

3. Some direct mouse manipulations

Consider we want to improve previous diagram in next way. The program transition state has also read input state. It can be as follows: program asks for input and after user specifies the input program makes calculations and prints results as earlier.

Diagram 3



Also we can improve our diagram replacing lines by arc. Now we have some approaches to do that. Truly speaking we almost always have at least two ways : using program interface or just typing in text window.

In this case we ll explain how to use the interface to achieve our goal. First let s concentrate on<*Input*> box part. We have two ways to do it using interface.

- We can specify **At** position for arrow and draw it from that position then draw box and finally specify **At** position for second arrow and draw it.

- We can draw arrow from calculate box then move it by mouse to desired location then draw box input and repeat with second arrow the same operation.

Suppose you want to use first alternative .Let us clarify the meanings of **At, To** options in **current element menu**.

At option is used to specify point on element border from it next element be drawn. This mode of operation allows to specify point only on current element border.You'll see the **point slider** which moves only on element border

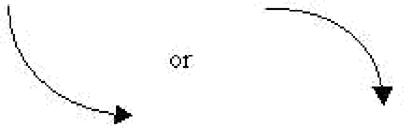
To option is used to specify end point of line/arrow/arc and draw it.

However we will use **At** button from the Pic token toolbar to show more interface options . It does almost the same thing but gives a posibillity to specify every point on the screen to be **At** location point for next insertion. For inserting standalone element it's clear that only this option of interface gives you desired effect (excluding text window of course). After *At* button clicked user specifies **At** point using mouse. The sign you are working in At , From-To mode is a different cursor looks like **+** .
Then choose left direction and click on arrow. Seems difficult? We can suggest you another alternative. Select <*calculate*> box to set it current then choose left direction and draw arrow. We can move it using mouse only. Click on arrow and without release left mouse button move it to needed position. This is so-called drag'n ' drop mode .By mouse you can move element/s and change their size.

Note that not all positions is available for moving. Program has an invisible grid so that elements are drawn from its default corners and at least you can move or enlarge element as this grid size **0.05x0.05** inch . In next section we' ll introduce grid settings and how they can be changed using advanced program features.

Also you should notice that all elements are standalone and changing position one of them doesn t cause others to change. In case you want to change position of others affected by previous change you should use group move introduced further.

We suppose you will like the second way more but sometimes you can specify position-using mouse and the first alternative become a useful one.You can add the second arrow in way you like more and the last thing we haven t done is replacing line and ar row by an arc. This as you guess can also be done or by mouse or using current element menu and properties dialog. Anyway you must draw a default arc first. Then you can change its position and size by mouse. The only thing you can t do without properties dialog is specifying arrowhead to arc.
Another thing, which we not faced, is arc curvative direction:

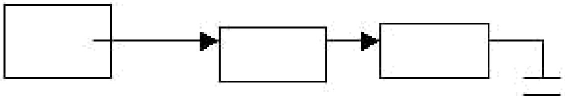


It is because of default arc curvative direction is anticlockwise. It can also be changed using element Properties dialog. In the next example we introduce some group manipulation, which also can be more useful and quick than every group element manipulation.

4.Direct Mouse manipulations with group of elements

Consider the next example:

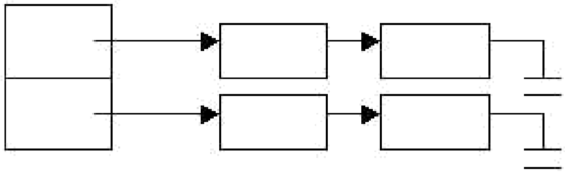
Diagram 4



We have a diagram that shows linked list and suppose we need to do diagram for a hash table. What can be done to make our work more efficient? Program allows you to make manipulations with group of elements. Group selection done by mouse. Click left mouse button and without release it pick out rectangle containing desired group of elements. Actions you can take with group are **Copy, Cut, Paste, Move** and **Delete**. To do a hash we need to copy the list and paste it to another place.

Edit toolbar Copy button and then Paste button may do copying.

Diagram 5



You should specify direction of copy as usual, this means where copy of group will appear. In our example we will choose down direction when the first box is marked as current. New copy will appear exactly after old list in this direction. In o ur example we want the old group and the new one to be close one to other so we will remain the new copy as she. If you need the new group to be at another location you can move it here.

Back

5. Using text window

If you are familiar with Pic language and prefer writing code to mouse manipulation you should use Text window for creating your picture. You have two Text windows:

One small text window at the bottom of the main window - here you have only few lines for inserting code

Second is just opened text editor invoked by clicking on Text Editor option in the View popup menu.

Second option is a regular text editor and **Diamond Draw** doesn’ t support any new operation in it. So we will concentrate on small text window. Here you have number of useful functions (in addition to regular editor functions) that you don’ t have in regular text editor. First of all everything typed in this window interpreted line by line and performed immediately so you can see the picture on the screen without compiling your program. Second you have an automatic line appending (if possible) by clicking the Tab key and automatic spelling. Wrong word will be signed by color. All these make your work with the text window more pleasant than regular text editor. So small not always weak!

Back

6. Advanced text manipulations and line styles .

Suppose we want to create the next diagram. Besides of unusual sizes we see here different font styles. Also we ll talk a little about text adjustment and everything you might faced with working with text.

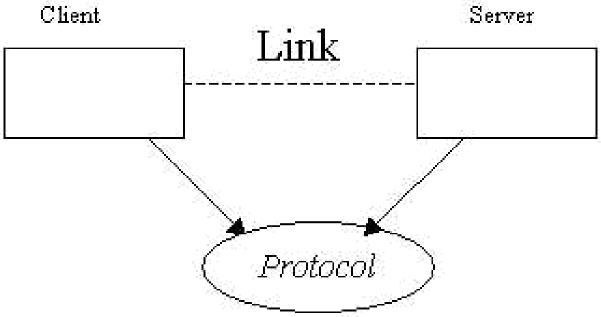


Diagram 6

We leave you to create initial diagram with protocol text in ellipse. Let s begin with client   server text adjustments. It as mentioned earlier also associated text is an optional element attribute. Associated text can be modified in properties dialog. In the dialog choose Text page and enter <*client*> <*server*> appropriately to the boxes. Then choose the *Above* radiobutton to specify that text will be placed above the box. In the same way we adjust <link>   text to line, but the text will be just Link   and not a desired Link .As you guess should change a font size for this string. Use Font size combo list at Edit toolbar. In our case the font size is 16, but this is unimportant detail. And finally we change size for < protocol> string and font style also using Edit toolbar.

Now a little about style option. All element we have drawn up to now used solid line style but it can be changed. Style can be specified as global for all drawing elements or only for current. (This is new option,that is not supported by PIC) In our example we made line Link dashed. Think a little where we change it? Of course using **current element menu** and element **Properties menu**. There we have styles settings: solid, dashed, dotted and invisible.

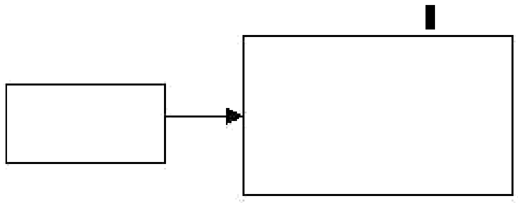
To modify style globally we must use **Edit toolbar**.

Back

7.Specific features built-ins, grid and defaults

Consider the next diagram:

Diagram7



The height and width of the big box is twice as a small one.Of course we can do it in the regular way (changing box's properties) but there is another approach. We can define height and width of an element in terms of other element size. In this case we'll define the big box size in terms of small box. For this purpose you should set relative sizes by choosing the **Relative Settings** menu option from the **Options** pulldown menu. The cursor will be changed and then you should pick up the basic element.

Basic element - the element which defines relative settings by its size.

For elements which have one dimention: arc, circle we define their size as relative width parameter. Now we can set the big box size as:

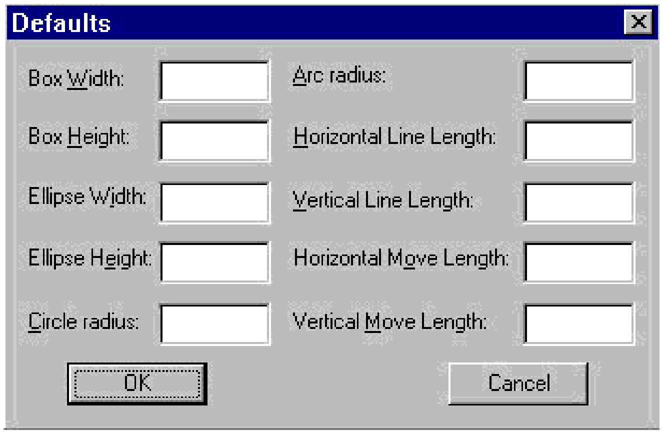
width: 2x
height: 2x

in the **Element Properties** menu instead of regular absolute size.Ok, but what is good for you can ask .Imagine you have group of elements and their sizes mentioned in terms of size one of them. If you decide to change them you don't have to do changes to everyone but only to element which defines **Relative sizes.**

Suppose we need to draw number of boxes with equal but not default size.
As you already noticed all elements in **Diamond Draw** has default parameters. By parameter we mean not only the size but also line style, text style and size etc.
Here we consider some ways to do it.

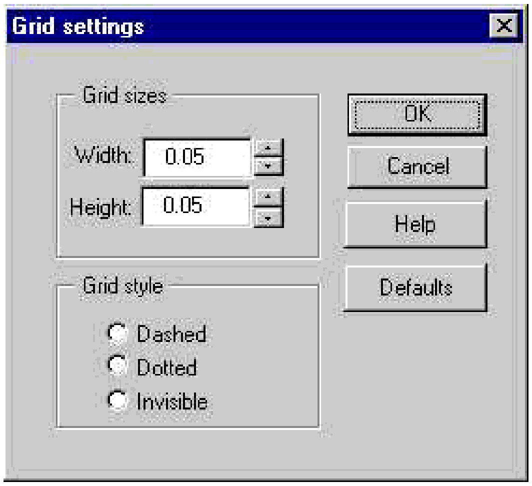
- One way to do it is set Defaults checkbox off and insert size for every box in **Properties dialog** which appear for every new element.
- Second one is specifying size for first box and make others using **Copy** or **Same** options. **Same button** on **pic token bar** performs an internal language command *same* , which just copies a current element in current direction.
- Third one is to use program **Built-ins**. In our case it s a**Loop** button at **pic token bar**. First we must note that program have also a **for..** command which may be conditional and more complicated. Our loop is exactly matches our example draw same element number of times. So we can use it only specifying number of copies.
- And last way we consider (but not really last one) to do it is to change default size for box. You should keep in mind that those changes will take effect for all boxes that drawn from now and till next change. So if you decided to change default size for box you can do it by **choosing Set Defaults option** in **Options** menu. The **Defaults Table** will appear and you can update any field

there.



Another feature that may be useful in your work in **Diamond-Draw** is **Grid**. Program have a default invisible grid which is **0.05x0.05** inch. As mentioned earlier you can drag an element only to grid corners.In the same way size changes proportionally to grid size. One can set grid to be visible using **Grid button** at Edit toolbar. You also can change grid size choosing **Grid settings** option in Option menu.

Following dialog will appear:



Width: 0.05

Height: 0.05

Changing grid size means that all further moving or element size changing will be possible proportionally to new grid settings.

The other thing you should note is that grid in fact **relative** to the first drawn element.Program centers the entire picture always using default grid multiples.(Const * 0.05) Our assumtion that element can be drawn only from grid corners,as a matter of a fact, makes default grid seem to be absolute.

However in case grid sizes changed using relative grid make sense because system centering picture may cause run out of entire grid corners.

Back

8.Mastering the interface

Sometimes you need to use "unusual" elements diamond,polygon, rectangle etc. We say unusual because they not supported by pic language directly. However we can use system standart library **macros** and save some time creating them.

Macros - once created diagrams consisting of basic pic elements which may be efficiently used in other diagrams.

Standart library can be browsed after you've clicked *Custom* button at **Pic token toolbar**. If you use one of the standart macros you have a posibility to set their necessary parameters using dialog for which of them. What you can do if desired element is not in standart library ? System allows you to create a diagram and then use this Diagram (Macro) in every place you need. Inserting a macro done by **InsertFile ,AppendFile options** in **File menu**. Appending mean inserting a macro at end of entire file of edited diagram. Anyway the picture updates and you can see the effect Inserting cause. if desired element is not in standart library ? System allows you to create a diagram and then use this Diagram (Macro) in every place you need.

Back

9. More

To read about specific options you should use a full manual. Here we provide some useful hints which save your time. First of all the manual you have just read can be found in program itself. To read about specific tools you see in the main window you can just click the Help button on Edit toolbar and with new cursor ? (means you are in help mode) point on interested graphic tool. To read more about other things you can, using Help menu and full manual divided by different contexts into parts.

Working with a program not a first time you can pay attention that most of time you are looking for appropriate button or menu option. To get the same effect of desired action managed using accelerators a special key sequences that perform an ac tion. The first thing you may think is how can I memorize all this stuff? The answer is simple. All key sequences organized as in DOS edit. For example you need Edit menu Move option you type Alt + E, M. Elements drawn by Ctrl + first letter of element besides of Arc performed by Ctrl+R (because we already have an arrow).

Finally we say that in order to use all program power you should learn how to work in text window. This of course forces you to learn PIC language itself a little. Doing some things using a text window allows to save more time but need more intelli gent steps. What are you like more depends on you only.

Back

Glossary

The modes of operations

First of all we must to do some assumption:

- Click on smth. means click by left mouse button on smth.
- Choose smth means the same.
- In the each dialogue window the *OK* button serves for performing the operation while the *Cancel* button discard the changes made in the dialogue and cancels the operation.
- In each dialogue window the *Enter* key is equivalent to pressing button while *Tab* key serves for passing to the next parameter of the dialogue.
- In the **Options** pop-up menu when the ’V’ sign is shown near the option it is **ON** otherwise the option is **OFF**.
- **Current picture** means the picture currently shown in the main window.
- In our program clipboard is simply Pic file.
- **Current element** is always defined. If no element is selected the last placed element is current.

Files manipulations:

- To open new picture:
  - From *File* pop up menu choose *New* or just click to the *New* button on the toolbar.
  - The main window will be clear and you can start to draw the new picture now.
- To save the picture shown on the screen with specified file name:
  - From *File* pop up menu choose *Save As*. In the appeared dialogue window specify the disk, directory and file name where you want the picture to be stored. Click the *OK* button to save the picture or *Cancel* button to discard the action.
- To save the picture with previously specified file name:
  - From *File* pop up menu choose *Save* or just click the *Save* button on the toolbar. If the file name not specified you will be asked to specify it by the Save As dialogue window.
- To open the previously saved picture:
  - From *File* pop up menu choose *Open* or just click the *Open* button on the toolbar. In the appeared dialogue window specify the disk, directory and file in it. You should also indicate the type of files you want to be shown, that is extension. As a rule the Pic files has the **.pic** extension.
- To close the current picture:
  - From *File* menu choose *Close*. If you have done some changes after the last saving the you will be suggested to save the picture before closing.
- To preview the current picture before printing:
  - From *File* pop up menu choose *Print Preview*. You will see the picture as it will be printed.
- To print the current picture:

    - From *File* pop up menu choose *Print* or just click the *Print* button on the toolbar. In the appeared dialogue window choose the printer name and number of copies to be printed.
- To append the file (containing picture) to the end of current picture:
    - From *File* pop up menu choose *Append File*. In the appeared dialogue window specify the disk, directory and file in it. You should also indicate the type of files you want to be shown, that is extension. As a rule the Pic files have the **.pic** extension. The chosen Pic file will be appended to the current file and the resulting picture will be shown on the screen.
- To insert the file (containing picture) into the current picture:
    - Select the current element after which you want to insert another file. From *File* pop up menu choose *Insert File*. In the appeared dialogue window specify the disk, directory and file in it. You should also indicate the type of files you want to be shown, that is extension. As a rule the Pic files have the **.pic** extension. The chosen Pic file will be inserted into the current file after current element and the resulting picture will be shown on the screen.
- To exit the program:
    - From *File* pop up menu choose *Exit.*

Special options:

- To change the scale range:
    - From *Options* pop up menu choose *Scale*. In the appeared dialogue window specify the desired scale range and units of measure .
- To change the defaults for figure parameters:
    - From *Options* pop up menu choose *Set* defaults. In the appeared dialogue window specify the default values for figure parameters: width of a box , height of a box , width of an ellipse , height of an ellipse , radius of a circle , radius of an arc , horizontal line length , vertical line length , horizontal move length , vertical move length.
- To show or hide the grid:
    - System has a default grid 0.05x0.05 inch and uses its corners as possible location for element borders.Grid is relative and not influenced upon picture centering related to the elements. Changing grid settings you specify new possible multiples for element size changing.**By default minimal element size changing is 0.05 inch**.
    - There is a *Show Grid* option in the *Options* pop up menu. When the grid is shown on the screen this option is marked by the **V** sign. Choosing this option you show or hide the grid.
    - click *Grid* button on toolbar. Choosing this option you can show or hide the grid
- To change grid settings:
    - Choose *Grid settings* option from *Options* pull down menu. In appeared dialog specify needed grid sizes or set differnt grid style
- To set relative setings:
    - Choose *Relative settings* option from *Option* pull down menu.Cursor will be changed and user can specify an element (**Basic element**)which size will initialize relative width and height program parameters.From this moment and up to next change one can use these parameters to set other element size in terms of basic element size.Changing basic element cause all relative elements to change. Syntax in Element Properties window:
        - (size)width : 2x
        - (position) X: +1.5x
        - Cause the element with double basic width to be drawn at X+1.5 basic width,Y relative to current insertion point X,Y.
- To show or hide the labels attached to elements:

    - You can attach the label to each pic element and refer to it in future code by this label. To learn more about labels see the Pic language documentation.
    - There is a *Show Labels* option in the *Edit* pop up menu. When labels are shown on the screen this option is marked by the **V** sign. By choosing this option you show or hide the labels.
- To show or hide the tips suggested by the system:
    - Tips are short help messages suggested by the system. Tips refer to the object currently pointed by the mouse cursor.
    - There is a *Show Tips* option in the *Edit* pop up menu. When tips are shown on the screen this option is marked by the **V** sign. By choosing this option you show or hide the tips.
- To show or hide the toolbar:
    - There is a *Show Toolbar* option in the *Edit* pop up menu. When the toolbar is shown this option is marked by the **V** sign. By choosing this option you show or hide the toolbar.
- To change the font style and size:
    - Choose the *Font* option in the *Edit* pop up menu. In the appeared dialogue window specify the font style and size. You can do the same from the toolbar. On the right side of toolbar there are two combo lists: *Font Style*and *Font Size* By clicking on these lists you get the suggested fonts or sizes.

Edit operations:

Assumptions: Most of these operations are duplicated at the pop up menu appeared when the right mouse button is pressed. They also present at **Edit toolbar**.

- To select the single element:
    - Just click on it.
- To select the group of elements:
    - Click the left mouse button and drag the mouse without releasing the button. When you release the button the rectangle is specified. All the elements within this rectangle become selected.
- To select all elements on the current picture:
    - From *Edit* pop up menu choose *Select All*. All elements will be marked by the dashed frame. Now you can copy them to the clipboard or cut them.
- To clear the picture:
    - From *Edit* pop up menu choose *Clear*. All the element of the current picture will be deleted. The same result is achieved when you are selecting all the element and then cutting them.
- To move the selected element/s:
    - Select the element/s. From *Edit* pop up menu choose *Move*. Now you can move the element to another location by keyboard arrow keys. When you want to finish the moving mode just press *Enter*.
    - You can also drag element by mouse.Select the element/s and drag it to desired location.
- To cut the selected element/s:
    - Select the element/s. From *Edit* pop up menu choose *Cut* or just click the *Cut* button on the toolbar. The selected element/s is cut to the clipboard.
- To copy the selected element/s:
    - Select the element/s. From *Edit* pop up menu choose *Copy* or just click the *Copy* button on the toolbar. The selected element/s is copied to the clipboard.
- To paste the element/s from the clipboard:
    - From *Edit* pop up menu choose *Paste* or just click the *Paste* button on the toolbar. The

clipboard s content will be placed after the current element.

- To undo the last element:
    - From *Edit* pop up menu choose *Undo*. From the nested pop up menu choose *Last Element.* If the last operations was done on group of elements the previous picture state will be restored.
- To undo the last parameter changing:
    - From *Edit* pop up menu choose *Undo*. From the nested pop up menu choose *Last Parameter*.
- To redo the last operation:
    - From *Edit* pop up menu choose *Redo* or just click the redo button on the toolbar.
- To fill the current element:
    - From *Edit* pop up menu choose *Fill*. The current element will be filled. This action is only available for the box, circle and ellipse.
- To attach the label to the current element:
    - From *Edit* pop up menu choose *Attach Label*. In the appeared dialogue window type the text that will be the label to the current element.
- To change the current element s properties:
    - From *Edit* pop up menu choose *Properties*. In accordance with the current element the appropriate dialogue window will appear. There are three kinds of properties dialogues:
        - Properties of box/circle/ellipse.
        - Properties of line/arc/spline.
        - Properties of standalone text.

First two dialogues contain two pages: **Element** and **Associated Text**. In the element page you can modify element s parameters at the AssociatText page is dealing with the text associated with the currer element. In this page you can set , modify or delete the associated text.

In the suggested **Element**page you can change the properties relevant for the current element. Irrelevant properties will be unavailable. For example in the dialogue No.1 the radius field will only be available for the circle while width and height will only be available for box and ellipse. You can modify the position , style , size and the fill attribute of the element.

In the second kind of the **Properties** window you can specify properties of line , arc or spline. You can specify the Topoint of the element when the start point is a previous element. If you want the element to be standalone set the **Standalone** radio button to the ONstate and specify the Fromparameter. You can also change the direction , radius (not for a line) or style (solid , dashed , dotted , invisible) and set the arrowhead.

For the standalone text you can modify the text itself and its position.

Built-in operations:

- To define the If ... Then ... Else statement:
    - From the *Built-in* pop up menu choose I*f ... Then ... Else* option. In the appeared dialogue window specify the condition and then-else operations. For more information see the Pic language documentation.

- To define the For ... To ... By statement:
    - From the *Built-in* pop up menu choose *For ... To ... By* option. In the appeared dialogue window specify the loop conditions and the operation. For more information see the Pic language documentation.
- To draw element number of times
    - From the *Built-in* pop up menu choose *Loop* option.In the appeared dialogue window specify number of copies and after that select element position and direction to draw.Element will be drawn this number oftimes in this direction from this position

View operations:

- To view the source pic code of the current picture:
    - From *View* pop up menu choose *View Source.* The separate text window with the pic source code will be opened.
- To edit the source pic code of the current picture:
    - From *View* pop up menu choose *Text Editor*. The default Text editor is opened with the source code of the current picture.

Element manipulations:

Assumptions :

All those operations,except *Custom* function ,may be performed from **text window** .

There is a *Defaults On/Off* radio button on the left side of the window. If it is on the new element is drawn with the default parameters otherwise the **Properties** dialogue window will appear every time you are drawing a new element.

You can attach a text for any Pic element. There is a *Text On/Off* radio button. If it is ON the **Attached text** dialogue window will appear every time you are drawing a new element otherwise elements are drawn without text. The text you are typing in this dialogue will be attached to the drawn element.

There is also standalone text which is the pic element and may be placed at any location in the picture.

All the elements can be drawn by the direct typing of pic commands in the text window described below. In this section we describe only the WYSIWYG approach for elements creation.

- To draw a box:
    - From *Elements* pop up menu choose *Box* or just click the *Box* button on the left part of the window.
- To draw a circle:
    - From *Elements* pop up menu choose *Circle* or just click the *Circle* button on the left part of the window.
- To draw an ellipse:
    - From *Elements* pop up menu choose *Ellipse* or just click the *Ellipse* button on the left part of the window.
- To draw a line:

    - From *Elements* pop up menu choose *Line* or just click the *Line* button on the left part of the window.
- To draw an arrow:
    - From *Elements* pop up menu choose *Arrow* or just click the *Arrow* button on the left part of the window.
- To draw an arc:
    - From *Elements* pop up menu choose *Arc* or just click the *Arc* button on the left part of the window.
- To draw a spline:
    - From *Elements* pop up menu choose *Spline* or just click the *Spline* button on the left part of the window.
- To draw the standalone text:
    - From *Elements* pop up menu choose *Text* or click *Text* button on the left part of the window. The Text dialogue window will appear. Type the text you want to be printed on the picture.
- To draw same element as previous one :
    - From *Elements* pop up menu choose *Same* or just click the *Same* button on the left part of the window
- To draw Custom element:
    - From *Elements* pop up menu choose *Custom* or just click the *Custom* button on the left part of the window.In appeared dialogue window choose element from standart library of elements.Next dialog allows you to set neseccary parameters for it.
- To change the drawing direction:
    - There are four buttons on the left side of the window: **Up, Down, Left, Right**. Click the appropriate button and all next drawings will be performed in this direction until you change it.
- To change the drawing style locally:
    - From the mouse right button pop up menu choose the desired style (solid, dashed, dotted or invisible) Current element will be redrawing with this style.
- To change the drawing style globally:
    - Click the appropriate button on the left side of the edit toolbar All new elements will be drawn with the new style until you ll change it.
- To change the element size:
    - Do the desired element current by the left mouse button clicking. Click on the dashed frame marking the element and drag it without releasing the mouse button. If there are some restrictions on the size they are kept during resizing. For example you can change only length for the line while you can change both width and height for the box. When resizing the circle the height and the width stay the same i.e. it remains to be circle.
- To move the element to another place:
    - Do the desired element current by the left mouse button clicking. Click on the element center and drag it without releasing the mouse button then drop it at needed place by the mouse button releasing.
    - Another way is to choose the Move option from the Edit pop up menu and then move the current element by keyboard arrow keys.
- To change the line or arrow direction:
    - Click the right vertex of the line or arrowhead. Drag the point to desired position without releasing the button.

At, To position actions:

- To specify standalone position for element:
    - click on *At* button at the left panel.Cursor will change to "+" to sign you are in **At** mode specify At location clicking left mouse button. Next elemnt will be drawn from this point in specifyed direction.
- To specify At location on element border:
    - click right mouse button.On appeared menu choose At option. By point slider you can specify definite position on current element border.Next elemnt wiil be drawn from this point in specifyed direction.
- To specify To element location(line/arrow/arc may be drawn to specifyed point from current element insertion point):
    - click right mouse button.On appeared menu choose To option. In dialog choose one from **Arc,Arrow,Line**.Then you can specify end location of chosen element by left mouse button.Specifying location chosen element will be moved according to the mouse moving and redrawn dashed. After **To** position specifyed element redrawn finally in currently defined style.
- To draw line,arc,arrow from definite to definite points:
    - click on *From-To* button at the left panel.Cursor will change to "+" to sign you are in At mode specifying source point.After point specification in dialog choose one **from arc,line,arrow**.Then you can specify destination point location of chosen element by left mouse button.Specifying location chosen element will be moved according to the mouse moving and redrawn dashed. After **To** position specifyed element redrawn finally in currently defined style.

Text window operations:

There are 4 lines in the text window placed at the screen bottom. The upper line is for system messages. Two next lines are intended for last two performed pic commands and the last line is intended for typing. There you can type the pic command and its result will appear on the picture when it become the valid pic command that is if you type box the box will drawn when the x letter is typed and it will be erased if you erase the x letter because bo is not a valid pic command.

The command typed in the text window is inserted to the pic source file as it was in the direct source pic code editing.

Another feature of this mode is automatic token completion. If you press the Tab key during the command typing the token is completed automatically if the typed word is a prefix of any valid pic token. If you use UNIX operating system you should be familiar with this feature -- in UNIX it exists for file names.

There is also automatic speller in the text window. Each word that isn t the valid pic token will be marked by the red color font.

Getting Help:

- To get the general information about the **Diamond-Draw** system:
  - From *Help* pop up menu choose *About*.
- To get the tutorial:
  - From *Help* pop up menu choose *Tutorial*.
- To search the help information by topics:
  - From *Help* pop up menu choose Index. In the appeared dialogue window there are an index of topics and edit box where you can type the wanted topic. When you type it the topic become active in the index window and you can get it in the separate window just by clicking on it.
- To get help on graphic interface componenets:
  - Click "?" button at Edit toolbar and point by new cursor to desired interface component.Small text window with explanation will be opened. To get help on next component just point on it.Right mouse button cancels this mode of operation.

Back